Data Scientist

STUDENT GUIDE

PEGA Academy

This document is current as of the date of publication only. Changes in the document may be made from time to time at the discretion of Pegasystems. This document remains the property of Pegasystems and must be returned to it upon request. This document does not imply any commitment to offer or deliver the products or services provided.

This document may include references to Pegasystems product features that have not been licensed by your company. If you have questions about whether a particular capability is included in your installation, please consult your Pegasystems service consultant.

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors. This document or Help System could contain technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Pegasystems Inc. may make improvements and/or changes in the information described herein at any time.

This document is the property of:
Pegasystems Inc.
1 Rogers Street
Cambridge, MA 02142
Phone: (617) 374-9600
Fax: (617) 374-9620
www.pega.com

Mission: Data Scientist

Products: Pega Platform™ ’24.1, Pega Customer Decision Hub™, Pega Customer Service™

URL: https://academy.pega.com/mission/data-scientist/v7

Date: 3 April 2024

Contents

Customer Decision Hub overview

Description

Familiarize yourself with the one-to-one customer engagement paradigm and discover how Pega’s omnichannel AI delivers the right action during every customer interaction.

Discover Next-Best-Action Designer, a powerful tool that guides decisioning architects in creating personalized customer experiences across all channels. Define business structures, implement constraints, configure prioritization, and deliver the next best actions through inbound, outbound, and paid channels.

Explore the Impact Analyzer tool, that conducts experiments by implementing alternative prioritization and engagement policy filters on a small control group of customers, to assess the effectiveness of next best actions.

Learning objectives

- Explain the basics of the Next-Best-Action approach.
- Describe the purpose of the Next-Best-Action Designer components: Taxonomy, Constraints, Engagement policy, Arbitration, and Channels.
- Analyze the decisions for a specific customer in Customer Profile Viewer.
- Describe the critical roles in implementing Pega AI.
- Use Impact Analyzer to monitor the performance of your next best actions.

1:1 Customer Engagement paradigm

The optimal outcome of every customer interaction is to provide a great experience while maximizing the value of the customer to the company. To achieve this outcome, you must perform the right action in the right channel at the right moment for each customer. In Pega Customer Decision Hub™, this feature is known as One-to-one Customer Engagement.

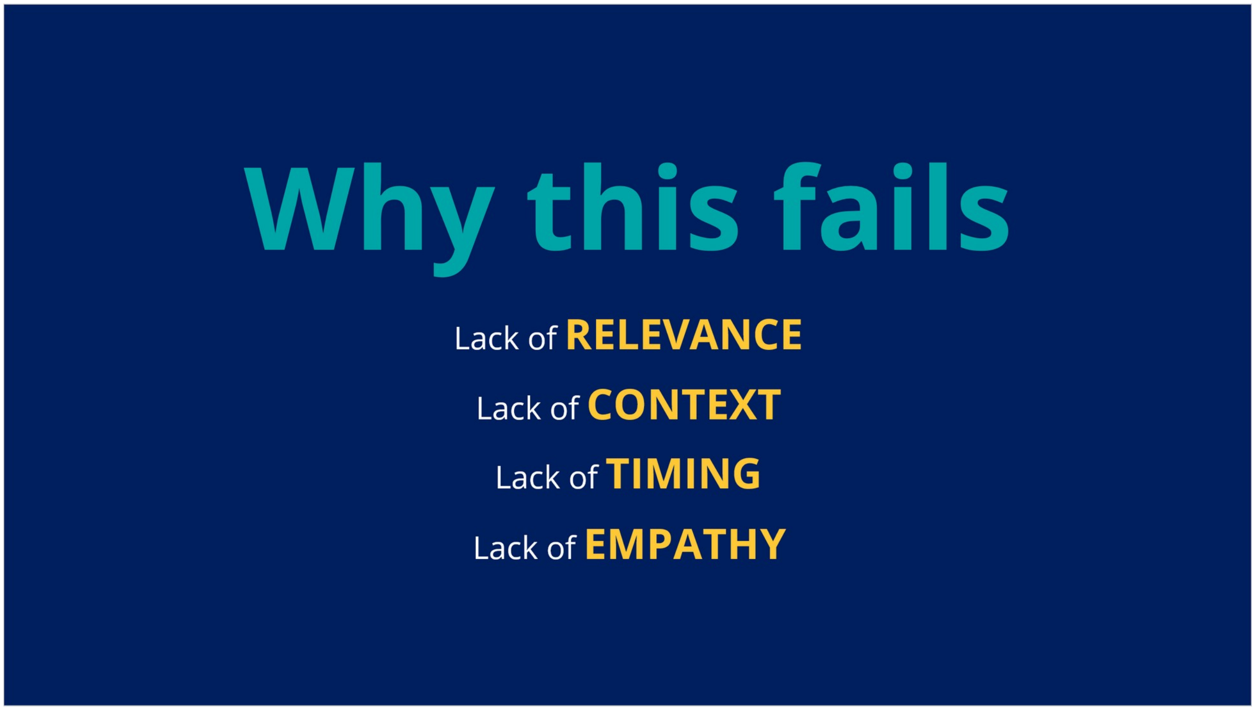
Transcript

The optimal outcome of every customer interaction is to provide a great experience while maximizing the value of the customer to the company.

In a traditional approach marketers search for potential customers from a database based on target demographics, geographies, or financial means to make a purchase. Then, marketers target this customer segment across all channels, and all these customers receive offers for a specific product.

The problem with this approach is that only a low percentage of customers respond or make a purchase. Marketers might hit their short-term goal, but in the long term, this approach affects the relationship with customers.

As a result, the traditional marketing approach fails because of a lack of relevance, context, timing, and empathy.



Why this fails

Lack of **RELEVANCE**

Lack of **CONTEXT**

Lack of **TIMING**

Lack of **EMPATHY**

Customers are more empowered than ever before. As a result, they have very high expectations for the experiences they receive from their service providers. Their experiences must make sense within the context of their lives and this means they must be meaningful, consistent, and personalized across every channel with which they interact.

In business, the optimal outcome of every customer interaction is to provide a great experience while maximizing the customer's value to the company. To achieve this, you must perform the right action in the right channel at the right moment for each customer.

Traditional mass-marketing techniques that use segments, batches, and campaigns do not move the needle anymore; they are antiquated and unsustainable. The market has recentered around real-time technologies and one-to-one customer interactions.

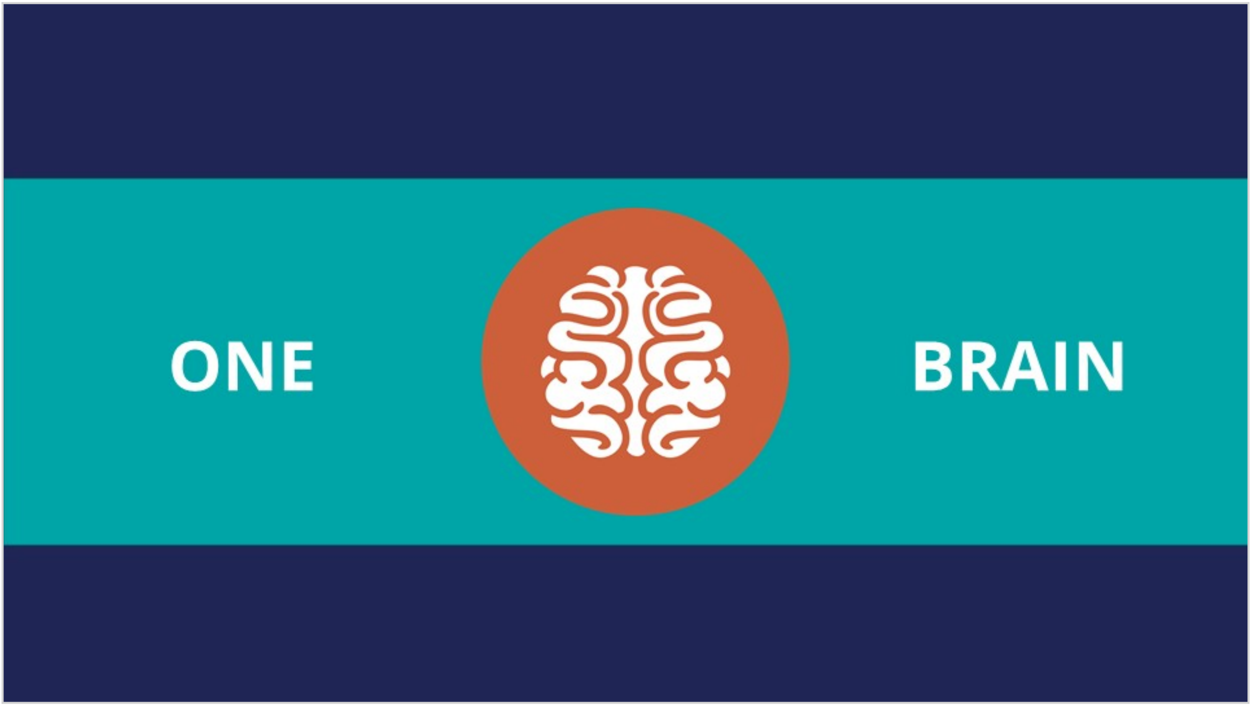
The Pega approach to these interactions is through One-to-one Customer Engagement.

Through One-to-one Customer Engagement, companies can transition their marketing away from a traditional one-to-many campaign-driven approach. A one-to-one approach allows companies to have consistent, contextual, and relevant conversations with individual customers across any channel or touch point.

The key to achieving One-to-one Customer Engagement is one centralized brain.

In other words, one piece of intelligence acts as a single decision authority across your application ecosystem.

Each channel or system profits from this single source of customer intelligence and can use it to gain insights or perform relevant actions.



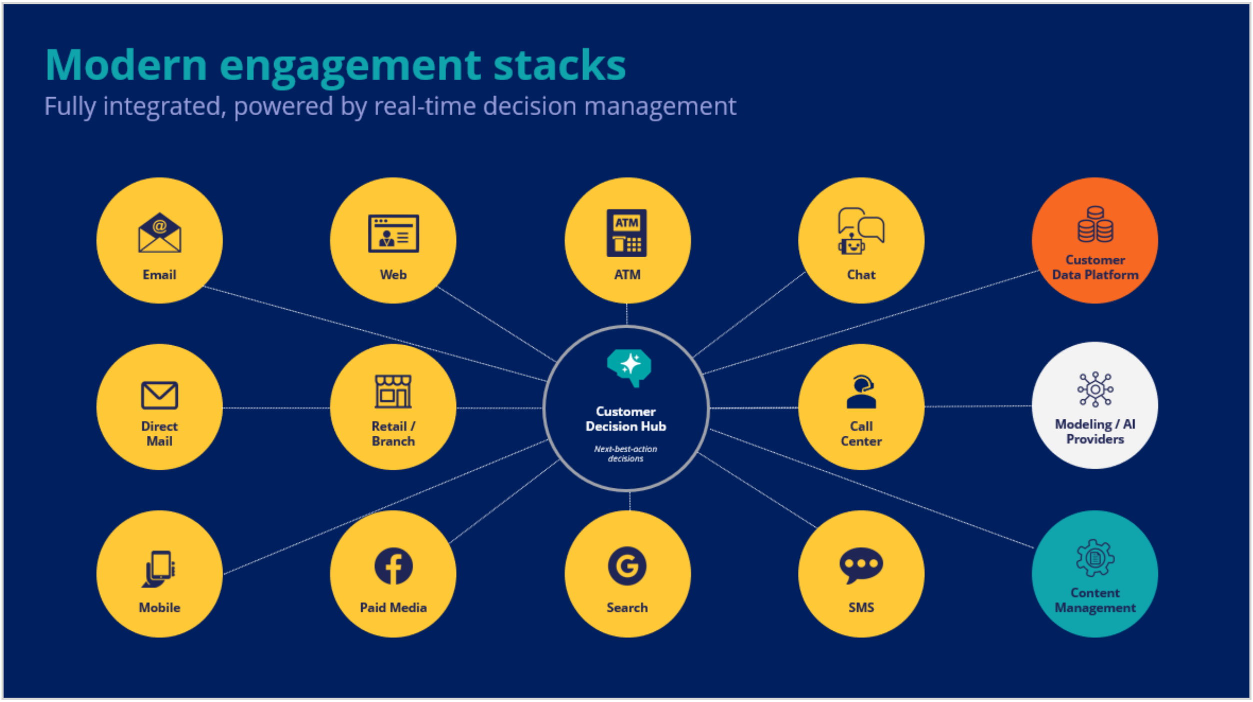
In Pega Customer Decision Hub, this centralized brain is the core feature that uses AI to enable One-to-one Customer Engagement.

In Pega Infinity, Pega Customer Decision Hub forms the core of the customer engagement platform, which sits at the center of existing systems and channels in an enterprise.

The “brain” collects data from every customer engagement across the enterprise to create predictions and decisions about every interaction in every channel.

Pega Customer Decision Hub can directly integrate with third-party content management platforms such as Adobe Experience Manager and use the content that you develop in these platforms for personalized customer engagement.

Continuous learning and decision-making are the foundation of a One-to-one Customer Engagement solution.



Modern engagement stacks

Fully integrated, powered by real-time decision management

Email

Web

ATM

Chat

Customer Data Platform

Direct Mail

Retail / Branch

Customer Decision Hub

Next-best-action decisions

Call Center

Modeling / AI Providers

Mobile

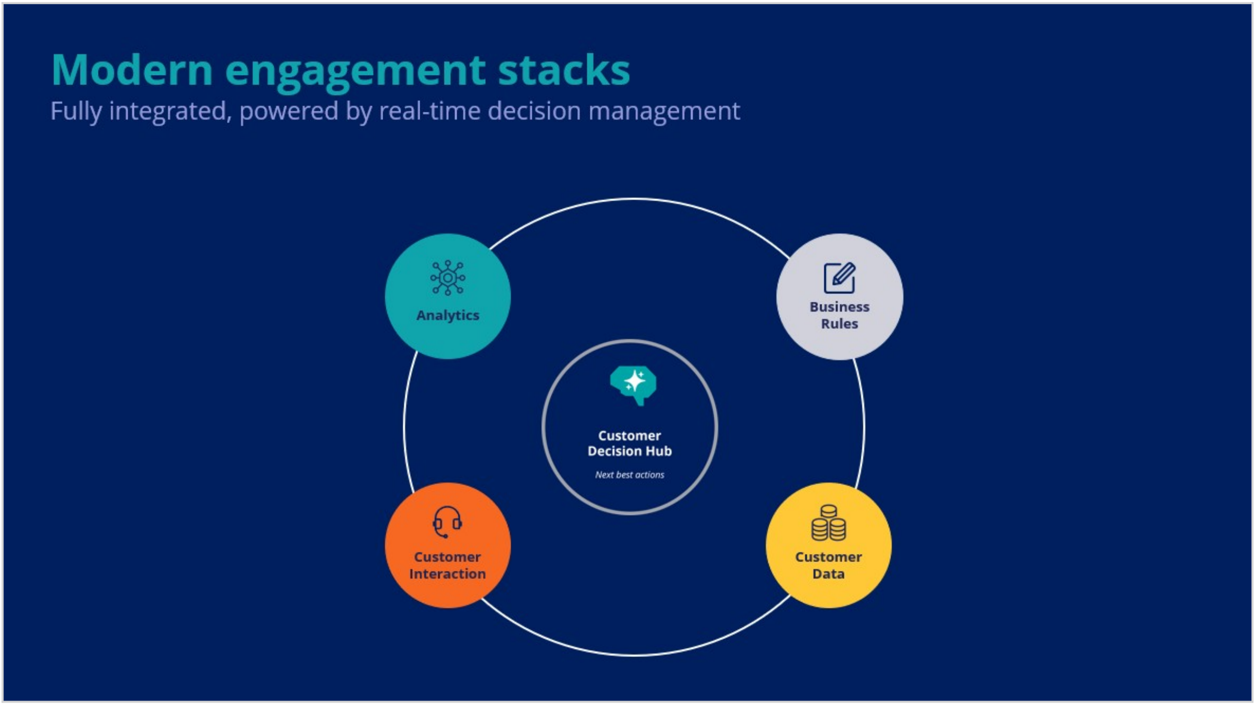
Paid Media

Search

SMS

Content Management

Customer Decision Hub combines analytics, business rules, customer data, and data collected during each customer interaction to create a set of actionable insights that it uses to make intelligent decisions. These decisions are known as the next best actions.



Modern engagement stacks

Fully integrated, powered by real-time decision management

Analytics

Business Rules

Customer Decision Hub

Next best actions

Customer Interaction

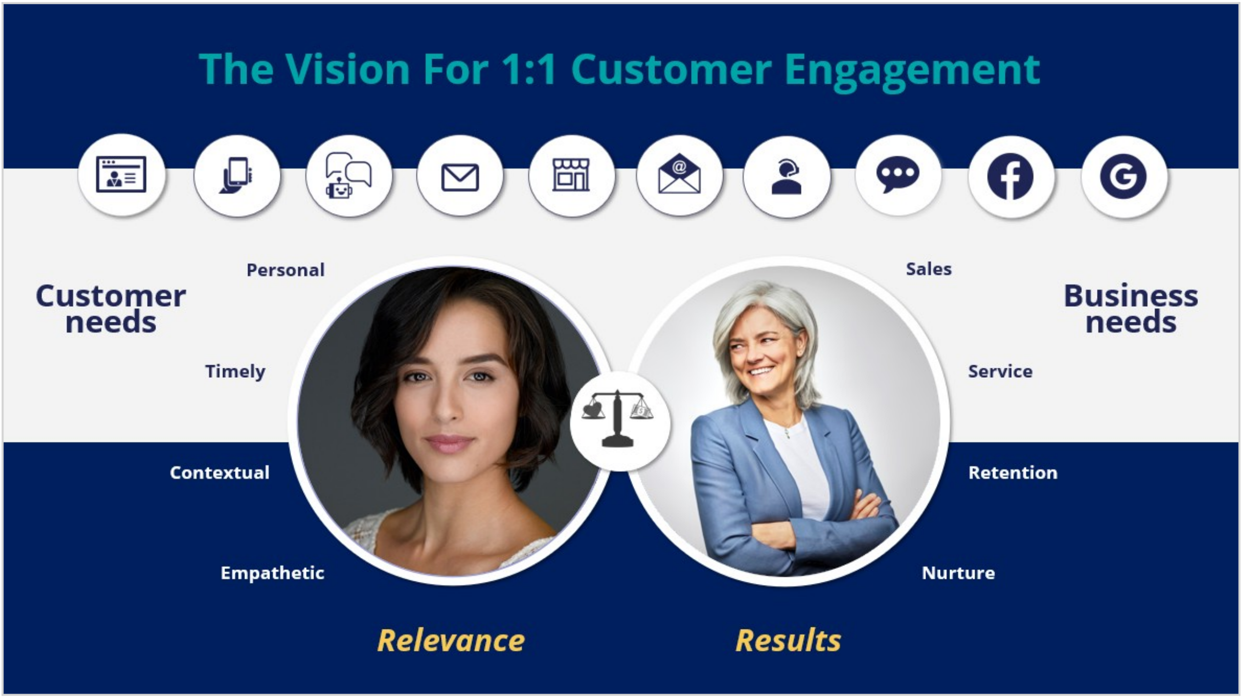
Customer Data

Every next best action weighs customer needs against business needs to optimize decisions based on priorities set by the business manager.

In the milliseconds before interacting with a customer, Customer Decision Hub combines thousands of business rules and predictive and adaptive models that determine customer needs and interests to ensure the next best action is relevant by keeping it personal, timely, contextual, and empathetic.

Customer Decision Hub identifies the best moments to make a sale, provide a service, make a retention offer, inform about an offer, or do nothing at all (for example, if no offer is relevant enough to warrant the customer's attention). The system distributes next-best-action decisions in real time to your channels, such as the web, mobile, and contact center. Pega Customer Decision Hub can also distribute next best actions to real-time paid channels such as Google, YouTube, Facebook, LinkedIn, and Instagram. Pega Customer Decision Hub also integrates with non-real-time outbound channels such as data management platforms (DMPs) and email.

After the system distributes the next best actions and the "brain" receives customer responses, the process begins again. Customer Decision Hub distributes new next best actions in milliseconds. It captures every customer interaction in every channel to ensure consistency and an optimized customer experience across channels.



For example, consider a customer, Miranda. With the centralized brain in place, instead of looking only at sales offers, the system begins with a list of all potential actions, such as service, retention, nurture, or a hardship message, everything that you can do for this customer now.

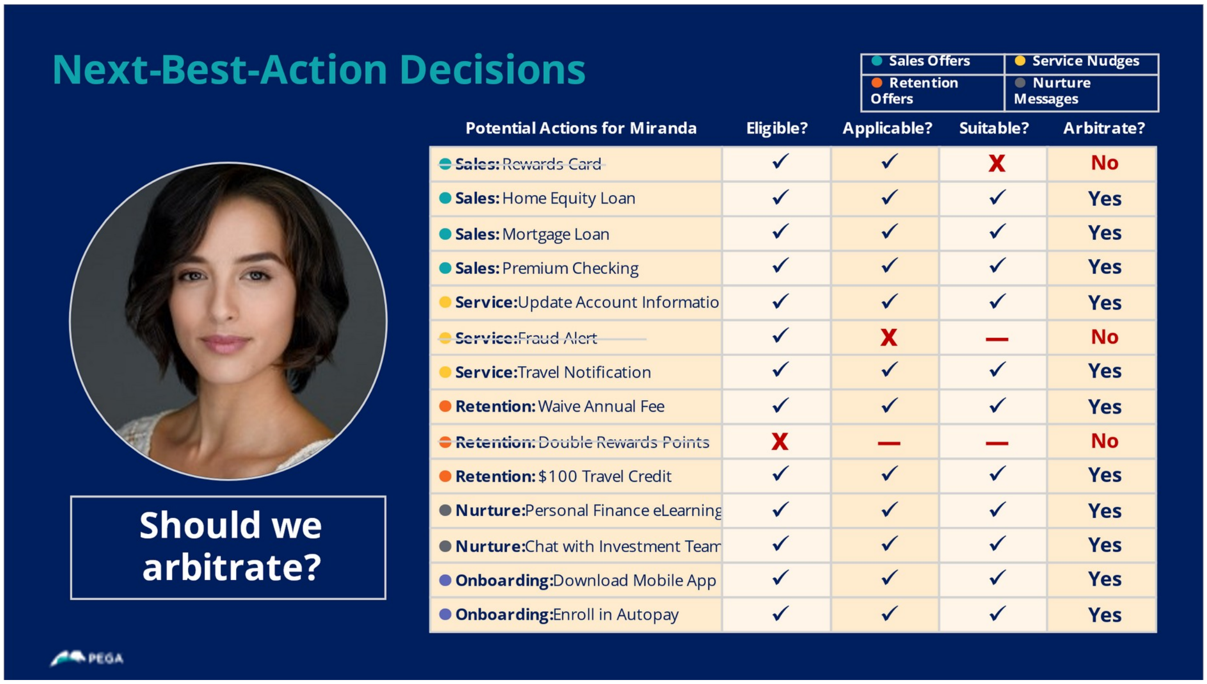
Although AI drives the next best actions, AI does not run rampant; it experiments on every person with every topic. Customer Decision Hub still allows marketers to maintain control and establish the criteria that AI must meet to consider beginning one of these conversations.

For example, you establish eligibility rules that state the business cannot sell a card to someone under the age of 18.

Then there are applicability rules to define if the action is appropriate at a given time. If Miranda already owns a competing or more valuable product, the business does not offer this product to her even though she is eligible.

Suitability rules determine whether an action is in the best interest of a customer. For example, Miranda might be eligible for a card, and her current card does not offer as high of a cash-back offer. But because Miranda cannot make her monthly payments and might end up in collections, the business does not offer the card to her even though it can.

So, if a customer fails to meet the conditions that you establish for an action, the AI does not consider that action.



Next-Best-Action Decisions

● Sales Offers	● Service Nudges
● Retention Offers	● Nurture Messages

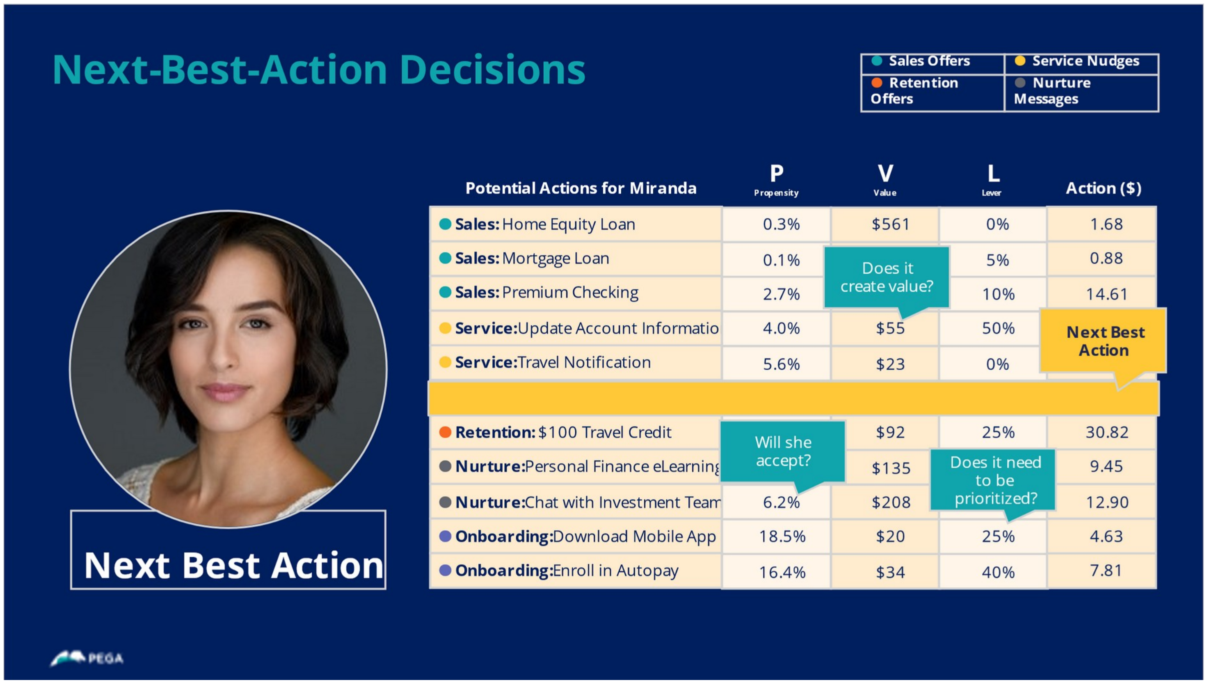
Potential Actions for Miranda	Eligible?	Applicable?	Suitable?	Arbitrate?
~~● **Sales:** Rewards Card~~	✓	✓	X	No
● **Sales:** Home Equity Loan	✓	✓	✓	Yes
● **Sales:** Mortgage Loan	✓	✓	✓	Yes
● **Sales:** Premium Checking	✓	✓	✓	Yes
● **Service:**Update Account Informatio	✓	✓	✓	Yes
~~● **Service:**Fraud Alert~~	✓	X	—	No
● **Service:**Travel Notification	✓	✓	✓	Yes
● **Retention:** Waive Annual Fee	✓	✓	✓	Yes
~~● **Retention:** Double Rewards Points~~	X	—	—	No
● **Retention:** $100 Travel Credit	✓	✓	✓	Yes
● **Nurture:**Personal Finance eLearning	✓	✓	✓	Yes
● **Nurture:**Chat with Investment Team	✓	✓	✓	Yes
● **Onboarding:**Download Mobile App	✓	✓	✓	Yes
● **Onboarding:**Enroll in Autopay	✓	✓	✓	Yes

Should we arbitrate?

Once the system narrows down the list, AI takes over. AI first determines how likely each action is desirable to the customer. Next, it determines the value each option generates for the business. What is the impact if the customer reads the content or accepts an offer? Does it give you more revenue or reduce your costs?

You can add levers to make adjustments based on the current business situation. For example, the business can decide to nudge an offer if it needs to meet its financial goals or ramp down an offer if it runs low on inventory.

For Miranda, the action that the system selects as the next best action is the one with the highest P*V*L value.



Next-Best-Action Decisions

● Sales Offers | ● Service Nudges
● Retention Offers | ● Nurture Messages

Potential Actions for Miranda	P Propensity	V Value	L Lever	Action ($)
● **Sales:** Home Equity Loan	0.3%	$561	0%	1.68
● **Sales:** Mortgage Loan	0.1%		5%	0.88
● **Sales:** Premium Checking	2.7%		10%	14.61
● **Service:**Update Account Informatio	4.0%	$55	50%	
● **Service:**Travel Notification	5.6%	$23	0%	
● **Retention:** $100 Travel Credit		$92	25%	30.82
● **Nurture:**Personal Finance eLearning		$135		9.45
● **Nurture:**Chat with Investment Team	6.2%	$208		12.90
● **Onboarding:**Download Mobile App	18.5%	$20	25%	4.63
● **Onboarding:**Enroll in Autopay	16.4%	$34	40%	7.81

Does it create value?

Next Best Action

Will she accept?

Does it need to be prioritized?

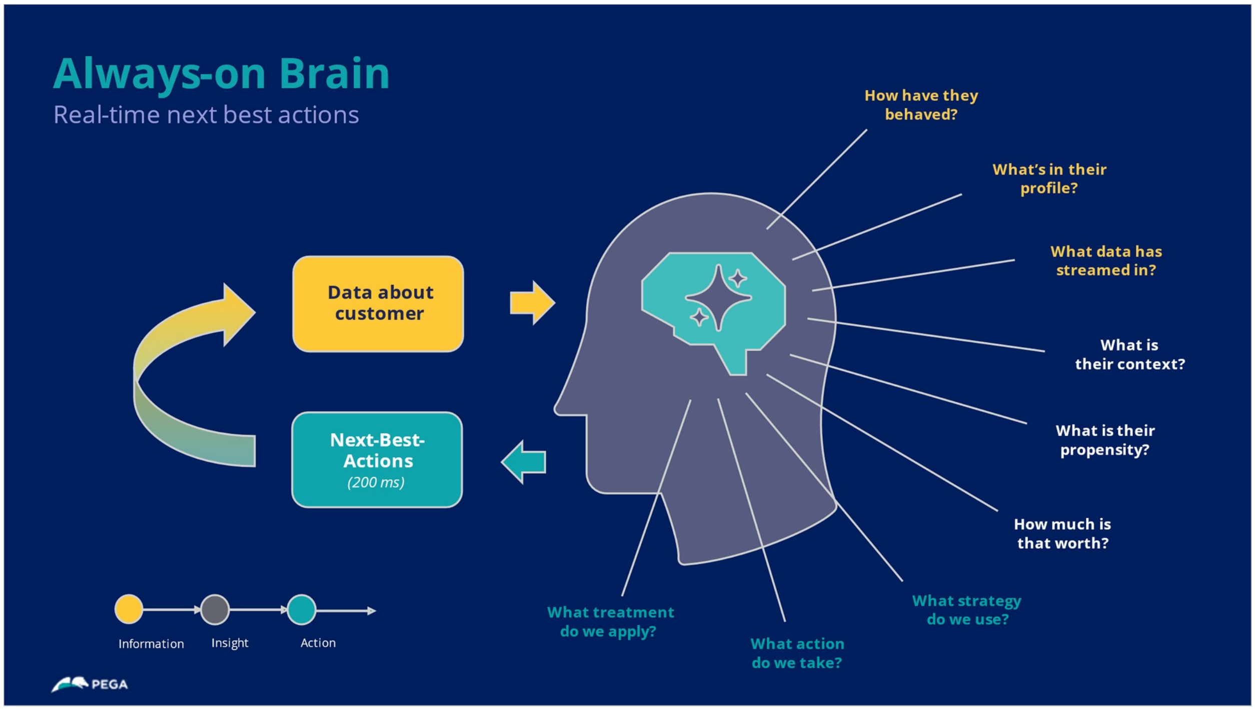
Next Best Action

In summary, Customer Decision Hub is the always-on brain that acts as a single, centralized decision authority.

It uses data about the customer, including past interactions, as input.

It uses advanced AI techniques to make predictions.

Decision strategies (which combine traditional business rules with predictive, adaptive, and text analytics) deliver consistent and personalized next best actions across all channels.



Always-on Brain

Real-time next best actions

Data about customer

Next-Best-Actions *(200 ms)*

Information → Insight → Action

How have they behaved?

What's in their profile?

What data has streamed in?

What is their context?

What is their propensity?

How much is that worth?

What strategy do we use?

What action do we take?

What treatment do we apply?

PEGA

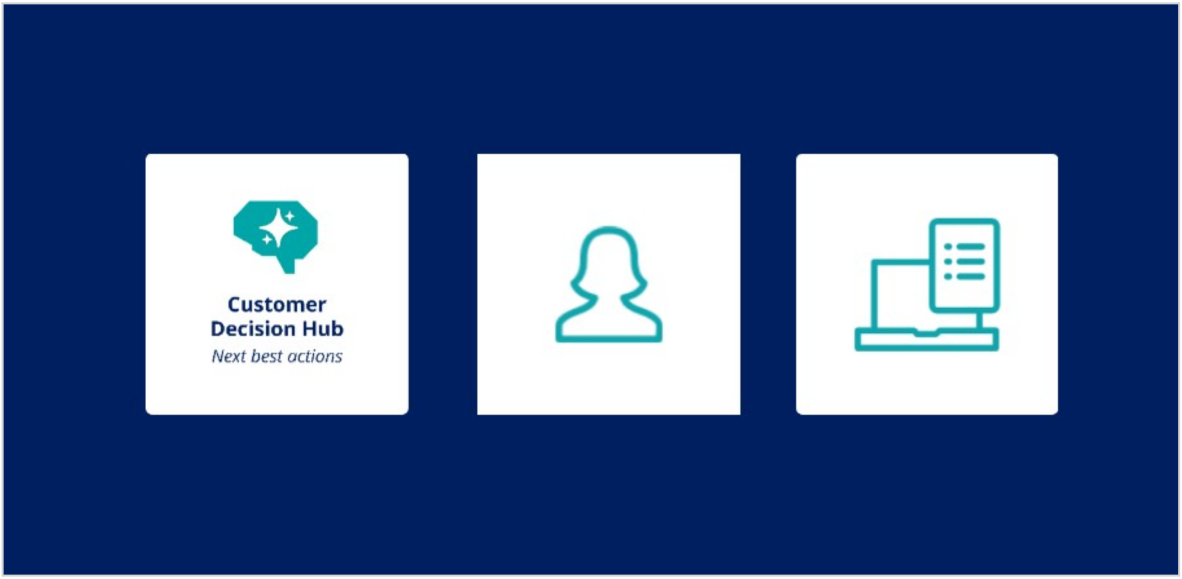
You have reached the end of this video.

The cross-sell on a website use case

U+ Bank uses Pega Customer Decision Hub™ to ensure that the customers of U+ Bank see the tailor-made offers when visiting its website. The centralized decision management "brain" of Customer Decision Hub selects the next best action that is displayed for a customer based on configurations implemented by the Decisioning Architect.

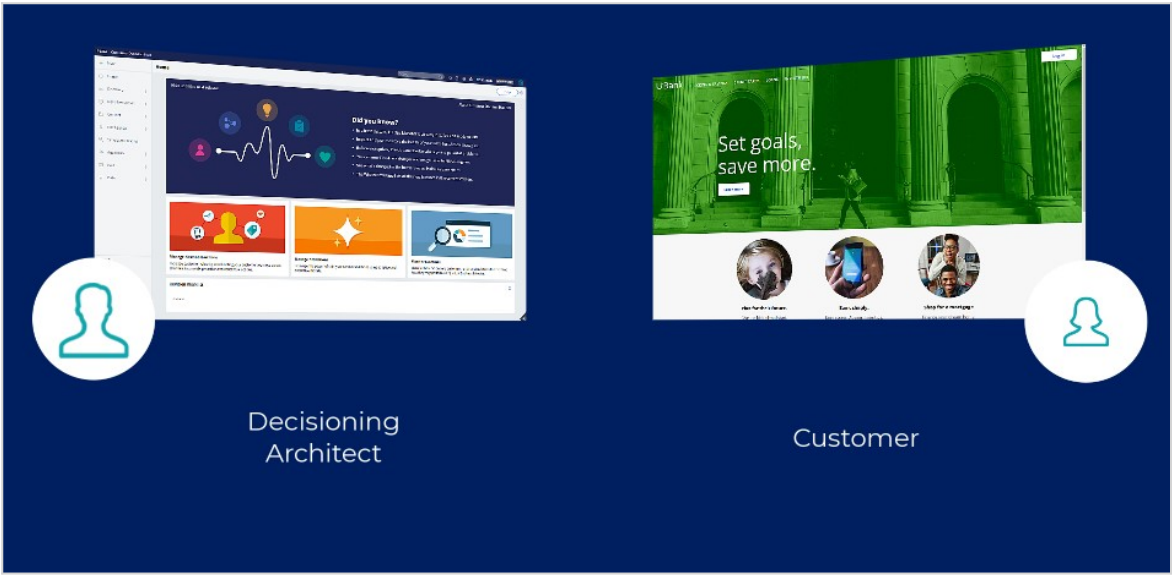
Transcript

Customer Decision Hub can deliver the next-best-action recommendations through various inbound and outbound channels. Outbound messages involve proactively reaching out to customers, while inbound channels demand action from a customer. One such inbound channel is the web. For example, when a customer visits the website, they see the intended offers.



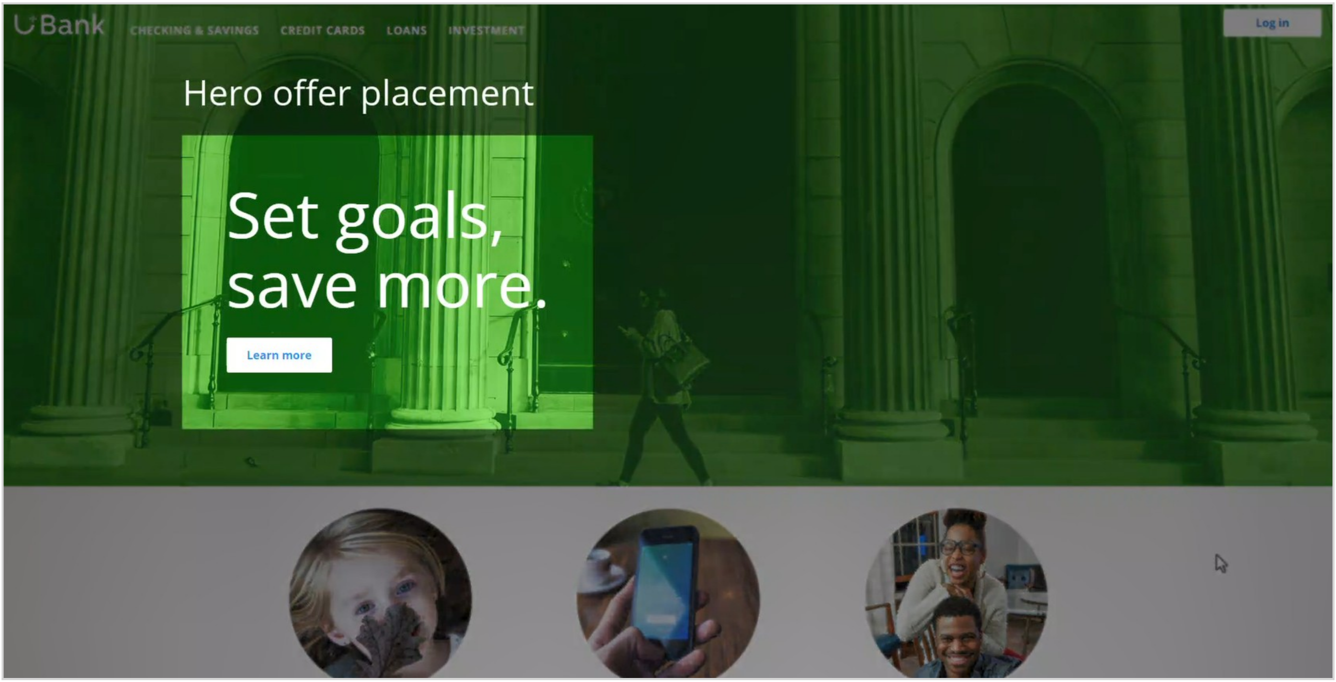
Consider the following web channel scenario, which is a typical cross-selling use case.

The next-best-action recommendations help to ensure that the customers of U+ Bank can see the tailor-made offer when visiting the website of the bank. The centralized decision management "brain" of Customer Decision Hub selects the next best action that is displayed for a customer based on configurations implemented by the Decisioning Architect.

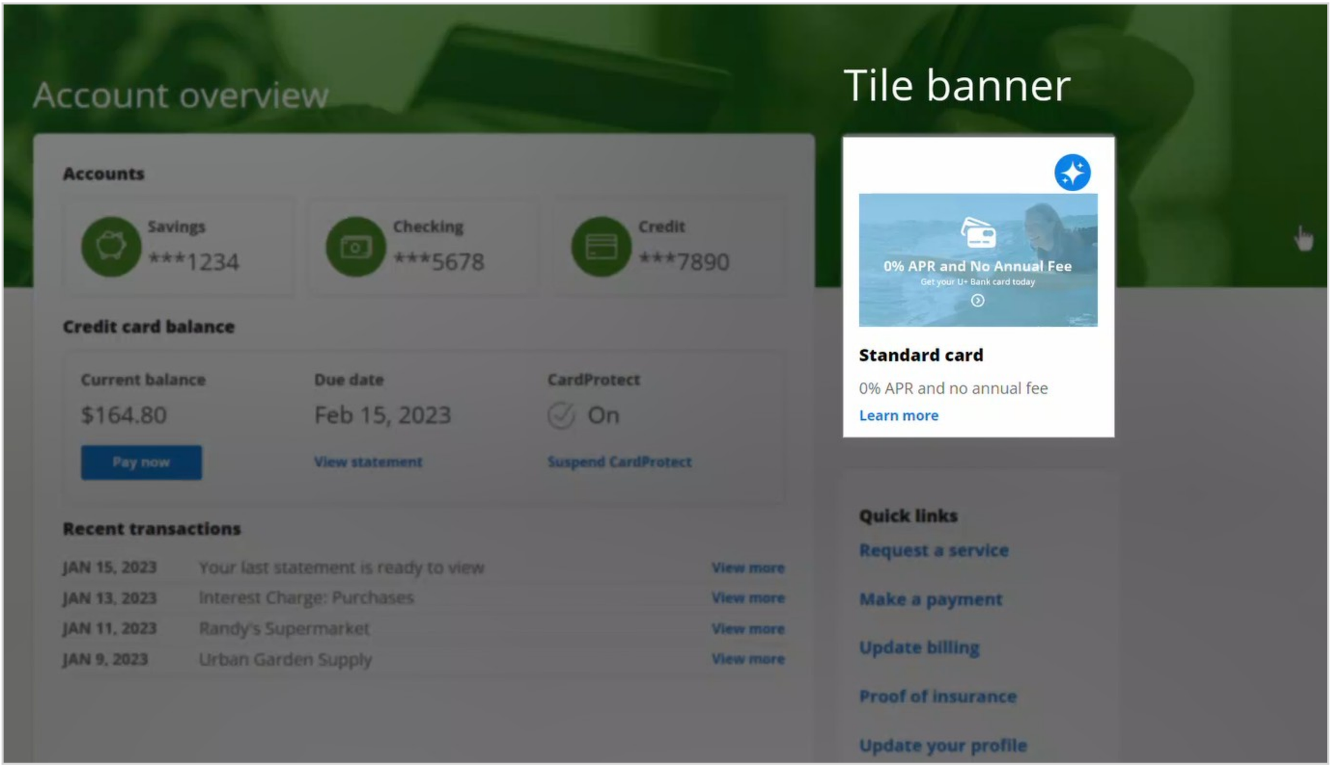


U+ is a retail bank that wants to use its website as a marketing channel to improve One-to-one Customer Engagement, drive sales, and deliver next best actions in real time. The bank has decided to use Pega Customer Decision Hub™ to recommend more relevant banner ads to its customers when they visit the website.

When visiting the U+ Bank website, Troy, a customer, can see banner ads on various pages. For example, on the home page, U+ displays a hero banner at the top of the page, which is typically a larger image with a larger typeface. Under that banner, there is space to display several tile banners, which are typically smaller.



When Troy logs in to his personal portal, he also sees a tile banner on the **Account overview** page.



Account overview

Accounts

Savings ***1234

Checking ***5678

Credit ***7890

Credit card balance

Current balance $164.80

Due date Feb 15, 2023

CardProtect ✓ On

Recent transactions

JAN 15, 2023	Your last statement is ready to view	View more
JAN 13, 2023	Interest Charge: Purchases	View more
JAN 11, 2023	Randy's Supermarket	View more
JAN 9, 2023	Urban Garden Supply	View more

Tile banner

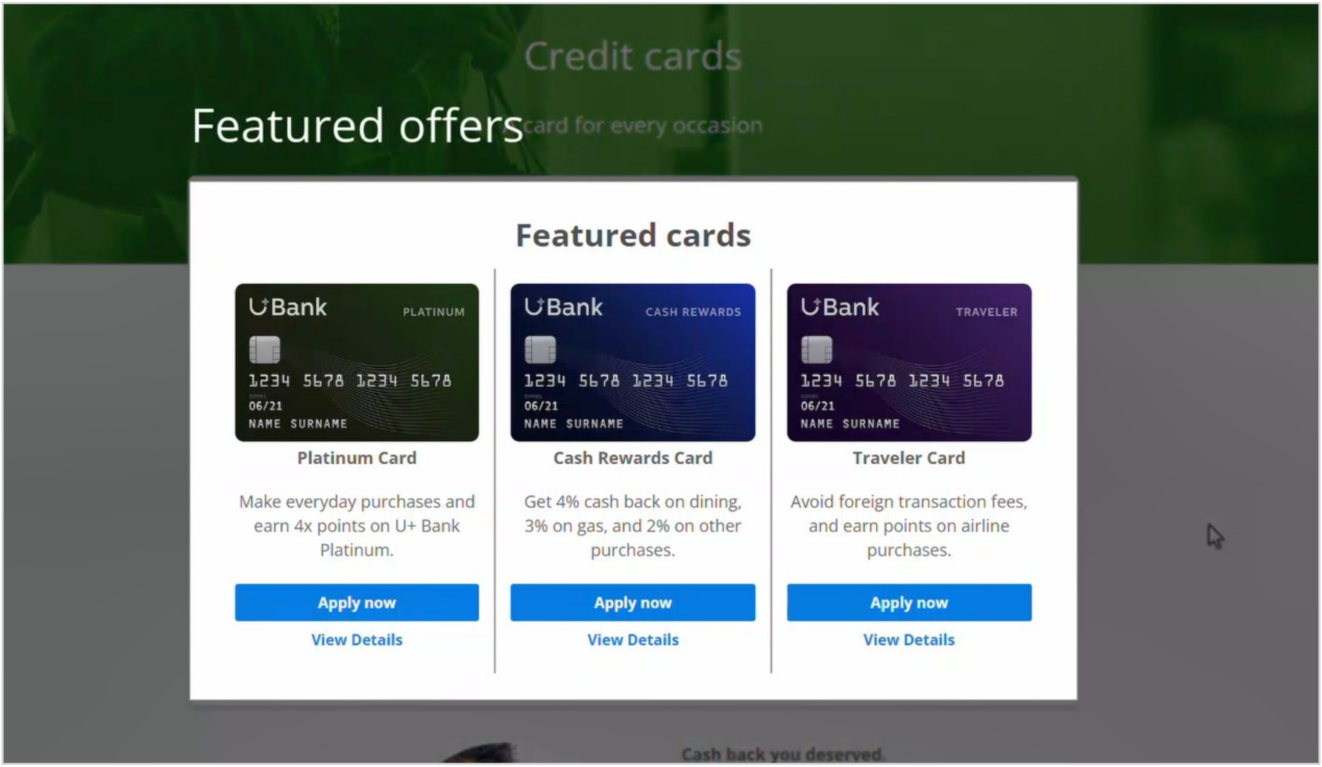
0% APR and No Annual Fee

Get your U+ Bank card today

Standard card

0% APR and no annual fee

The main goal of U+ at this stage is to increase customer web engagement. When Troy clicks the **Learn more** link, the action shows his interest. This interaction is recorded as a click-through and helps measure the web engagement of the customer. After clicking **Learn more**, Troy can see featured credit card offers for which he can apply.



Featured offers

Featured cards

Platinum Card

Make everyday purchases and earn 4x points on U+ Bank Platinum.

Cash Rewards Card

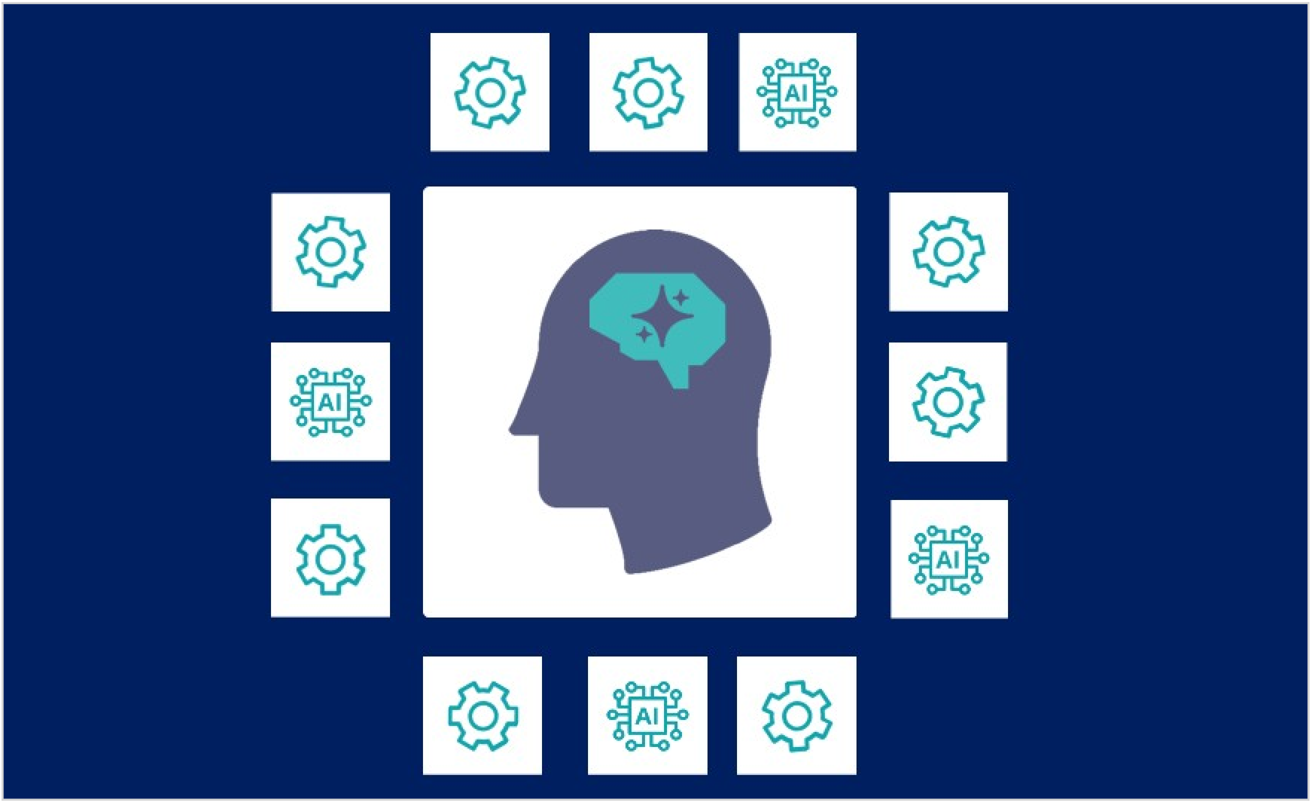
Get 4% cash back on dining, 3% on gas, and 2% on other purchases.

Traveler Card

Avoid foreign transaction fees, and earn points on airline purchases.

Behind the scenes of the next best actions, the complex decision engine is working to rank and select the best offer to display for each customer who visits the website.

A combination of artificial intelligence (AI) and other business rules determine the options.



To summarize, when visiting the bank's website, the customer can see the next best action that the "always on" centralized decision management "brain" of Customer Decision Hub selected from a set of actions defined and configured by the Decisioning Architect.

You have reached the end of this video.

Critical roles in implementing Pega AI

At the start of a Pega Customer Decision Hub™ implementation, having all key stakeholders at the table is essential. Data Scientists play a critical role in the adaptive modeling process. Learn about the common roles in the project and how they interconnect to ensure the successful implementation of Pega AI.

Transcript

This video describes the roles of the key stakeholders in the implementation of a Customer Decision Hub project.

At the start of the project, having all key stakeholders at the table is essential. Data Scientists play a critical role in the adaptive modeling process, as they understand the technical details of the models, and can make recommendations for how they can be improved. They also ensure that the models properly comply with legal and regulatory obligations.

The model governance team provides oversight for all predictive models used by the organization and may require detailed documentation or a review process before the launch.

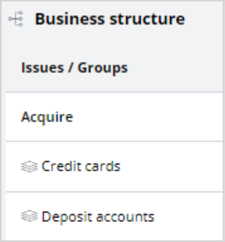
The Data Engineering group is responsible for delivering data to the CDH application, as well as providing clear definitions and dictionaries for the predictors.

The CDH Implementation team is responsible for implementing a Pega Customer Decision Hub application to meet business requirements, and ensuring that the system is configured to support the business processes. The team typically includes System Architects who are familiar with Pega Platform™ and the organization's data model.

Marketing Ops collaborates with the Lines of Business to configure the outbound schedule in Next-Best-Action Designer to drive actions and treatments. They also update strategy logic within the CDH application to appropriately consume the model outputs. This allows for the optimization of customer interactions across all channels. The Lines of Business may have questions about how adaptive models will impact their line's share of voice in channels.

Finally, the Project Governance team manages the project, arbitrates any debates between the different interests among these groups, and ultimately has decision-making authority. It is important to ensure that all perspectives are accounted for, even in organizations with a few people as the main stakeholders. All stakeholders should be involved in the discussion throughout the project's lifecycle. Doing so will ensure the successful implementation of Pega AI.

Determining the **scope** of the project is the first key decision that the project team has to make. To determine the **actions** to include in the project, they identify the specific business issue to address and the objectives of the decision strategy.



Business structure

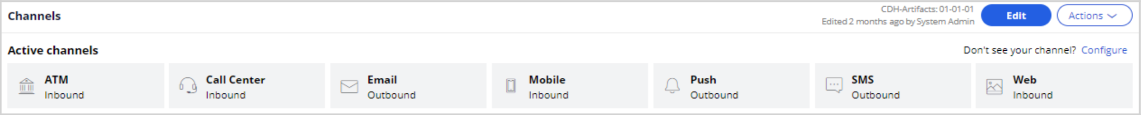
Issues / Groups

Acquire

Credit cards

Deposit accounts

The issue and objectives will also determine which **channels** they use to communicate with customers.



Channels

CDH-Artifacts: 01-01-01
Edited 2 months ago by System Admin

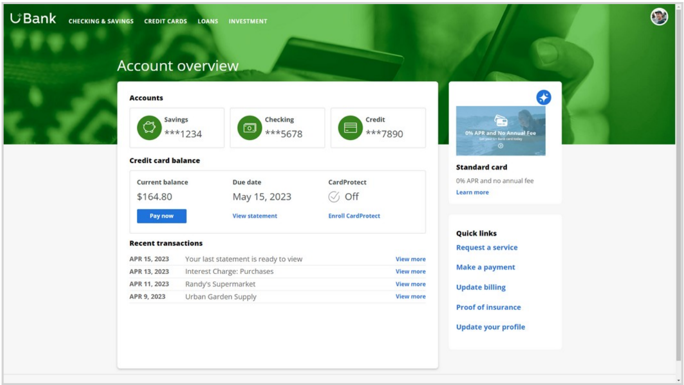
Active channels

Don't see your channel? Configure

ATM Inbound	Call Center Inbound	Email Outbound	Mobile Inbound	Push Outbound	SMS Outbound	Web Inbound

The project team needs to consider which channels will be most effective in reaching the target audience and achieving the desired outcomes.

Targeting customer touchpoints, such as click-throughs, across all applicable channels for all customers can be a good strategy for achieving fast results.



Account overview

Accounts

Savings	Checking	Credit
***1234	***5678	***7890

Credit card balance

Current balance	Due date	CardProtect
$164.80	May 15, 2023	Off
Pay now	View statement	Enroll CardProtect

Recent transactions

APR 15, 2023	Your last statement is ready to view	View more
APR 13, 2023	Interest Charge: Purchases	View more
APR 11, 2023	Randy's Supermarket	View more
APR 9, 2023	Urban Garden Supply	View more

Standard card

0% APR and no annual fee

Learn more

Quick links

Request a service

Make a payment

Update billing

Proof of insurance

Update your profile

Alternatively, launching in only one channel or for actions in certain issues or groups can be a more controlled rollout approach. This can be beneficial for companies that want to test the waters before fully implementing Customer Decision Hub across all business issues and channels.

The success of the project is defined in terms of achieving specific goals or as Key Performance Indicators that are configured by the marketers.

Exploring Next-best-Action decisions

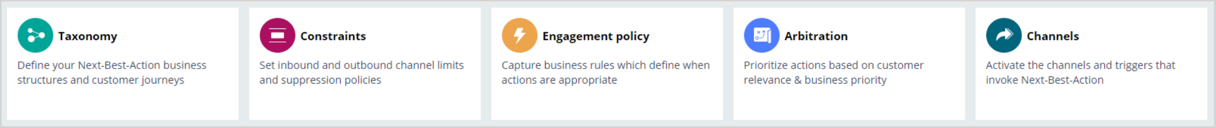
Next-Best-Action Designer organizes the high-level sequence of steps needed to customize the next-best-action strategy to suit the business requirements. Pega Customer Decision Hub™ uses these settings to generate an underlying next-best-action strategy framework and the enterprise-level strategies that come with it.

These decision strategies determine the personalized set of next best actions for each customer. To observe the decisions made for a specific customer, explore the Customer Profile Viewer.

Transcript

Next-Best-Action Designer guides the decisioning architect by creating a next-best-action strategy that delivers personalized customer experiences across all channels. The tabs across the top of the user interface represent the steps to define the next best actions:

- Use the **Taxonomy** component to define the business structure for your organization.
- Use the **Constraints** component to implement channel limits and constraints.
- Use the **Engagement policy** component to define the rules that control for which actions customers qualify for.
- Use the **Arbitration** component to configure the prioritization of actions.
- Use the **Channels** component to configure when and where the next best actions initiate.



Taxonomy
Define your Next-Best-Action business structures and customer journeys

Constraints
Set inbound and outbound channel limits and suppression policies

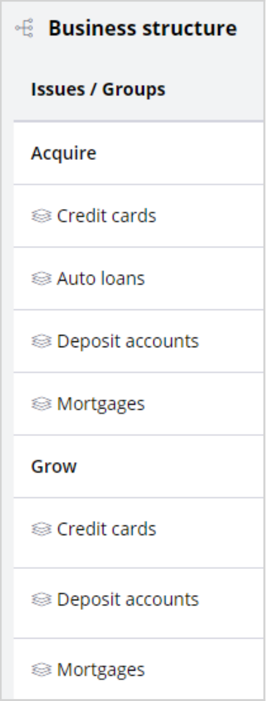
Engagement policy
Capture business rules which define when actions are appropriate

Arbitration
Prioritize actions based on customer relevance & business priority

Channels
Activate the channels and triggers that invoke Next-Best-Action

The system uses these definitions to create an underlying Next-Best-Action Strategy framework. These decision strategies are a combination of the business rules and AI models that form the core of Customer Decision Hub, which determines the personalized set of next best actions for each customer.

Use the **Taxonomy** tab to define the hierarchy of business issues and groups to which an action belongs.

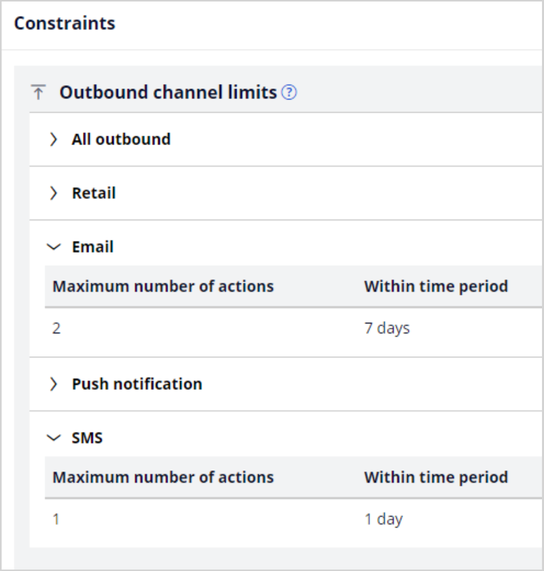


Business structure

Issues / Groups
Acquire
Credit cards
Auto loans
Deposit accounts
Mortgages
Grow
Credit cards
Deposit accounts
Mortgages

A business issue is the purpose behind the actions that you offer to customers. For example, actions to acquire new customers belong to the **Acquire** business issue. The **Grow** business issue groups the actions to cross-sell to existing customers. Business groups organize customer actions into categories. You can create groups for products, such as credit cards, mortgages, or auto loans, to offer these to potential customers.

Use **Constraints** to specify contact limits and limit overexposure to a specific action or group of actions. For example, you do not want your customers to receive more than two emails per week or one SMS message daily.



Constraints

Outbound channel limits

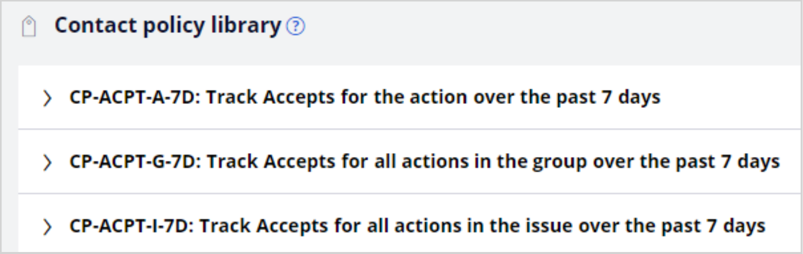
- All outbound
- Retail
- Email

Maximum number of actions	Within time period
2	7 days

- Push notification
- SMS

Maximum number of actions	Within time period
1	1 day

You can define more extensive suppression rules by creating Contact Policy rules in the library. Contact Policy rules are reusable across all business issues and groups. For example, you suppress an action for a customer for seven days after the customer has seen an ad for that action five times.



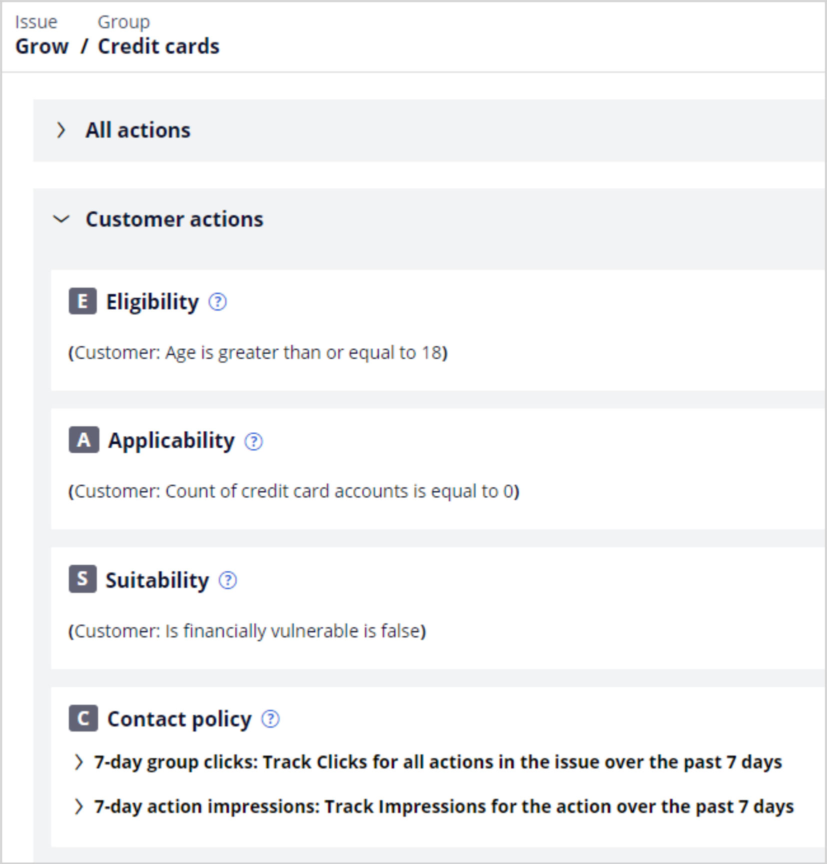
Contact policy library

- CP-ACPT-A-7D: Track Accepts for the action over the past 7 days
- CP-ACPT-G-7D: Track Accepts for all actions in the group over the past 7 days
- CP-ACPT-I-7D: Track Accepts for all actions in the issue over the past 7 days

Use **Engagement policies** to define when specific actions or groups of actions are appropriate for customers. There are four types of engagement policies:

- **Eligibility** determines whether a customer qualifies for an action or group of actions. For example, an action might only be available for customers over a specific age or who live in a specific geographic location.
- **Applicability** determines whether an action or group of actions is relevant for a customer at a particular point in time. For example, a discount on a specific credit card might not be relevant for a customer who already owns a card.
- **Suitability** determines whether an action or group of actions is appropriate for a customer for ethical or empathetic reasons. For example, a credit card offer might not be appropriate for a customer who is financially vulnerable, even though it might be profitable for the bank.
- **Contact Policies** determine when to suppress an action or group of actions and for how long. For example, you can suppress an action after the customer receives a

specific number of promotional messages.



Issue Group
Grow / Credit cards

› **All actions**

⌄ **Customer actions**

E **Eligibility** ⓘ

(Customer: Age is greater than or equal to 18)

A **Applicability** ⓘ

(Customer: Count of credit card accounts is equal to 0)

S **Suitability** ⓘ

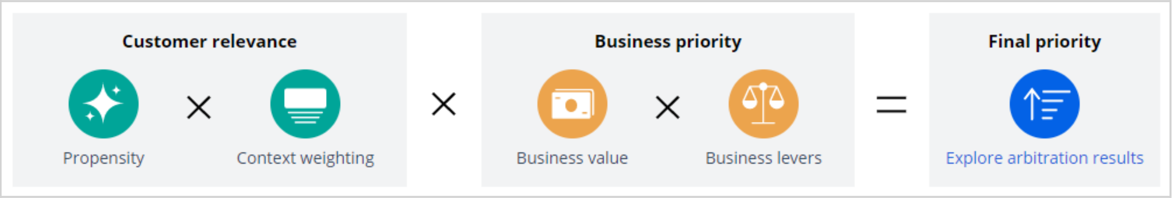
(Customer: Is financially vulnerable is false)

C **Contact policy** ⓘ

› **7-day group clicks: Track Clicks for all actions in the issue over the past 7 days**

› **7-day action impressions: Track Impressions for the action over the past 7 days**

Arbitration determines how Customer Decision Hub prioritizes the list of eligible and appropriate actions from each group.



The factors that the system weighs in arbitration are **Propensity**, **Context weighting**, **Business value**, and **Business levers**. Numerical values represent the factors.

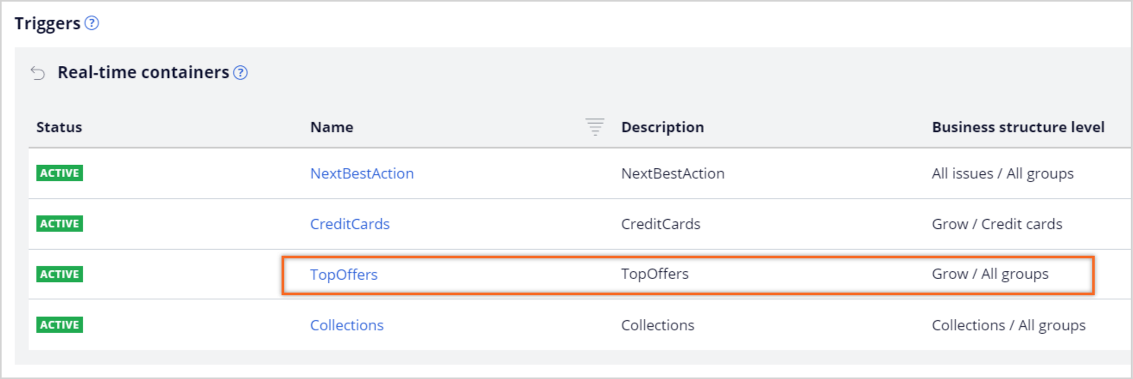
Propensity is the likelihood that a customer responds positively to an action, and AI calculates the propensity. For example, clicking an offer banner or accepting an offer in the contact center are considered positive behaviors.

Context weighting allows you to assign a weighting to a specific context value for all actions within an issue or group. For example, if a customer contacts the bank to change the home address, the weight of the **Service** context increases, and the highest priority action is to ensure that the system delivers the relevant service to the customer.

Business value enables you to assign a financial value to an action and prioritize high-value actions over low-value ones. For example, promoting an unlimited data plan might be more profitable for the company than a limited data plan.

Business levers enable you to accommodate urgent business priorities by specifying a weight for an action, group, or business issue. A simple formula determines a prioritization value, which is used to select the top actions.

Next-Best-Action Designer enables the delivery of next best actions through inbound, outbound, and paid channels. A trigger is a mechanism where an external channel, for example, a website, invokes the execution of a Next-Best-Action decisioning process for specific issues and groups. The result returns to the invoking channel. A real-time container is a placeholder for content in an external real-time channel. For example, a website invokes a real-time container, *TopOffers*, before loading the account landing page.

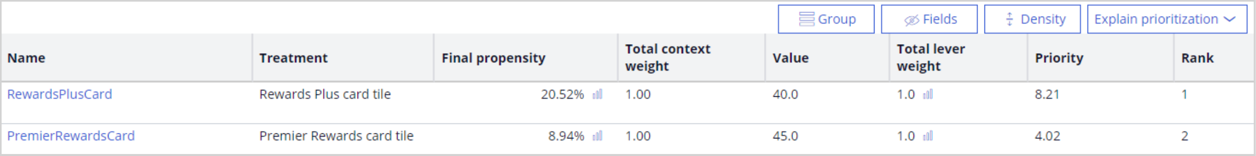


Status	Name	Description	Business structure level
ACTIVE	NextBestAction	NextBestAction	All issues / All groups
ACTIVE	CreditCards	CreditCards	Grow / Credit cards
ACTIVE	TopOffers	TopOffers	Grow / All groups
ACTIVE	Collections	Collections	Collections / All groups

You can explore the arbitration matrix in the Customer Profile Viewer report to understand what actions the system offers to customers. For example, U+ Bank uses Customer Decision Hub to decide which credit card offer to show when customers log in to the bank's website. Joanna is a U+ Bank customer.

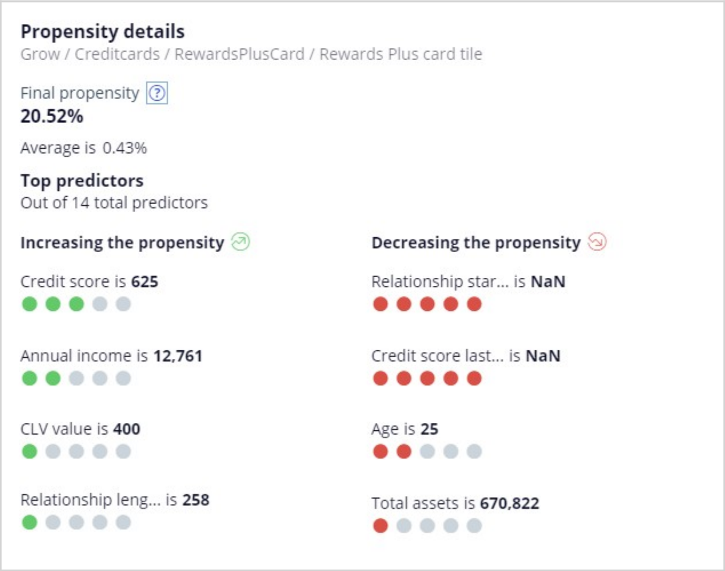
For this use case, the direction is inbound, the channel is the web, and TopOffers is the real-time container service that manages communication between Customer Decision Hub and the website. Joanna meets all criteria for two credit card offers: the Rewards Plus card, and the Premium Rewards card.

The **Explain prioritization** view shows the values for the arbitration formula components: propensity, context weight, business value, and lever weight. The business value of the Premier Rewards card is higher than the value of the Rewards Plus card. But the likelihood that Joanna is interested in a credit card offer is highest for the Rewards Plus card, which is the top offer.



Name	Treatment	Final propensity	Total context weight	Value	Total lever weight	Priority	Rank
RewardsPlusCard	Rewards Plus card tile	20.52%	1.00	40.0	1.0	8.21	1
PremierRewardsCard	Premier Rewards card tile	8.94%	1.00	45.0	1.0	4.02	2

To aid the transparency of the AI, you can view the predictors that are most influential in the calculation of the propensity.



Propensity details

Grow / Creditcards / RewardsPlusCard / Rewards Plus card tile

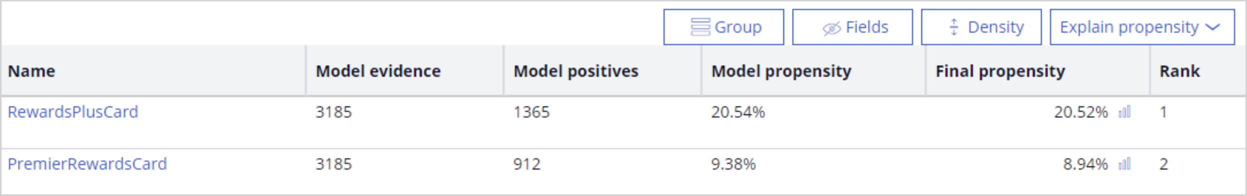
Final propensity
20.52%

Average is 0.43%

Top predictors
Out of 14 total predictors

Increasing the propensity	Decreasing the propensity
Credit score is **625** ●●●○○	Relationship star... is **NaN** ●●●●●
Annual income is **12,761** ●●○○○	Credit score last... is **NaN** ●●●●●
CLV value is **400** ●○○○○	Age is **25** ●●○○○
Relationship leng... is **258** ●○○○○	Total assets is **670,822** ●○○○○

To understand how the computation for the action propensity works, inspect the **Explain propensity** view, where you can see the model evidence, the number of positive responses, the initial model raw propensity and the final propensity that Customer Decision Hub uses in arbitration.



Name	Model evidence	Model positives	Model propensity	Final propensity	Rank
RewardsPlusCard	3185	1365	20.54%	20.52%	1
PremierRewardsCard	3185	912	9.38%	8.94%	2

You have reached the end of this video. You have learned:

- How Next-Best-Action Designer is organized according to the following high-level sequence of steps that are necessary to configure the next-best-action strategy:
  - Defining the business structure for your organization.
  - Implementing the channel limits and constraints.
  - Defining the contact policies that control for which actions a customer qualifies.

    - Configuring the prioritization of actions.
    - Configuring when and where to initiate the next best actions.
- How you can analyze the decisions for a specific customer in Customer Profile Viewer to aid model transparency.

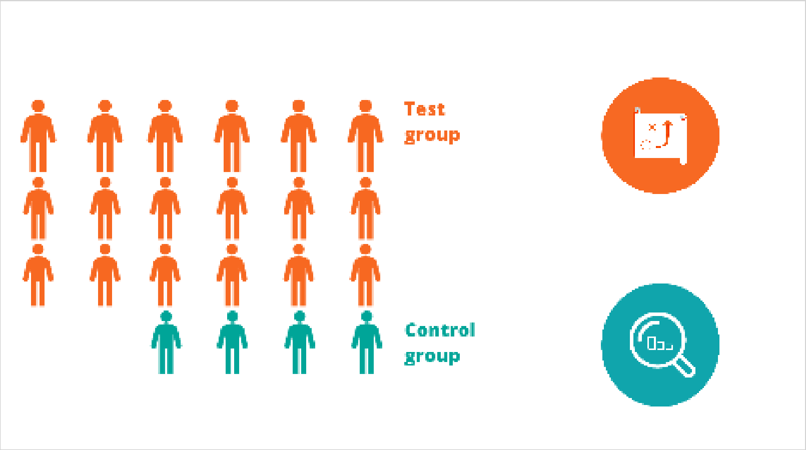
Monitoring the health of the system using Impact Analyzer

Impact Analyzer is a proactive monitoring tool that shows the overall health of your actions and highlights opportunities for improvement. The tool provides insight into the performance of your next best actions through the results of experiments that tell you what actions and configurations are working as expected, and where you can still improve.

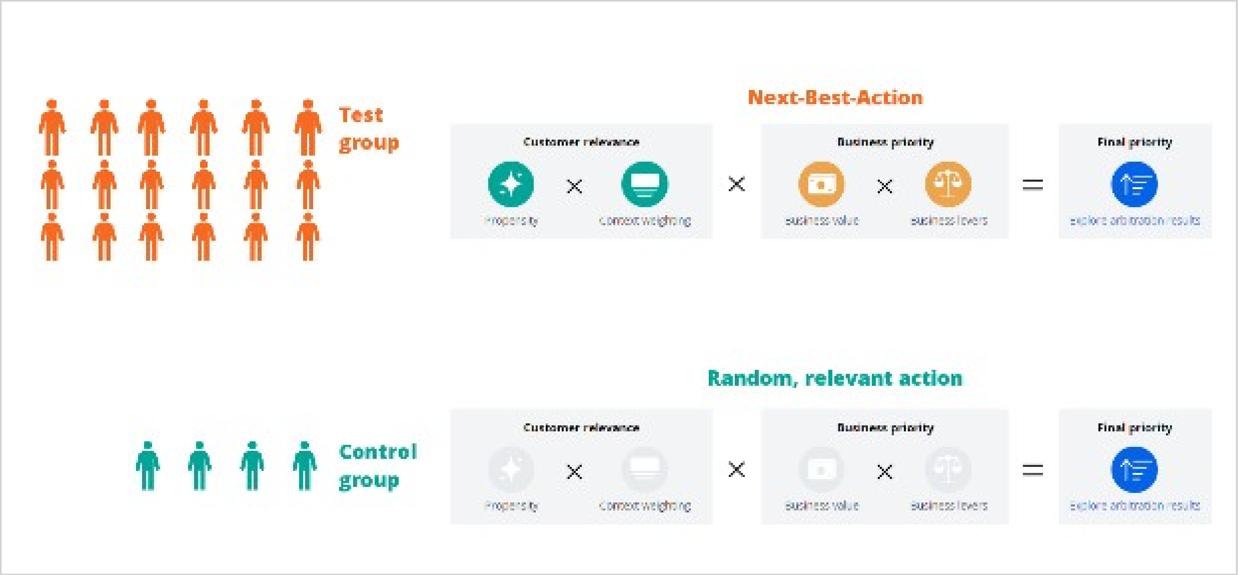
Transcript

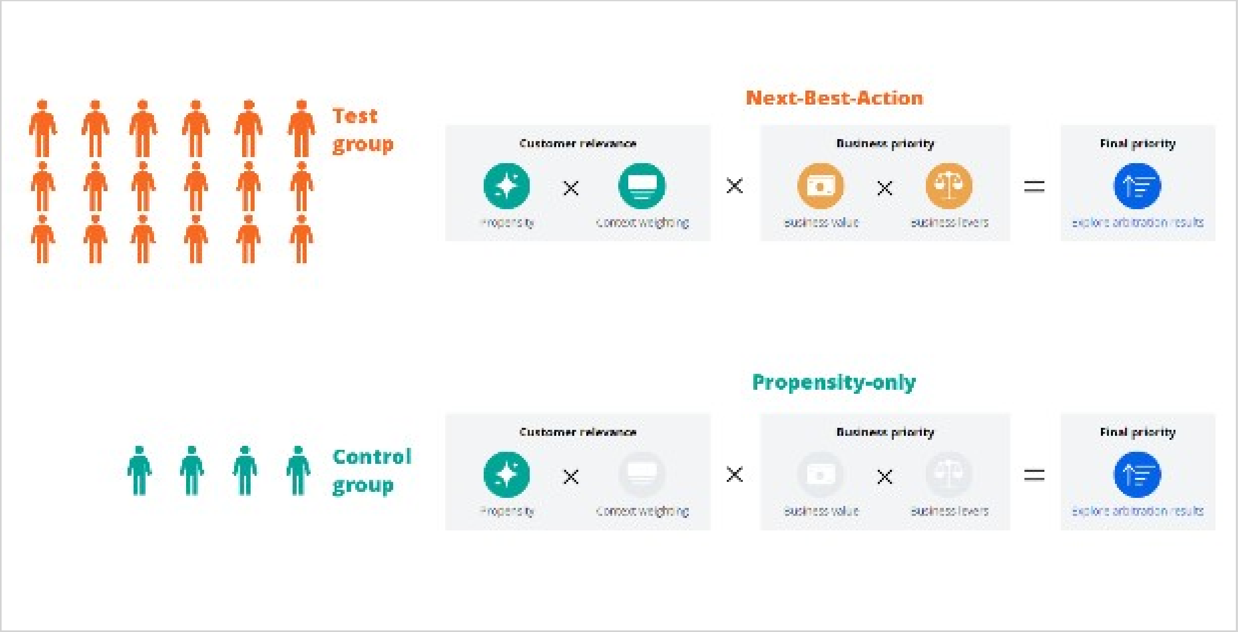
Impact Analyzer is a feature of Pega Customer Decision Hub™ that uses experimentation to track the effectiveness and overall health of next best actions. By utilizing this tool, you can gain insights into the effects of arbitration. Impact Analyzer conducts experiments with alternative prioritization and engagement policy filters on a small control group of customers. This enables you to assess whether or not the next best action is producing a positive effect. In addition to monitoring for basic lift, these control group experiments can also reveal opportunities to optimize and align the next best actions with your business goals.

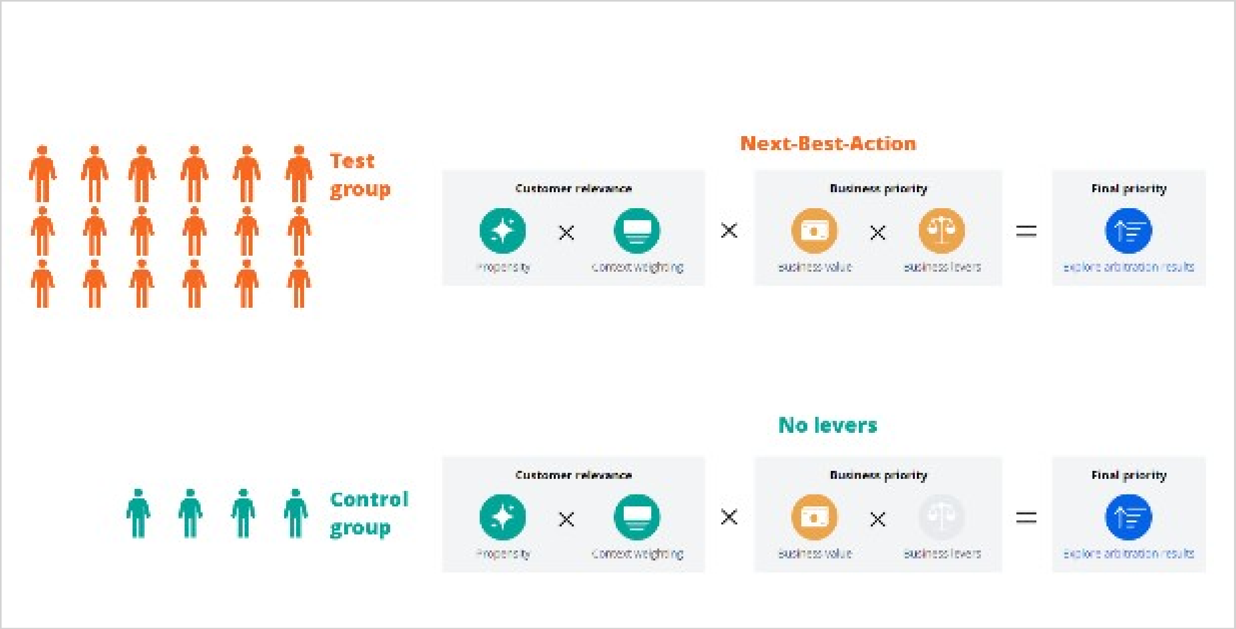
An experiment is a test with two sets of participants, the test group and the control group, in which the test group receives the originally arbitrated action and the control group receives an alternate action based on the type of experiment.

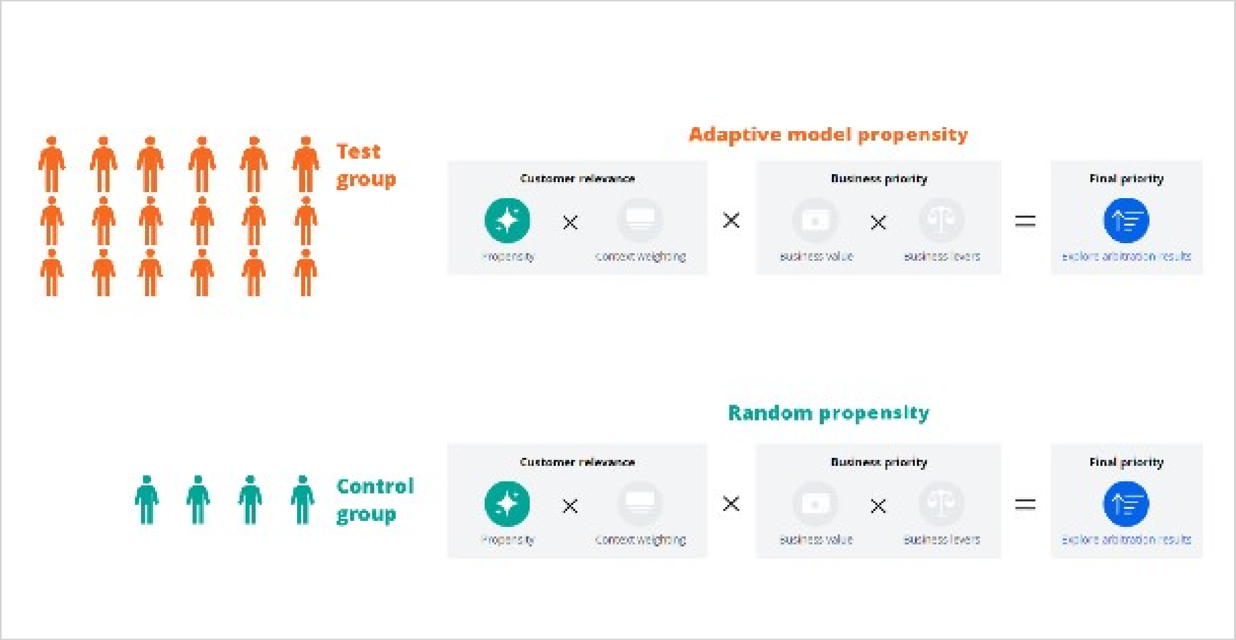


You can use Impact Analyzer to carry out a variety of tests. For example, you can assess the effectiveness of the next best action against a random relevant action or actions, chosen based on propensity alone. Additionally, you can measure the performance of the next best action when arbitrating with no levers. You can also evaluate the performance of the adaptive model propensity against a random propensity.


Test group
Next-Best-Action
Customer relevance
Propensity
×
Context weighting
×
Business priority
Business value
×
Business levers
=
Final priority
Explore arbitration results
Control group
Random, relevant action
Customer relevance
Propensity
×
Context weighting
×
Business priority
Business value
×
Business levers
=
Final priority
Explore arbitration results

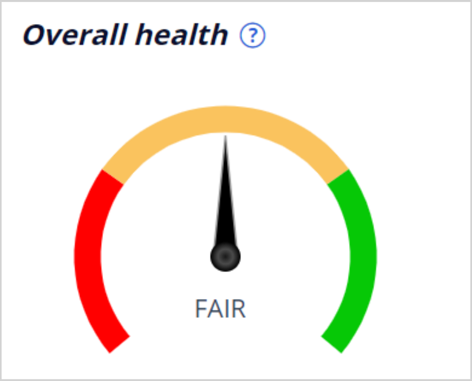

Test group
Next-Best-Action
Customer relevance
Propensity
×
Context weighting
×
Business priority
Business value
×
Business levers
=
Final priority
Explore arbitration results
Control group
Propensity-only
Customer relevance
Propensity
×
Context weighting
×
Business priority
Business value
×
Business levers
=
Final priority
Explore arbitration results





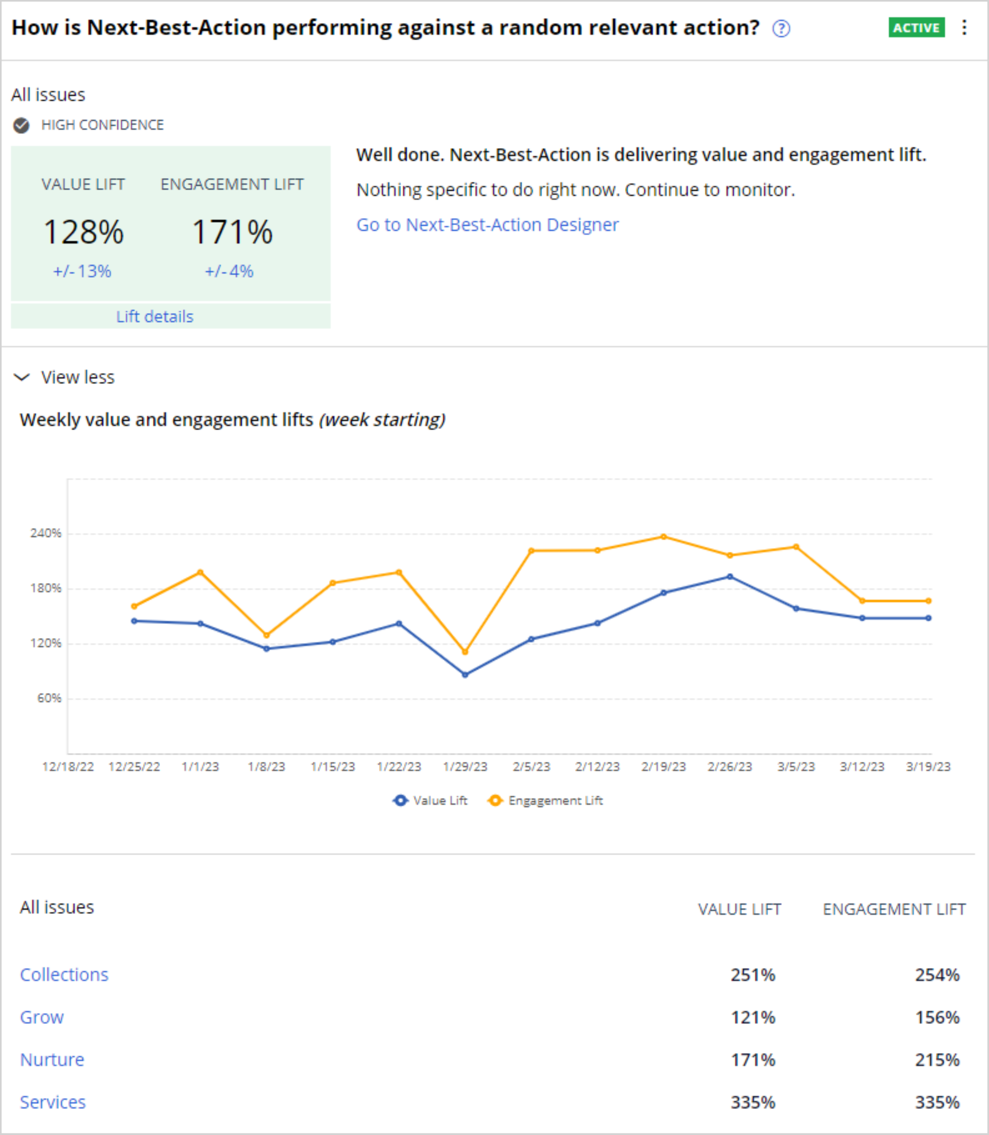
Impact analyzer continuously monitors the health of the next best actions in the experiment, and presents the results in the form of two key performance indicators (KPIs): value lift, and engagement lift. Value lift measures how much a business can make from suggesting certain next best actions to customers, compared to a random relevant action. Engagement lift measures how much more likely customers are to click or accept a suggested next best action, compared to the random relevant action.

When a next best action is delivering lift, the overall health indicator highlights in green. The middle position of the indicator shows that there is room for improvement to better align actions to the business. The red highlight [the section on the left] means that something is wrong and needs your attention. The indicator is colored gray when the general health is not available due to insufficient data in one or more active tests.



Impact Analyzer is available under **Discovery** tab in the left menu of Customer Decision Hub.

One of the experiments that you can run is called *How is Next-Best-Action performing against a random relevant action?*



How is Next-Best-Action performing against a random relevant action? ACTIVE

All issues

HIGH CONFIDENCE

VALUE LIFT	ENGAGEMENT LIFT
128%	171%
+/-13%	+/-4%

Lift details

Well done. Next-Best-Action is delivering value and engagement lift.

Nothing specific to do right now. Continue to monitor.

Go to Next-Best-Action Designer

View less

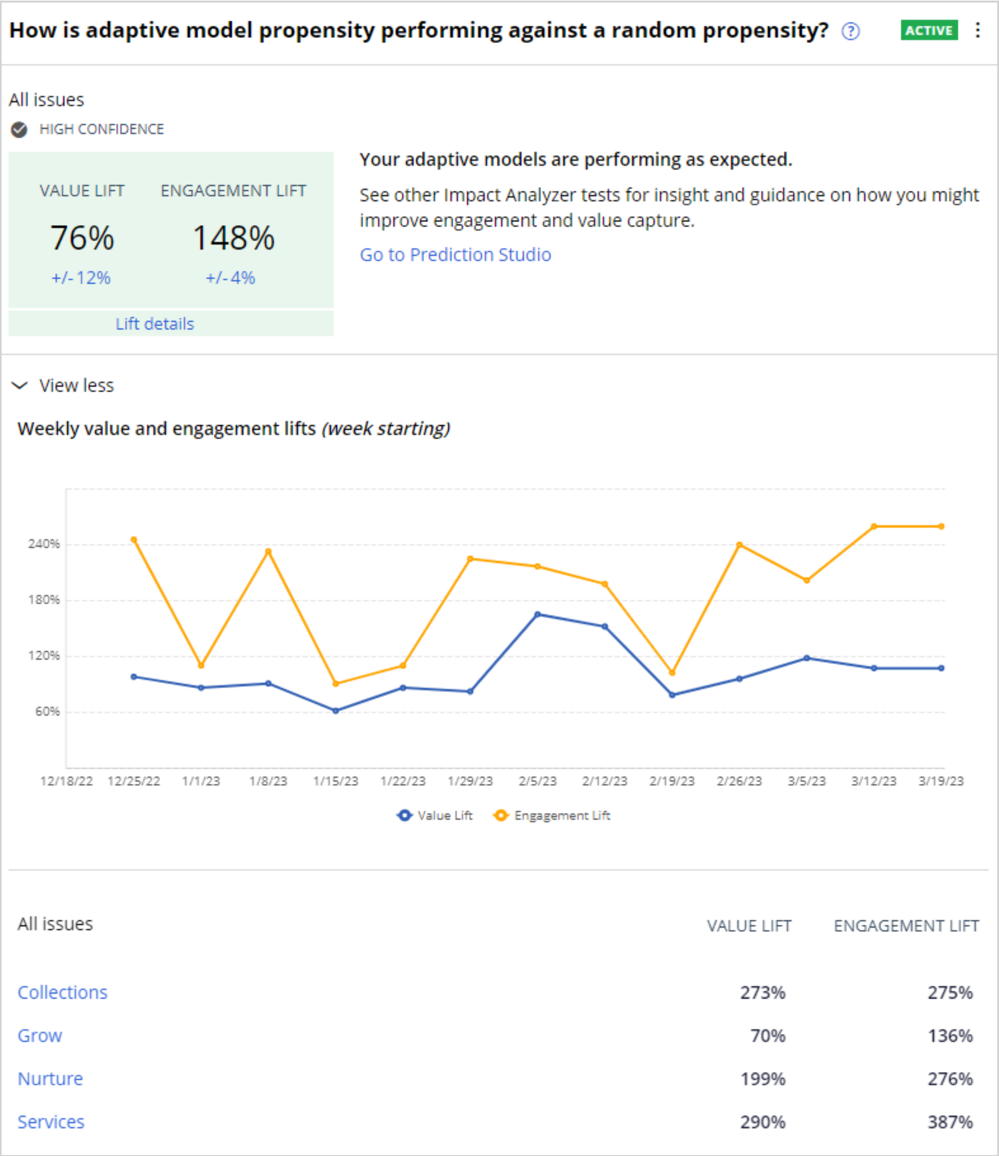
Weekly value and engagement lifts *(week starting)*

All issues	VALUE LIFT	ENGAGEMENT LIFT
Collections	251%	254%
Grow	121%	156%
Nurture	171%	215%
Services	335%	335%

This test serves as a baseline check to ensure that the next best action is beating the random relevant action. At the very least, we expect the value lift - and ideally, both the value and engagement lift - to be positive, at the highest level across all issues and groups aggregated. If both value and engagement lifts are negative, there is an issue. This could mean that there is a problem with the underlying adaptive models, or that your engagement policies are too targeted. Another reason might be that your levers are excessively biased, and the next best action lacks the aptitude to optimize engagement or to capture value.

The test widget presents a value lift of 128 percent and an engagement lift of 171 percent. Both values are positive, and the test is highlighted in green.

Another experiment is called *How is adaptive model propensity performing against a random propensity?*



How is adaptive model propensity performing against a random propensity? ACTIVE

All issues

HIGH CONFIDENCE

VALUE LIFT	ENGAGEMENT LIFT
76%	148%
+/- 12%	+/- 4%

Lift details

Your adaptive models are performing as expected.

See other Impact Analyzer tests for insight and guidance on how you might improve engagement and value capture.

Go to Prediction Studio

View less

Weekly value and engagement lifts *(week starting)*

All issues	VALUE LIFT	ENGAGEMENT LIFT
Collections	273%	275%
Grow	70%	136%
Nurture	199%	276%
Services	290%	387%

This experiment compares only Pega AI-driven model propensity with no context weight, no business value, and no business levers, against only randomly generated propensity. The experiment is another baseline check to ensure that the adaptive models that a next best action uses are delivering lift over random propensity.

To monitor for basic lift, both presented experiments should be on at all times.

Impact Analyzer offers a range of benefits to its users. One of the most significant benefits is the ability to collect data in real-time, allowing for real-time monitoring and testing of KPIs. Impact Analyzer also enables users to look for opportunities that they may have otherwise missed.

The primary user of Impact Analyzer is the Next Best Actions operations team, who are responsible for ensuring that the system runs optimally. However, the tool brings value to various lines of business, sales, marketing, and Data Scientists.

You have reached the end of this video. You have learned:

- How Impact Analyzer continuously monitors the performance and health of the system.
- How Impact Analyzer conducts experiments by implementing alternative prioritization and engagement policy filters on a small control group of customers, to assess the effectiveness of next best actions.
- How the experiments reveal opportunities to optimize and align next best actions with business goals.

Customer Decision Hub predictions

Description

Discover the Pega Customer Decision Hub™ predictions in the Prediction Studio portal, a comprehensive workspace for Data Scientists who manage customer interactions. Delve into predefined predictions for communication channels, adaptive models, and customer-level decisions to optimize customer experiences and effectively respond to changes in customer behavior.

Familiarize yourself with the tools and processes used to manage business changes in the Pega 1:1 Customer Engagement solution.

Learning objectives

- Describe how adaptive models drive the predefined Pega Customer Decision Hub™ predictions.
- Describe how the outcomes of supporting predictive models can function as predictors for the adaptive models.
- Describe the tasks of a Data Scientist in a Customer Decision Hub project.
- Describe the steps involved in managing business changes in one-to-one customer engagement.

Customer Decision Hub predictions

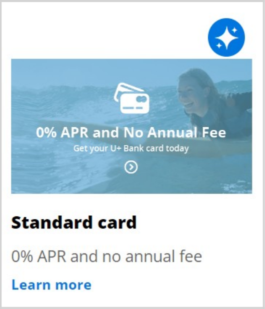
Prediction Studio is the workspace in which you manage the predictions that Pega Customer Decision Hub™ uses to optimize customer interactions.

Transcript

This video provides an overview of the Customer Decision Hub predictions in the Prediction Studio portal. Prediction Studio is the workspace for data scientists, offering tools to manage predictions used by Customer Decision Hub for optimizing customer interactions.

Decision strategies utilize predictions to determine the optimal action and treatment for customers across all interaction channels. Customer Decision Hub includes predefined predictions for each relevant communication channel. The predefined *Predict Inbound CallCenter Propensity* prediction calculates the likelihood of a customer accepting a proposition offered by a Customer Service Representative when contacting the call center. The *Outbound Email Propensity* prediction calculates the likelihood that a customer clicks a link in an email.

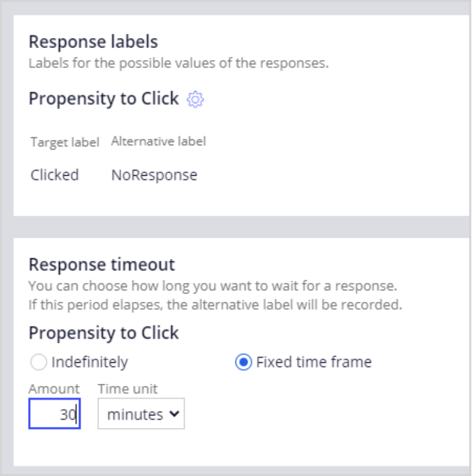
The prediction that calculates the propensity of a customer clicking on a web banner is named *Predict Web Propensity*. For instance, in a cross-sell scenario, U+Bank leverages its website as a marketing tool. Upon customer login, the website presents a personalized credit card offer in a banner.



If a customer is eligible for multiple credit cards, the *Predict Web Propensity* prediction calculates the propensity of receiving a positive response from the customer for each of these cards. Customer Decision Hub determines which credit card to offer based on business rules, interaction context, and propensity.

The context of the *Predict Web Propensity* prediction includes the treatment that a customer receives, in this case, a personalized banner in the web channel. This allows the capture of variations in customer preferences, such as the background color of a credit card offer in a web banner.

The prediction's response labels are “Clicked” for the target behavior and “NoResponse” for the alternative behavior. The NoResponse outcome is recorded after the response timeout expires.



Response labels

Labels for the possible values of the responses.

Propensity to Click

Target label	Alternative label
Clicked	NoResponse

Response timeout

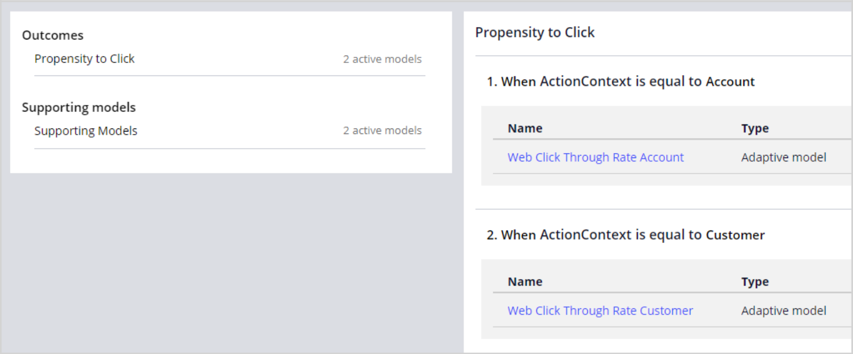
You can choose how long you want to wait for a response.
If this period elapses, the alternative label will be recorded.

Propensity to Click

○ Indefinitely ● Fixed time frame

Amount	Time unit
30	minutes

Predictive models drive predictions.



Outcomes

Propensity to Click	2 active models

Supporting models

Supporting Models	2 active models

Propensity to Click

1. When ActionContext is equal to Account

Name	Type
Web Click Through Rate Account	Adaptive model

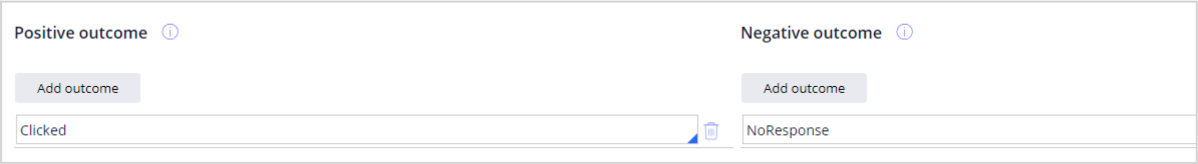
2. When ActionContext is equal to Customer

Name	Type
Web Click Through Rate Customer	Adaptive model

The propensity predictions integrated into Customer Decision Hub utilize adaptive models that learn from customer responses and receive automatic updates every hour. This ensures that U+Bank automatically responds to changes in customer behavior while the adaptive models self-optimize.

Customer Decision Hub makes customer-level decisions, while account-level decisions are made for a specific account associated with the customer. For example, a customer-level decision might be to send a promotional email to a customer based on the overall engagement with the company, while an account-level decision might be to send a reminder email to a customer based on their account balance.

The *Predict Web Propensity* prediction propagates the outcomes of a model based on the *Web Click Through Rate* adaptive model configuration. The outcomes of the adaptive model configurations align with the response labels of the prediction.



Positive outcome

Add outcome

Clicked

Negative outcome

Add outcome

NoResponse

Business requirements might necessitate the use of a predictive model trained on historical data or a highly transparent scorecard model. These models can be incorporated as supporting models within the prediction, with their outcomes serving as predictors for the adaptive models.

You have reached the end of this video. What did it demonstrate?

- The predictions that come with Customer Decision Hub in the Prediction Studio portal.
- The *Predict Web Propensity* prediction utilizes the *Web Click Through Rate* adaptive model configuration to determine the outcomes.
- The outcomes of supporting predictive models can function as predictors for the adaptive models.

Customizing Prediction Studio

Prediction Studio is the workspace that provides tools to create, monitor and update predictions and the predictive models that drive them. Learn how to customize Prediction Studio by configuring the notifications that alert the data scientist team that attention is required.

Transcript

Hi! I'm Kevin, a Data Scientist who's new to Pega. I've been assigned to a Customer Decision Hub implementation project. I'm going to reach out to Bella, the Lead System Architect on the project, to get some recommendations on customizing Prediction Studio.

Hi Kevin, thrilled to have you on board!

Let's kick off by configuring the notifications that keep the data scientist team in the loop when attention is needed. Bella, can you tell me more about the notifications generated by Prediction Studio?

Well, Prediction Studio generates a variety of notifications that offer insights into the performance of your predictions, adaptive models, and predictive models. You can access these notifications by clicking on the Notifications icon in the header of Prediction Studio. These notifications are valuable for monitoring your models, updating their configuration if necessary, and enhancing the predictions that drive Customer Decision Hub decisions. If you want to modify notification settings, you can navigate to the Monitor & notification settings section in Prediction Studio.

Got it. So, just to clarify, I have to keep logging into Prediction Studio to read the notifications, right?

No, you can also receive daily email summaries that highlight high-impact notifications. So, you don't need to access Prediction Studio unless the notifications prompt you to do so.

That's great! I can stay informed without constantly logging in.

Now, I noticed there are different notification categories. In your opinion, which notifications in the Performance category are the most important?

Some of the crucial notifications in this category include sudden drops in predictive performance, models with low performance, and changes in lift or accuracy. The additional notifications you enable will depend on the specific use case.

I can imagine a flood of notifications coming our way. Is there any way to limit the number of notifications to avoid information overload?

Certain notifications have customizable thresholds that you can adjust according to your preferences. This allows you to define when you want to receive specific types of notifications. By customizing the thresholds, you can receive notifications at a frequency that suits your needs.

So, just to summarize, when implementing a new Customer Decision Hub project, the data scientist team configures the notifications that alert them to important issues impacting the models and predictions in the system.

Absolutely. These notifications help the team promptly address critical matters.

Thanks a lot, Bella! That was incredibly insightful!

Change management process

Introduction

When business requirements change rapidly, the software development process needs to be more agile, while still producing high-quality, reliable software. This topic covers the importance of the change management process, and how it works in a one-to-one customer engagement project for Pega Cloud® customers.

Transcript

This video describes the change management process in a one-to-one customer engagement project and the Pega cloud environments that support the process.

Let’s start by understanding some background on change management in enterprise software development projects.

In recent decades, organizations have been using computer software to automate many traditionally manual tasks.

Business processes need to constantly evolve as customer behavior and market conditions change. Naturally, the software that supports business practices needs to evolve as well. As the pace of change in business requirements increases, the software development process needs to be more agile, while still producing high-quality and reliable software.

Here’s a simplified view of an enterprise software development cycle. It consists of four high-level stages.

Developers develop new software or update existing software.

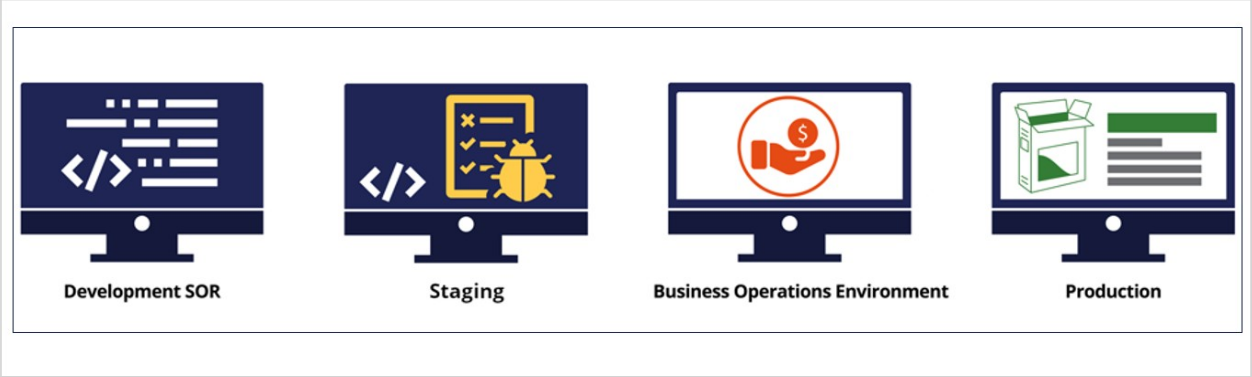
The work from several developers is merged into a single system in the integration phase.

The new software version goes through testing, and the final, approved software (or a software change) is deployed into production.

This cycle repeats for new as well as incremental updates to existing software.

A software development process is supported by different environments.

Let’s study the environments available to Pega Cloud® customers in support of a one-to-one customer engagement project using Pega Customer Decision Hub™.



- A development environment is one in which developers create new versions of the application by adding enhancements or fixing issues. It is referred to as system of record (SOR) for Pega applications.

- A staging environment is used for various testing such as functional testing, unit testing, and user acceptance testing.

- The business operations environment (BOE) is a replica of the production environment. However, it contains only a sample of the production data. This is where the business operations team creates and tests new business artifacts and conducts simulations.

- The production environment is the main system that propagates the next best actions to external channels, collects customer responses, and is where the AI learning happens. It is also used for live monitoring of key performance indicators.

In a one-to-one customer engagement project, changes to the application can be classified into two categories: enterprise changes and business changes.

Enterprise changes are the changes that developers make to the Pega application. An example of enterprise changes are extensions to the core Pega application and its integration points with external systems.

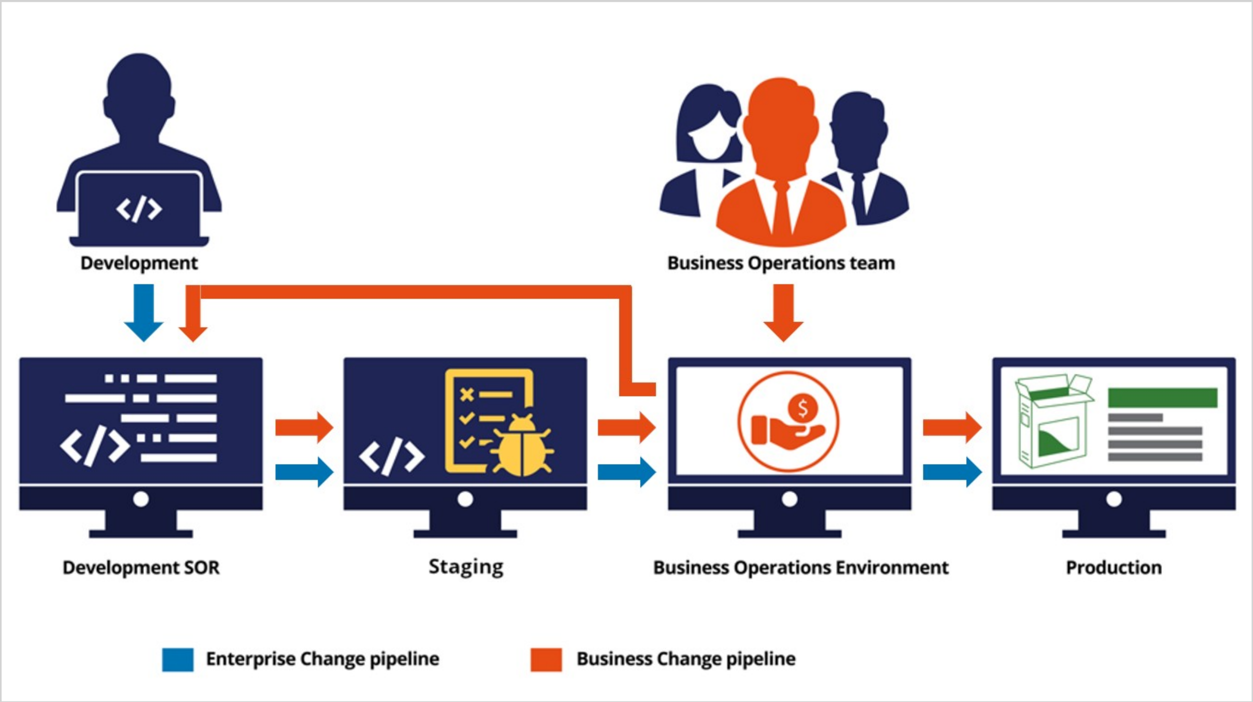
Developers make these changes in the development environment. Changes to the application are pushed to other environments through the enterprise-change pipeline managed by the Pega Deployment Manager™.

Examples of business changes include creating a new action or updating an existing action with new treatments or engagement policies. Also, this environment is used to carry out various simulations and analyses, for example, to test if there is an ethical bias in the decisions made by the next-best-action strategy framework. Business changes are made by the business operations team in the business operations environment.

The business content team uses the 1:1 Operations Manager portal to initiate changes in the business operations environment.

In the BOE, the changes are carried out in an application overlay. The application overlay defines the scope in which business users can change the application (for example, by modifying actions, treatments, decision strategies, and so on) to accommodate the constantly changing business conditions and requirements.

Changes from the business operations environment are pushed to the development environment and from there to other environments through the business change pipeline.


Development
Business Operations team
Development SOR
Staging
Business Operations Environment
Production
🟦 Enterprise Change pipeline
🟧 Business Change pipeline

You have reached the end of this video which showed you:

- The importance of the change management process in an enterprise software development project.
- The high-level software development cycle.
- The cloud environments provided by Pega for a one-to-one customer engagement project using Pega Customer Decision Hub.
- The flow of enterprise and business changes through the enterprise and business change pipelines.

Adaptive models

Description

Online, adaptive models are crucial to next-best-action decision strategies in Pega Customer Decision Hub™. Adaptive Decision Manager is a key component of Pega Decision Management, and helps organizations deliver tailored experiences that balance customer relevance and business priorities.

Discover the crucial role of input fields as predictor data in enhancing the predictive performance of adaptive models. Adaptive models automatically optimize by selecting the best subset of predictors from various relevant data sources.

Learn how to safely monitor a new adaptive model configuration by shadowing an active model.

Learning objectives

- Explain how adaptive analytics supports Customer Decision Hub in the selection of the next best action for each customer.
- Describe how ADM generates Bayesian models.
- Explain how arbitration works.
- Request a decision for a customer in Customer Profile Viewer.
- Explore the original model propensity and model evidence.
- Configure additional potential predictors for an adaptive model.
- Shadow an adaptive model to safely monitor a new adaptive model configuration.

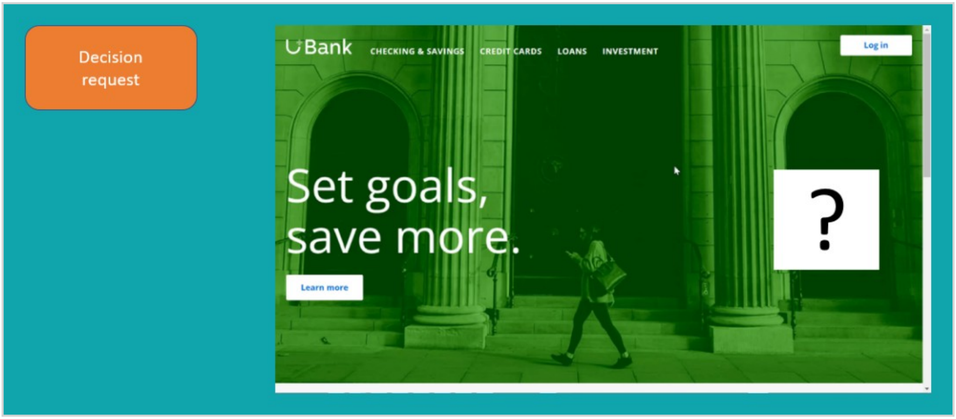
Adaptive analytics

Online, adaptive models play a crucial part in Pega's next-best-action decision strategies as they are used to predict a customer's propensity for all available actions. Adaptive models are an important element in providing highly personalized and relevant actions to each individual customer - helping brands achieve the goal of true 1:1 customer engagement. Pega Adaptive Decision Manager (ADM) provides a full set of capabilities in Prediction Studio that data scientist can make use of to create, train, and manage their self-learning models.

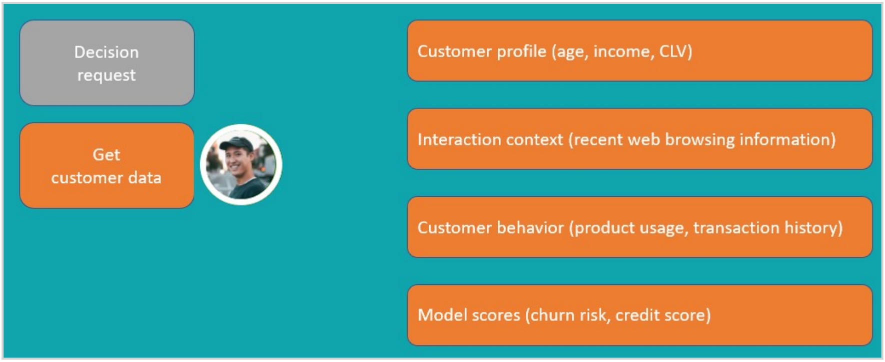
Transcript

This video shows you how adaptive analytics supports Pega Customer Decision Hub™ in the selection of the next best action to take for each customer.

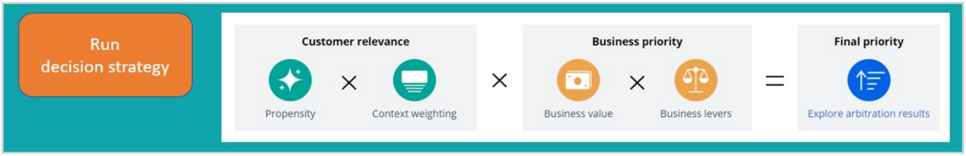
Adaptive Decision Manager (ADM) is a component of Pega Decision Management that businesses can use to implement online, adaptive models that drive predictions about customer behavior, like clicking or ignoring a web banner that offers a credit card on a bank’s website. When customer Troy logs in to the U+ Bank website, a decision request is sent to a client node of Pega Platform.



ADM retrieves all available customer data, which may include the customer profile, the interaction context, past customer behavior, and model scores.


Decision request
Get customer data
Customer profile (age, income, CLV)
Interaction context (recent web browsing information)
Customer behavior (product usage, transaction history)
Model scores (churn risk, credit score)

Using this data as input, the system runs a decision strategy to determine which credit card is the best offer for Troy, balancing customer relevance and business priority.



The result of the decision strategy, the Next-Best-Action for customer Troy, is the Standard Card, which is then displayed on the website. There are two possible outcomes of the interaction.

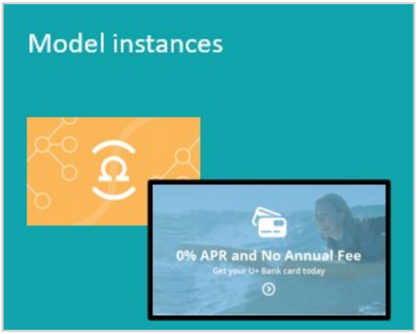
If Troy is interested and clicks on the web banner, ADM records the outcome as target behavior. If he ignores the banner, ADM records the outcome as alternative behavior. The prediction predicts the probability that a customer shows the target behavior.

The adaptive models that drive the widely used predictions that ship with Customer Decision Hub™ use a Bayesian algorithm. The prediction that calculates the propensity that a customer will click on the web banner is the Predict Web Propensity prediction. The adaptive model rule that drives this prediction is the Web Click Through Rate model.

Predict Web Propensity prediction
Web Click Through Rate adaptive model rule

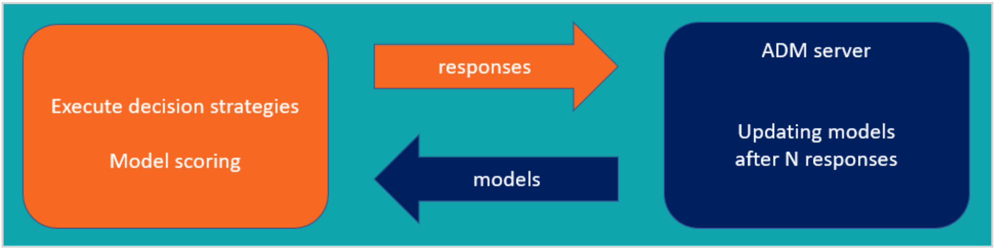
A data scientist configures the settings of both the prediction and the adaptive model rule, including the customer fields that are available as features to the model rule. Customer fields that are unsuitable as features, for example the customer ID, should be excluded.

An adaptive model rule typically generates many adaptive model instances without human intervention, because each unique combination of an action, treatment, direction, and channel, will generate a model the first time a decision strategy runs that references the model.



Model instances

ADM captures the responses and updates the adaptive model instances regularly, so they continuously learn from customer responses and adapt to account for changing customer interests and needs. The ADM server is physically separated from the nodes that process decision strategies and model executions, so that the laborious process of updating models does not impact decisioning speed.



Execute decision strategies

Model scoring

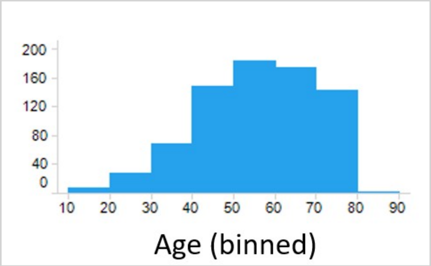
responses

models

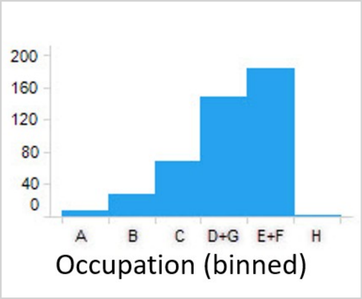
ADM server

Updating models after N responses

The Bayesian algorithm that generates and updates the model instances consists of 4 steps: preprocessing, feature selection, scoring, and transformation of scores to propensities. Preprocessing involves binning of the predictor values. For numeric predictors, ADM creates intervals with similar behavior. Customers aged 42 and aged 43 may have similar propensities to show target behavior and, after binning, reside in the same interval.

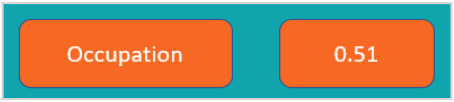


For symbolic predictors, ADM groups values with similar customer behavior.

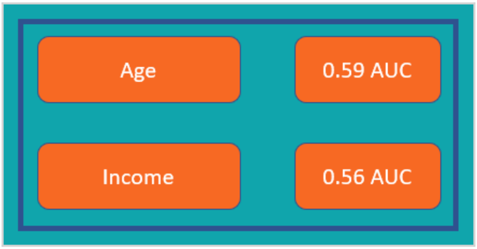


The granularity of the binning is a trade-off between performance and the statistical robustness of the predictor. Appropriate default settings for binning are provided and should only be changed by an experienced data scientist for specific use cases.

Next, ADM selects features based on their individual univariate performance against the outcome, measured as the area under the curve (AUC) of an ROC graph. By default, the univariate performance threshold is set to 0.52 AUC. A value of 0.5 represents no performance of a predictor, and the default threshold will exclude only features with a very low performance.



Additionally, ADM groups predictors that are highly correlated and then selects the best predictor from each group, to reduce unwanted complexity.



Next, a Naïve Bayes calculation is executed for the model using all selected predictors.

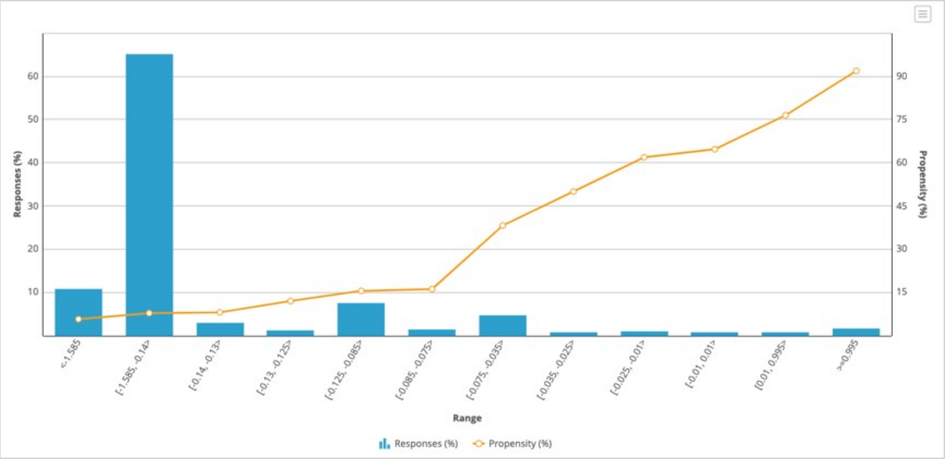
This simple and scalable calculation is based on Bayes' theorem, which says that the probability of A, if B is true is equal to the probability of B, if A is true, times the probability of A being true, divided by the probability of B being true.

$$P(A|B) = (P(B|A) \times P(A))/P(B)$$

Naïve Bayes relies on the assumption that the predictors are independent. The grouping of correlated predictors in the previous feature selection step minimizes the uncertainty introduced by this assumption. The ADM algorithm uses the posteriori log odds - that is, the logarithm of the posterior probability of target behavior divided by one minus this probability - as the score.

Score (Log odds) = Ln(P/(1-P))

The final postprocessing step transforms the raw Naïve Bayes scores to true propensities. The algorithm creates score intervals in such a way that the propensity for each next bin always increases to optimize the accuracy of the models.



This video has concluded. What did it show you?

- How adaptive analytics supports Pega Customer Decision Hub in the selection of the next best action for each customer.
- How ADM generates Bayesian models.

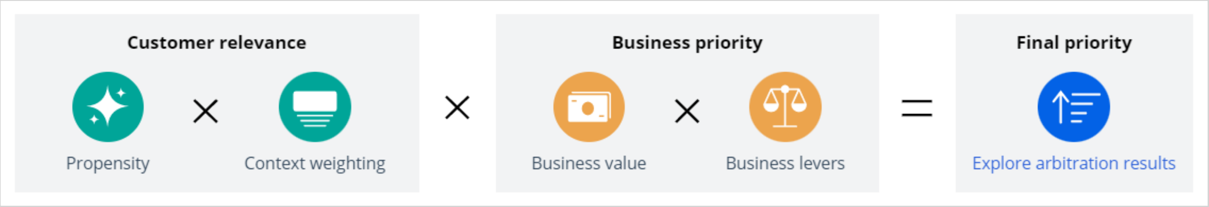
Action arbitration with AI

U+Bank wants to use Pega Customer Decision Hub™ to show a personalized credit card offer in a web banner when a customer logs in to their website. Customer Decision Hub uses AI to arbitrate between the offers for which a customer is eligible.

Transcript

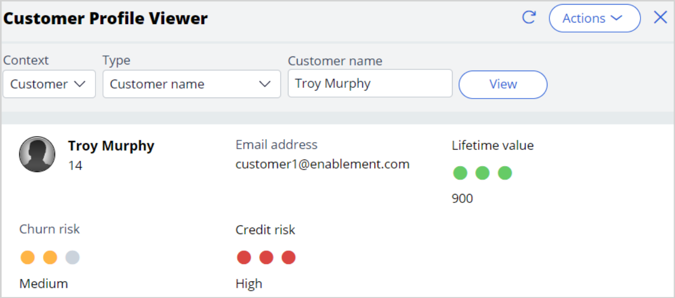
U+ Bank wants to optimize the cross-selling of their credit cards on the web by using Pega Customer Decision Hub to show a credit card offer in a web banner when a customer logs in to their account. Customer Decision Hub uses the Predict Web Propensity prediction to arbitrate between the offers for which a customer is eligible. This demo explores how AI-based arbitration works and the advanced settings of the Predict Web Propensity prediction.

In Customer Decision Hub, the Next-Best-Action Designer contains the arbitration settings.



Arbitration aims to balance customer relevance with business priorities to decide which offer to show to the customer. To achieve this balance, the system multiplies the numerical values that represent propensity, context weighting, business value, and business levers to arrive at a prioritization value, which determines the top actions. Propensity is the predicted likelihood that a customer shows the target behavior, in this case, clicking a web banner.

In Customer Profile Viewer, you can examine the next best actions for the customer Troy.



Customer Profile Viewer

Context: Customer
Type: Customer name
Customer name: Troy Murphy

Troy Murphy
14

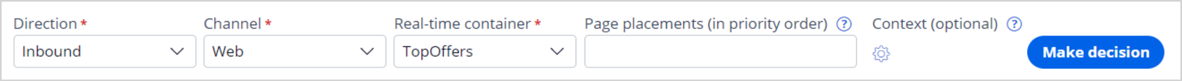
Email address: customer1@enablement.com

Lifetime value: ●●● 900

Churn risk: ●●○ Medium

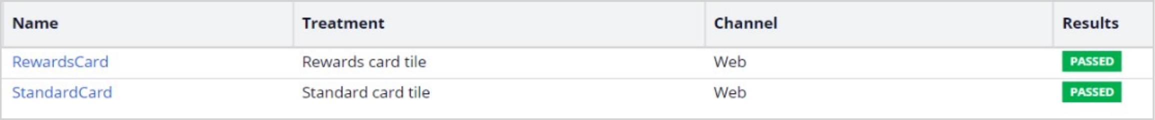
Credit risk: ●●● High

For the current use case, the direction is **Inbound**, and the channel is the **Web**. *TopOffers* is the real-time container service that manages communication between Customer Decision Hub and the website of the bank.



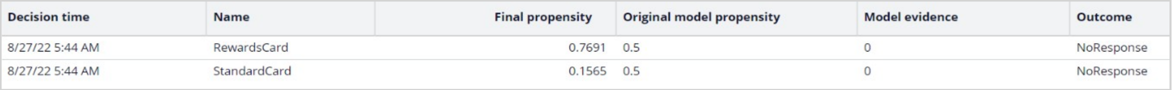
Direction *	Channel *	Real-time container *	Page placements (in priority order)	Context (optional)
Inbound	Web	TopOffers		

When you request a decision for Troy, the Customer Profile Viewer shows you the offers for which Troy is eligible. Based on the engagement policy rules, Troy is eligible for two credit card offers: the Rewards Card and the Standard Card.



Name	Treatment	Channel	Results
RewardsCard	Rewards card tile	Web	PASSED
StandardCard	Standard card tile	Web	PASSED

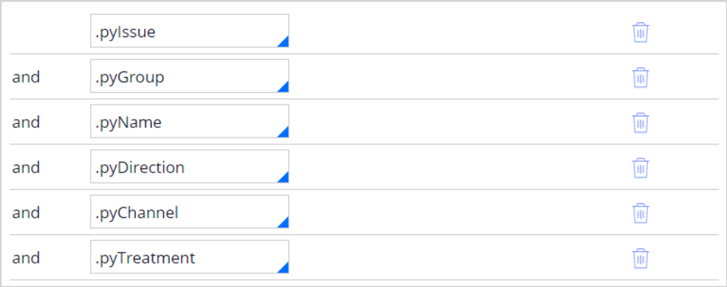
Initially, the model evidence is zero because the system has not yet captured any responses.



Decision time	Name	Final propensity	Original model propensity	Model evidence	Outcome
8/27/22 5:44 AM	RewardsCard	0.7691	0.5	0	NoResponse
8/27/22 5:44 AM	StandardCard	0.1565	0.5	0	NoResponse

With zero evidence, the original model propensity is 0.5, or the flip of a coin. The final propensity that the system uses in the prioritization formula deviates from the original model propensity because it depends not only on the original model propensity but also on a mechanism that introduces noise while the evidence is low. The noise decreases while the model learns from the target and alternative responses, and the original model propensity and the final propensity converge. This mechanism assures that new actions receive exposure even when their models are still immature.

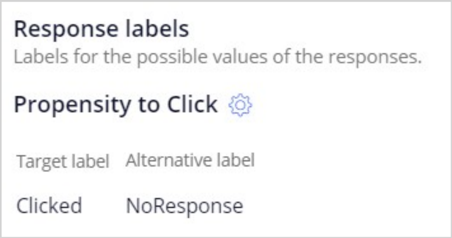
U+Bank uses the *Predict Web Propensity* prediction that comes with Customer Decision Hub out of the box to calculate the propensities. The *Predict Web Propensity* prediction calculates the final propensities for each combination of action and treatment in the inbound web channel.



	.pyIssue
and	.pyGroup
and	.pyName
and	.pyDirection
and	.pyChannel
and	.pyTreatment

A control group field determines a small percentage of customers that receive a random offer. During the production phase of the project, you can determine the impact of AI on the business by comparing the success rate of the offers that is based on AI, and the control group offers.

The target response has a **Clicked** label by default. For the alternative response, the label is **NoResponse**.



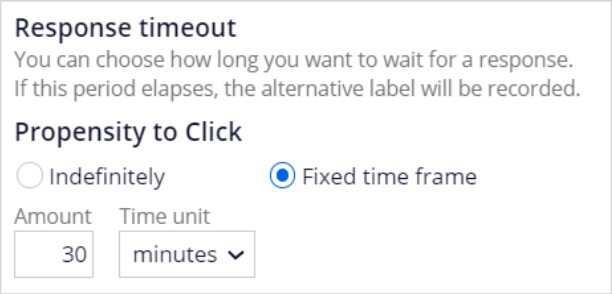
Response labels

Labels for the possible values of the responses.

Propensity to Click

Target label	Alternative label
Clicked	NoResponse

The **Response timeout** setting determines how long the system waits for a response from the customer after the impression. In a web scenario, the response timeout is 30 minutes by default, but an outbound channel requires a response timeout of several days to provide customers with enough time to respond to the message.



Response timeout

You can choose how long you want to wait for a response. If this period elapses, the alternative label will be recorded.

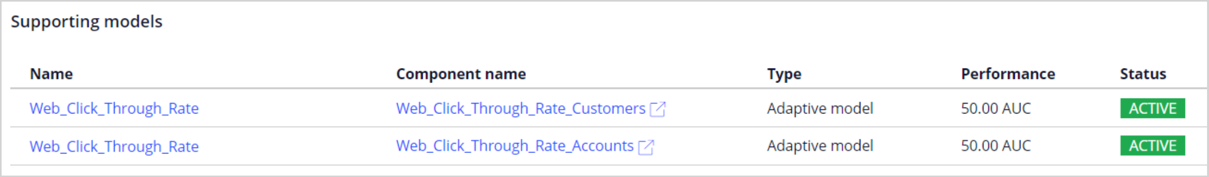
Propensity to Click

○ Indefinitely ● Fixed time frame

Amount: 30

Time unit: minutes

The *Web Click Through Rate* adaptive model configuration drives the prediction and supports decisions on both the customer and the account level.

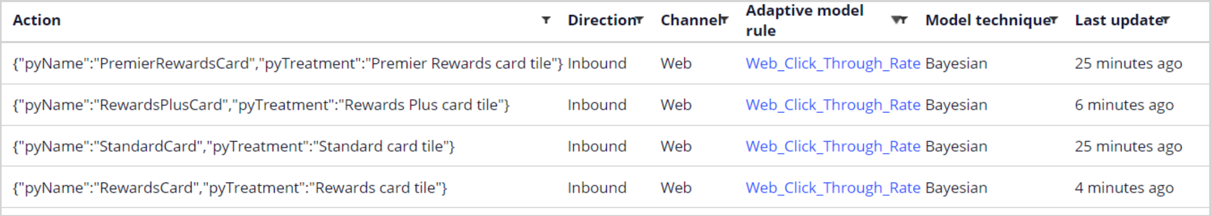


Supporting models

Name	Component name	Type	Performance	Status
Web_Click_Through_Rate	Web_Click_Through_Rate_Customers	Adaptive model	50.00 AUC	ACTIVE
Web_Click_Through_Rate	Web_Click_Through_Rate_Accounts	Adaptive model	50.00 AUC	ACTIVE

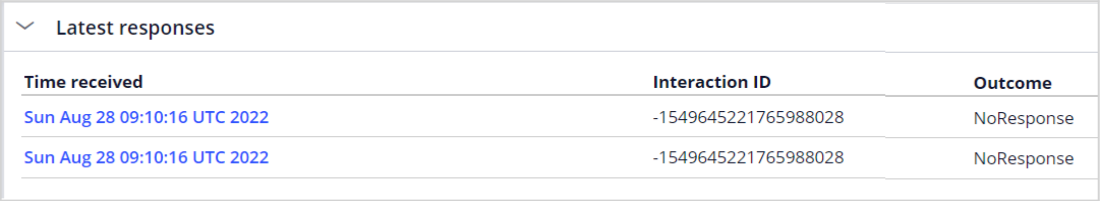
For each credit card offer and treatment that the customer is eligible for, an adaptive model based on the *Web Click Through Rate* configuration calculates the likelihood that the customer clicks the banner.

Every hour, Adaptive Decision Manager updates the models to learn from the recent interactions.



Action	Direction	Channel	Adaptive model rule	Model technique	Last update
{"pyName":"PremierRewardsCard","pyTreatment":"Premier Rewards card tile"}	Inbound	Web	Web_Click_Through_Rate	Bayesian	25 minutes ago
{"pyName":"RewardsPlusCard","pyTreatment":"Rewards Plus card tile"}	Inbound	Web	Web_Click_Through_Rate	Bayesian	6 minutes ago
{"pyName":"StandardCard","pyTreatment":"Standard card tile"}	Inbound	Web	Web_Click_Through_Rate	Bayesian	25 minutes ago
{"pyName":"RewardsCard","pyTreatment":"Rewards card tile"}	Inbound	Web	Web_Click_Through_Rate	Bayesian	4 minutes ago

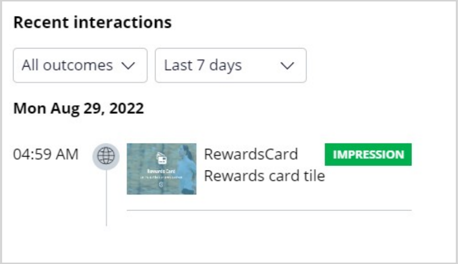
The outcome of both decision requests for Troy is *NoResponse*.



Latest responses

Time received	Interaction ID	Outcome
Sun Aug 28 09:10:16 UTC 2022	-1549645221765988028	NoResponse
Sun Aug 28 09:10:16 UTC 2022	-1549645221765988028	NoResponse

When customer Troy logs into the website, Customer Decision Hub displays the Rewards Card offer, and Adaptive Decision Manager records an impression.



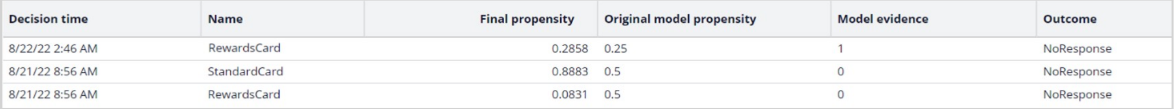
Recent interactions

All outcomes ⌄ Last 7 days ⌄

Mon Aug 29, 2022

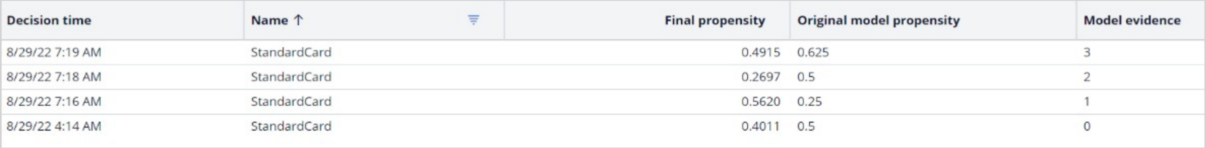
04:59 AM RewardsCard IMPRESSION
Rewards card tile

If Troy ignores the banner, Adaptive Decision Manager records *NoResponse* as the outcome after the response timeout elapses, in this case, 30 minutes. After the next update, the model calculates a lower propensity for Troy, as the new evidence that Troy is not interested in the offer weighs in.



Decision time	Name	Final propensity	Original model propensity	Model evidence	Outcome
8/22/22 2:46 AM	RewardsCard	0.2858	0.25	1	NoResponse
8/21/22 8:56 AM	StandardCard	0.8883	0.5	0	NoResponse
8/21/22 8:56 AM	RewardsCard	0.0831	0.5	0	NoResponse

Adaptive Decision Manager records a target response if Troy is interested in the Standard Card offer and clicks on the banner. After an update of the model, the propensity goes up.



Decision time	Name ↑	Final propensity	Original model propensity	Model evidence
8/29/22 7:19 AM	StandardCard	0.4915	0.625	3
8/29/22 7:18 AM	StandardCard	0.2697	0.5	2
8/29/22 7:16 AM	StandardCard	0.5620	0.25	1
8/29/22 4:14 AM	StandardCard	0.4011	0.5	0

You have reached the end of this video. What did it show you?

- How arbitration works.
- How to request a decision for a customer in Customer profile viewer.
- How to explore the original model propensity and model evidence.

Predictors of adaptive models

In a Pega Customer Decision Hub™ project, the input fields you select as predictor data for adaptive models play a crucial role in the predictive performance of those models. A model's predictive power is at its highest when you include many relevant data sources. Adaptive models automatically select the best subset of predictors.

Transcript

Hi! I'm Kevin, a Data Scientist new to Pega and assigned to a Customer Decision Hub implementation project in financial services.

I will ask Bella, the Lead Decisioning Architect on the project, for recommendations on predictors that adaptive models use to calculate the likelihood of a positive outcome.

Hi Kevin! Nice to see you again!

So, Bella, could you guide me on how to select the predictors for the adaptive models?

A good approach is to start with data that is already in the application and then develop a roadmap to gradually incorporate and derive more data over time.

Keep in mind that when we talk about selecting predictors, we're actually talking about selecting predictor candidates.

You can have a wide range of candidate predictors at your disposal, sometimes even several hundred or more.

During the production phase of the project, Adaptive Decision Manager automatically identifies the best subset of these candidates to become active in the models.

It also groups correlated predictors together and selects the one with the strongest relationship to the outcome.

So I can throw in any data that I can get my hands on?

No, it is crucial to comply with all internal policy, legal, and regulatory requirements.

Also, you should avoid using customer identifiers as predictors for adaptive models, as they typically have a unique value for each customer and are not predictive.

Got it! So, where can I see what data sources are available in the application?

In the Customer Decision Hub portal, you can find a list of all the available data sources on the Profile Data Sources landing page.

Clear! So, do you have any general recommendations on the data that I should use?

Yes, I do. For the best results, it's advisable to use predictors that provide data from various data sources.

Start by incorporating customer profile information such as age, gender, and customer lifetime value.

Additionally, consider adding relevant fields like scores from externally generated predictive models as predictor candidates, and account attributes if you have account-level actions.

That is all static data. I see that event data sources cannot be associated with the customer context.

Correct. You need to summarize event data before you can use these summaries as predictors.

Event data contains information on past customer behavior.

The strongest predictors to predict future behavior typically contain data about past behavior.

How about the Financial services clickstream summary?

The Financial services clickstream is a summary included in the Customer Profile Designer component and used for aggregating clickstreams on financial services-based web pages.

Recent web browsing information can be highly relevant and, therefore, very predictive.

I'm currently working on the Predict Web Propensity prediction, that calculates the likelihood a customer clicks on a web banner.

Can you show me how to add candidate predictors to enhance its predictive power?

Sure! The Predict Web Propensity prediction uses the Web Click Through Rate adaptive model configuration.

There are three types of predictors based on data sources: regular predictors, parameterized predictors, and predictors based on interaction history summaries.

Regular predictors are input fields that are available on the primary page where the Adaptive Model rule is defined.

Parameterized predictors are input fields that are not available on the primary page where the adaptive model rule is defined, but which are on the Strategy Results page.

Predictors based on interaction history summaries are a predefined set of predictors that are based on interaction history summaries.

These predictors are enable by default.

Predictors can be one of two types: numeric or symbolic.

The system uses the property type as the default predictor type during the initial setup, but you can change the predictor type.

For example, when you know a numeric predictor has a small number of distinct values, change the predictor type from numeric to symbolic.

You can add a single or multiple candidate predictors.

Let’s add all the Financial services clickstream to the model.

That is pretty straightforward!

So to summarize, I should use a wide range of data sources, that include the customer profile and customer behavior, as input for my models.

Yes, that sums it up!

Thanks, Bella! This has been very helpful.

My pleasure.

Shadowing an Adaptive Model

When new potential predictors become available in Pega Customer Decision Hub™ but have not yet been approved by the governance team, you can shadow the model that drives a Prediction with a copy that has access to the new predictors. In shadow mode, the challenger model has no impact on the Business Outcomes but learns from the production data.

Transcript

Hi, I'm Hugo, a Data Scientist new to Pega. I'm working on a Customer Decision Hub project in financial services, responsible for monitoring the Adaptive Models that support next-best-action decisions.

Recently, the System Architect made a new set of potential predictors available in the Application.

They haven't been approved by the governance team yet, so I want to measure the predictive power of these new predictors without influencing the Business Outcomes.

I'm planning to ask Bella, the Lead System Architect on the project, how to set that up.

Hi Hugo, great to see you again!

Hi Bella! How can I safely evaluate newly available predictors of Adaptive Models?

To measure the predictive power of the potential predictors for Adaptive Models, you can shadow the model that drives the Prediction with a copy that has access to these new predictors.

Got it! Here's my Use Case: U+ Bank utilizes Customer Decision Hub to personalize the credit card offers on their website that customers see.

The Predict Web Propensity prediction calculates the propensity, or likelihood, that a customer will click on the banner displaying the credit card offer.

Customer Decision Hub uses these propensities to determine which offer to show.

Let’s see. The Adaptive Model Configuration that drives the Predict Web Propensity prediction is the Web Click Through Rate Customer configuration.

So first, you'll want to introduce a challenger model and use a copy of the active model configuration for the candidate models.

Next, add the new predictors that you want to monitor to the challenger model.

I'm looking to add the outcome of a churn model as a potential predictor to the model configuration.

The model has not been approved by the Governance team, but I’m curious to see its contribution to the predictive power of the adaptive models.

That is where the shadow mode comes in.

In shadow mode, the challenger model has no impact on the Business Outcomes but learns from the production data.

This allows you to monitor the predictive power of new predictors without violating company policies.

Perfect! So, I'll need a change request, correct? I work in a Business Operations Environment.

No. When you submit your changes for deployment, Customer Decision Hub automatically creates a change request in the current revision.

The Revision Manager ensures that all change requests are completed and then deploys the revision to all other environments, including production.

Great! The candidate models are new, so I'd assume they start learning from scratch.

Exactly, the challenger models begin learning when customers respond to the credit card offers after deployment of the Shadowing Pattern.

Let's say that, after a while, I notice that the new predictors significantly contribute to the predictive power of the models.

When the governance team approves the new predictors, how do I put them to good use?

When the challenger models outperform the active models, you can promote them.

A Champion Challenger Pattern lets you use the challenger model for a limited number of decisions.

Alternatively, you can replace the active model and make the new predictors available for all decisions.

Crystal clear! So, to summarize, to safely introduce new predictors in the Adaptive Models, I create a copy of the active model configuration and use it in a Shadowing pattern.

When the new predictors are approved, I can use the challenger models for a percentage of the decisions in a Champion Challenger Pattern, or completely replace the active model.

Yes, that sums it up.

Thanks for all the info, Bella!

Anytime, Hugo! Let's keep innovating!

Monitoring adaptive models

Description

It is a regular data scientist task to inspect the health of the out-of-the-box Pega Customer Decision Hub™ predictions and the adaptive models that drive them, and share the findings with the business team. The predictive performance and success rate of individual adaptive models provide information that can help business users and decisioning architects to refine business processes. Learn how to monitor the performance of predictions, adaptive models, and predictors.

Learning objectives

- Describe the lift metrics of a predictions
- Name the key metrics of adaptive models visualized in the bubble chart
- Inspect individual active and inactive predictors
- Explain how predictors with similar predictive performance are grouped
- Examine the propensity distribution and the trend for the whole model

Regular monitoring of adaptive models

Regular monitoring of adaptive models

Adaptive models will learn from all customer interactions, adjusting to changing behavior over time. To confirm the continuing accuracy of your adaptive models, perform the following tasks regularly:

- Check the performance and success rate of your models every two weeks.
- Inspect predictors every two or three months.

The purpose of regular inspection is to detect factors that negatively influence the performance of the adaptive models and the success rate of the actions.

Identifying technical problems

Look for adaptive models with a success rate of zero. This means that the actions for these models do not have any positive responses.

Identifying actions for which the model is not predictive

Look for adaptive models with low performance. Consider adding additional data as predictors.

Identifying actions that have a low number of responses

Look for adaptive models with a low number of responses. Discuss the eligibility criteria set in the Next-Best-Action Designer with the business. Changing the exclusion settings may increase the number of responses.

Identifying actions that are offered so often that they dominate other actions

Look for adaptive models with a high number of responses. A high number of responses might be fine from the business point of view. However, if necessary, prioritization can be adjusted in the Next-Best-Action Designer.

Identifying actions with a low success rate

Look for adaptive models with a low success rate. If the model performance is high, the relevance to the customers is high, but the action is unattractive and should be discussed with the business.

Inspecting an adaptive model

Inspect your model after introducing a new action, adding or removing a predictor, or changing prioritization. Take note of the active and inactive predictors.

Inspecting predictors

Check the details of a predictor with a low performance score. A possible cause can be too many missing values for the predictor. Look at the top predictors and in the bins that have a particularly high or low success rate.

Identifying predictors that are never used

Because unused predictors have only a minor effect on model performance, you do not need to remove them from an adaptive model configuration; however, you can conduct an occasional cleanup as part of your maintenance activities. An unused predictor might still become relevant for a future action.

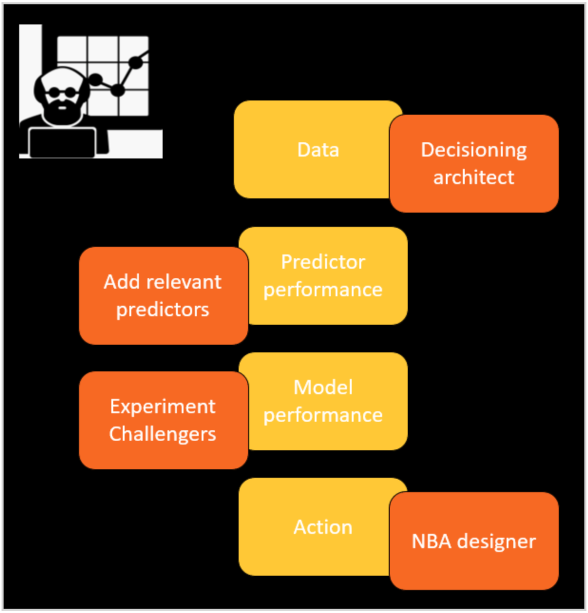
Monitoring predictions

In the production phase of a project, in response to Prediction Studio notifications or as a regular health check, a data scientist analyzes the predictions, models and predictors with the aim of providing insights to the governance team and to fix any issues with prediction performance.

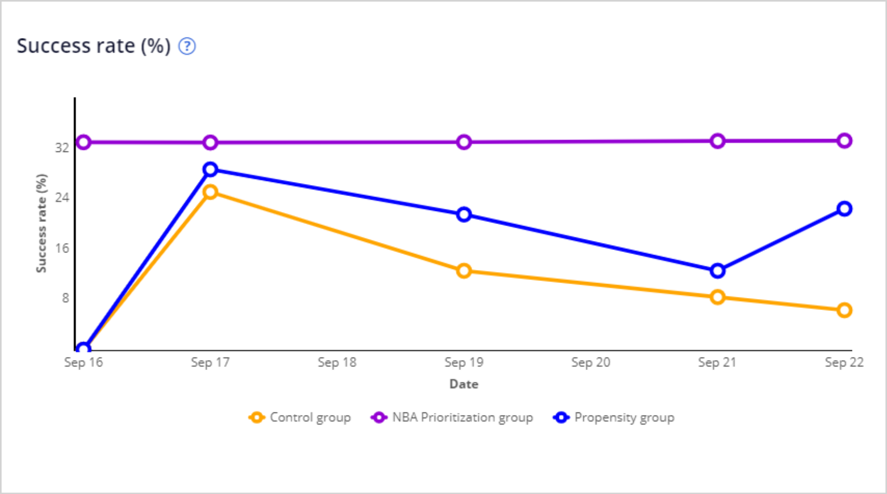
Transcript

This video shows you how to monitor predictions, models, and predictors in the production phase of a project.

In response to the notifications, a data scientist logs in to Prediction Studio in the production environment or in the business operations environment, examines the relevant charts, and checks the suggestions in the notifications. If it is a data quality issue, you refer the work to a decisioning architect. If it is a predictor performance issue, you add more relevant predictors. If it is a model performance issue, you run experiments and introduce challenger models through the MLOps process. If the action, not the model or the predictors, is the issue, you refer the work to the NBA designer team.

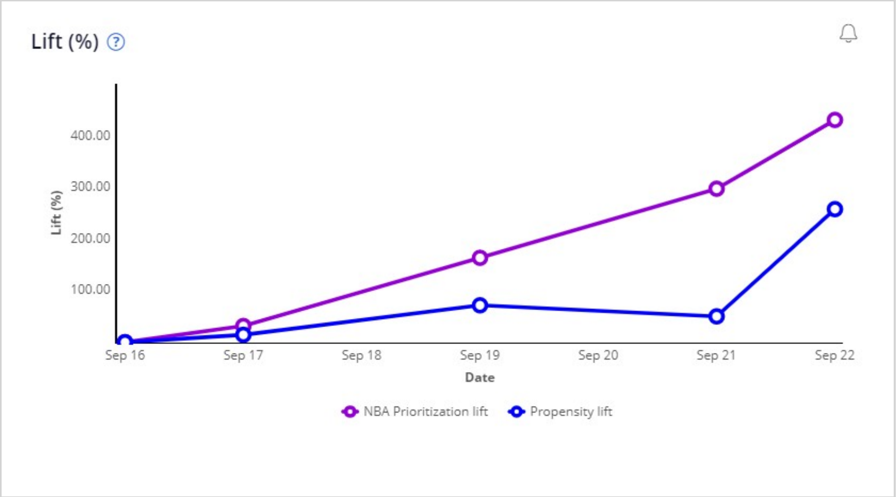


For the analysis of predictions, several charts are available. As an example, U+Bank uses the *Predict Web Propensity* prediction to calculate the likelihood that a customer will click on a web banner that offers one of the credit cards for which the customer is eligible. Success rate measures how successful the credit card offers are. The success rate is defined as the number of times the credit card offers are clicked, divided by the number of times they are offered.



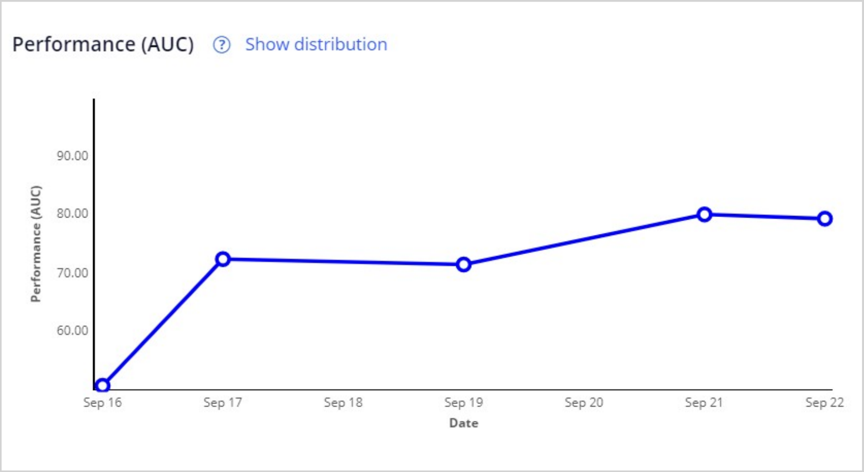
This chart shows the data for offers based on the performance of the Next Best Action strategy, with the majority of the customers in the *NBA Prioritization group*. A small percentage of the customers, who are the *Control group*, receive a random relevant offer. Another small percentage of the customers, who are the *Propensity group*, receive an offer based only on the model propensity with context weight, value and levers disabled. The *Control group* and the *Propensity group* are set to small percentages of the population in *Impact Analyzer*. A prediction needs attention when the success rate decreases over time, as this indicates a problem with the underlying models.

Lift is the impact on engagement that the AI generates. The lift metric can tell us how much better predictions made by a model are when compared to a series of random predictions for customers in the control group.



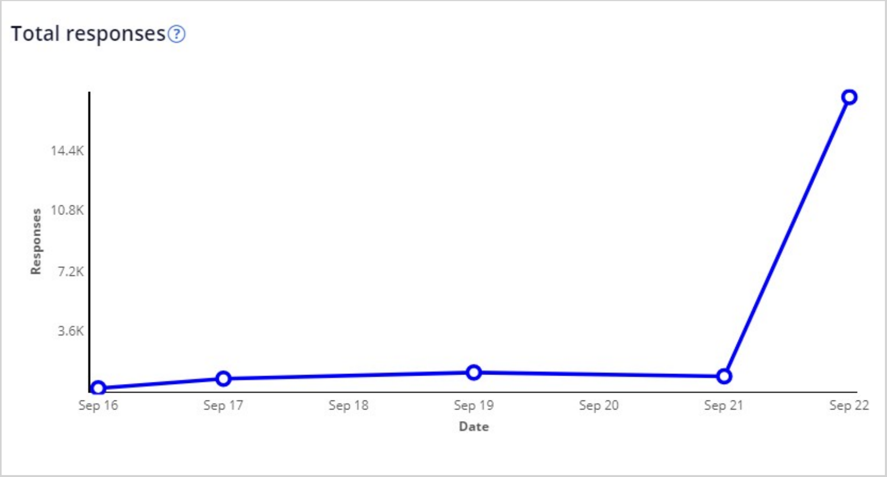
NBA Prioritization lift is the difference in the success rate of the *NBA Prioritization group* over the *Control group*. Similarly, *Propensity lift* compares the *Propensity group* to the *Control group*. The prediction needs attention when the lift decreases over time. The lift may be low if the model is still immature, or if the control group is not representative enough of the actual population.

Performance measures the accuracy of a prediction in predicting an outcome, and ranges from 50 to 100.



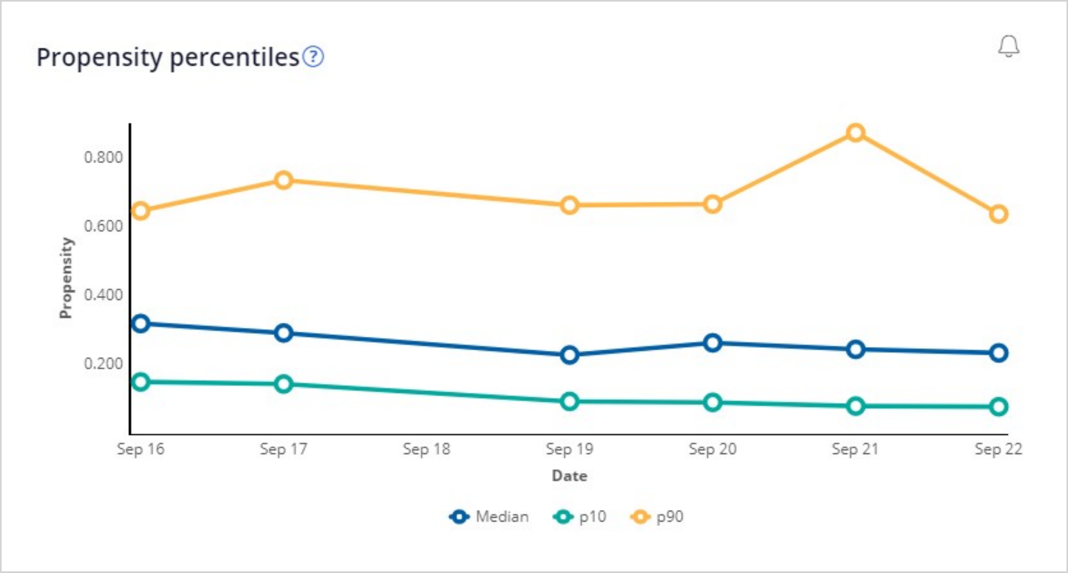
The prediction needs attention when the performance decreases over time. This may indicate issues with the predictor set of the models.

Total responses measures the number of responses the models receive to base their output on.



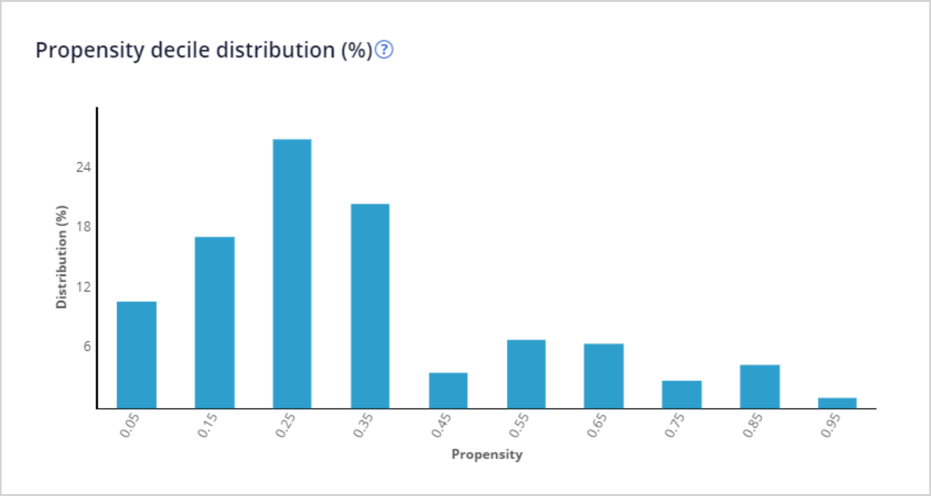
The prediction needs attention when the number of responses received is zero for a week or more. This may have multiple causes and needs to be looked at by the Decisioning Architect.

The Propensity percentiles chart shows the propensity values for the lowest 10%, for the highest 90% and for the median.



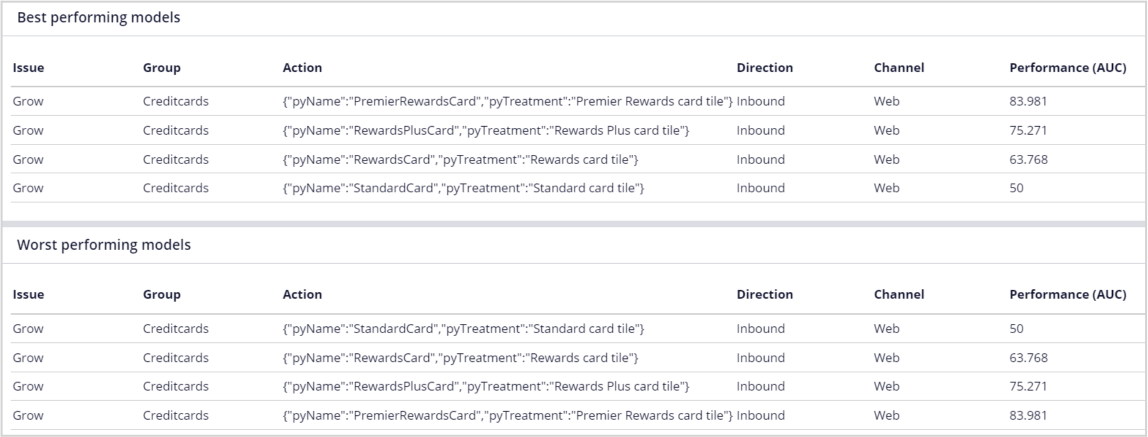
This metric needs attention when there is a drastic change in percentile values, especially the median, compared to the previous week. Percentile values can drastically change if the models are experiencing data drift, or when the models receive too many missing values, resulting in percentile drift.

Propensity decile distribution shows the frequency of a certain propensity in a range from 0 (no likelihood), to 1 (certainty).



For inbound use cases, the propensities peak at higher values than outbound use cases and continue reducing over the higher value deciles, indicating that models are more confident in making inbound predictions than outbound predictions. The prediction needs attention when the propensity is very similar across the deciles or moving up towards the higher deciles. This may indicate overfitting of the model towards the data.

Model analysis offers listings of the best and worst performing adaptive models.



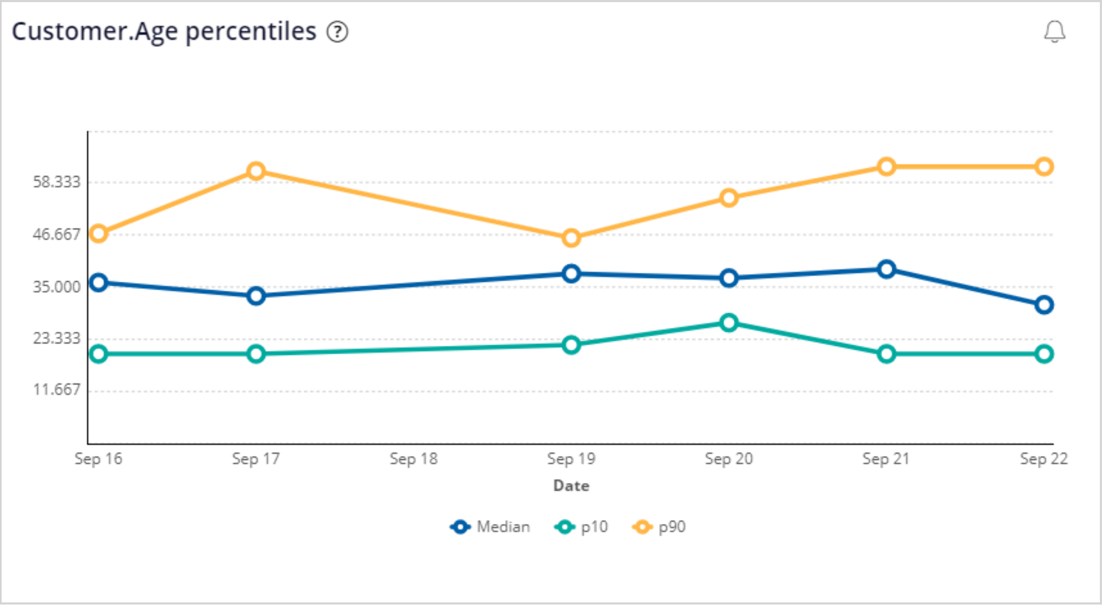
Best performing models

Issue	Group	Action	Direction	Channel	Performance (AUC)
Grow	Creditcards	{"pyName":"PremierRewardsCard","pyTreatment":"Premier Rewards card tile"}	Inbound	Web	83.981
Grow	Creditcards	{"pyName":"RewardsPlusCard","pyTreatment":"Rewards Plus card tile"}	Inbound	Web	75.271
Grow	Creditcards	{"pyName":"RewardsCard","pyTreatment":"Rewards card tile"}	Inbound	Web	63.768
Grow	Creditcards	{"pyName":"StandardCard","pyTreatment":"Standard card tile"}	Inbound	Web	50

Worst performing models

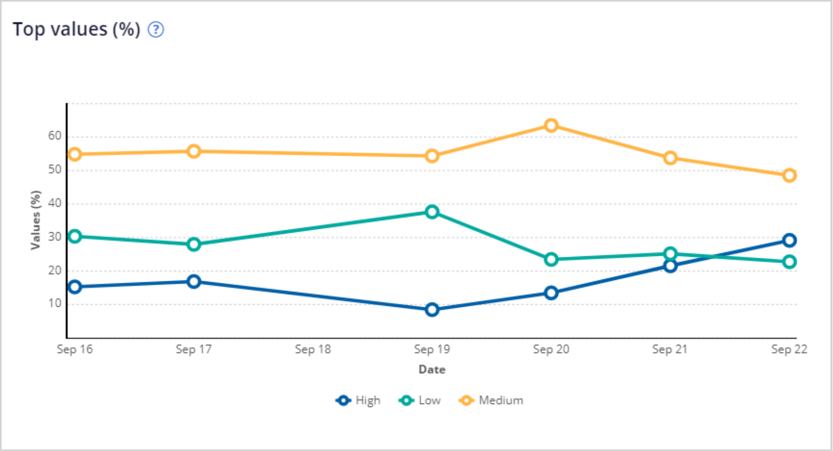
Issue	Group	Action	Direction	Channel	Performance (AUC)
Grow	Creditcards	{"pyName":"StandardCard","pyTreatment":"Standard card tile"}	Inbound	Web	50
Grow	Creditcards	{"pyName":"RewardsCard","pyTreatment":"Rewards card tile"}	Inbound	Web	63.768
Grow	Creditcards	{"pyName":"RewardsPlusCard","pyTreatment":"Rewards Plus card tile"}	Inbound	Web	75.271
Grow	Creditcards	{"pyName":"PremierRewardsCard","pyTreatment":"Premier Rewards card tile"}	Inbound	Web	83.981

The percentiles for the values of the Age predictor give an idea of the range of values for this numerical predictor.



The predictor needs attention when there is a drastic change in percentile values, especially the median, compared to the previous week, as the models experience data drift, or receive too many missing values.

This chart shows the trend of the most frequent values of the symbolic predictor CLV:



The predictor needs attention when a trend line drastically changes, which indicates an underlying change in the data that may cause data drift or concept drift.

This demo has concluded. What did it show you?

- How to monitor Customer Decision Hub predictions.

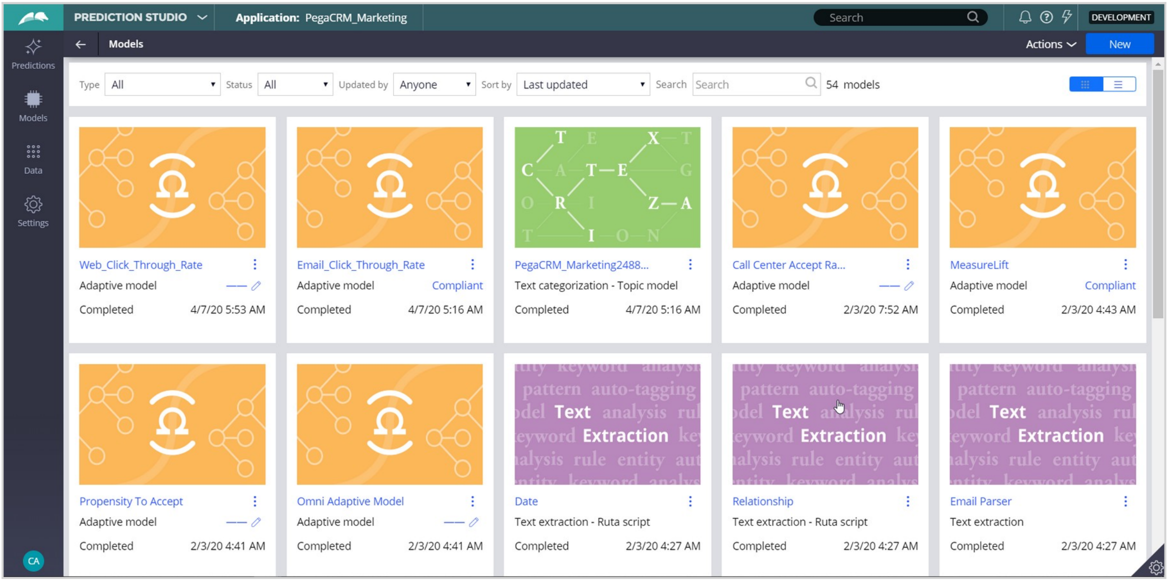
Monitoring adaptive models

Introduction

The predictive performance and success rate of individual adaptive models provide information that can help business users and decisioning consultants to refine the Next-Best-Actions of the company. Monitoring the health of adaptive models and their predictors is a regular data scientist task that can be performed in Prediction Studio in the production environment or the business operations environment.

Transcript

This demo will show how to inspect the health of your adaptive models and their predictors. This is a regular data scientist task. The predictive performance and success rate of individual adaptive models provide information that can help business users and decisioning consultants to refine the Next-Best-Actions of the company.



Type: All · Status: All · Updated by: Anyone · Sort by: Last updated · Search: Search · 54 models

Web_Click_Through_Rate
Adaptive model
Completed 4/7/20 5:53 AM

Email_Click_Through_Rate
Adaptive model · Compliant
Completed 4/7/20 5:16 AM

PegaCRM_Marketing2488...
Text categorization - Topic model
Completed 4/7/20 5:16 AM

Call Center Accept Ra...
Adaptive model
Completed 2/3/20 7:52 AM

MeasureLift
Adaptive model · Compliant
Completed 2/3/20 4:43 AM

Propensity To Accept
Adaptive model
Completed 2/3/20 4:41 AM

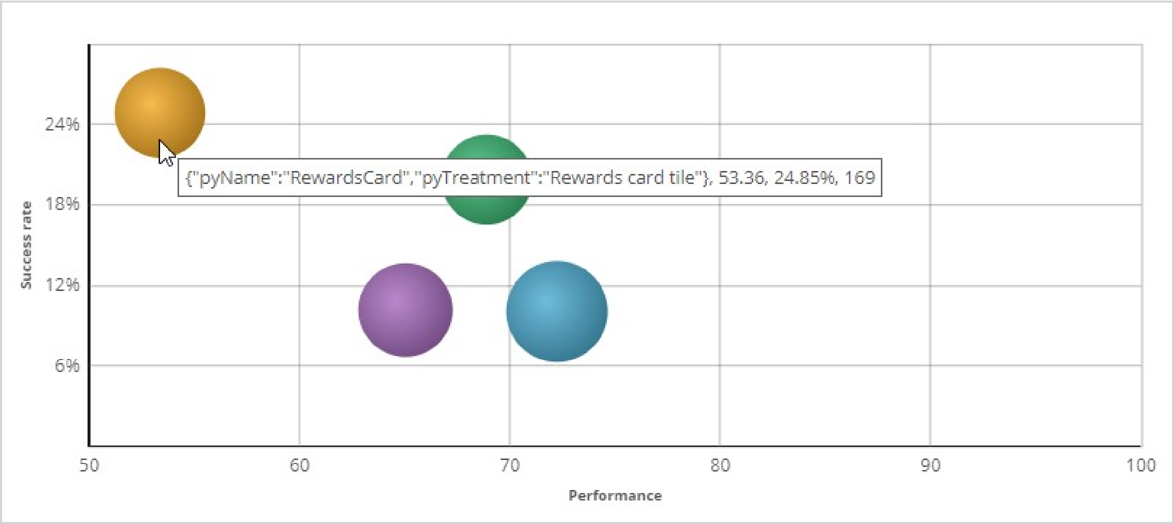
Omni Adaptive Model
Adaptive model
Completed 2/3/20 4:41 AM

Date
Text extraction - Ruta script
Completed 2/3/20 4:27 AM

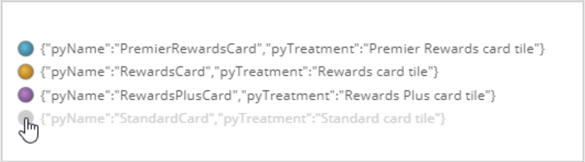
Relationship
Text extraction - Ruta script
Completed 2/3/20 4:27 AM

Email Parser
Text extraction
Completed 2/3/20 4:27 AM

We will inspect the Web_Click_Through_Rate model, that calculates the propensity that a customer will respond positively to an offer made on the web channel. The Monitor tab of an adaptive model configuration shows a bubble chart that visualizes the key metrics of all models generated.



Each bubble represents the model for a specific action. The size of a bubble indicates the number of responses (positive and negative) to that action that have been used in the adaptive learning process. In this example, there is a model for every action belonging to the Credit Card group. When you hover the cursor over a bubble, you can view the name of the action, the performance, the success rate, and the number of responses. In the legend, the display of models can be toggled on and off.

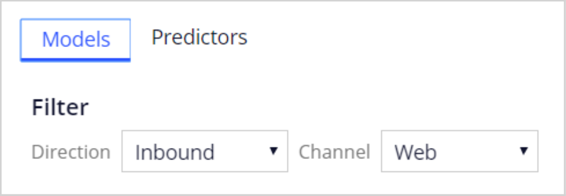

{"pyName":"PremierRewardsCard","pyTreatment":"Premier Rewards card tile"}
{"pyName":"RewardsCard","pyTreatment":"Rewards card tile"}
{"pyName":"RewardsPlusCard","pyTreatment":"Rewards Plus card tile"}
{"pyName":"StandardCard","pyTreatment":"Standard card tile"}

The Performance axis indicates the accuracy of the outcome prediction. The model performance is expressed in the Area Under the Curve (AUC) unit of measurement, which has a range between 50 and 100. The higher the AUC, the better a model is at predicting the outcome.

The Success rate axis indicates the success rate expressed in percentages. In this example, the success rate represents how often a web banner is clicked. The system calculates this rate by dividing the number of times a banner is clicked by the total number of times the banner was shown on the website.

The information you see here is extracted from the Adaptive Data Mart, which is a reporting view of the Adaptive Decision Manager (ADM) server. The Adaptive Data Mart is built automatically by a process running in the background. This process creates snapshots at regular time intervals. This means that the data as reported may not be the same as the data you see when you view it in real-time. You can refresh the view to synchronize the data.

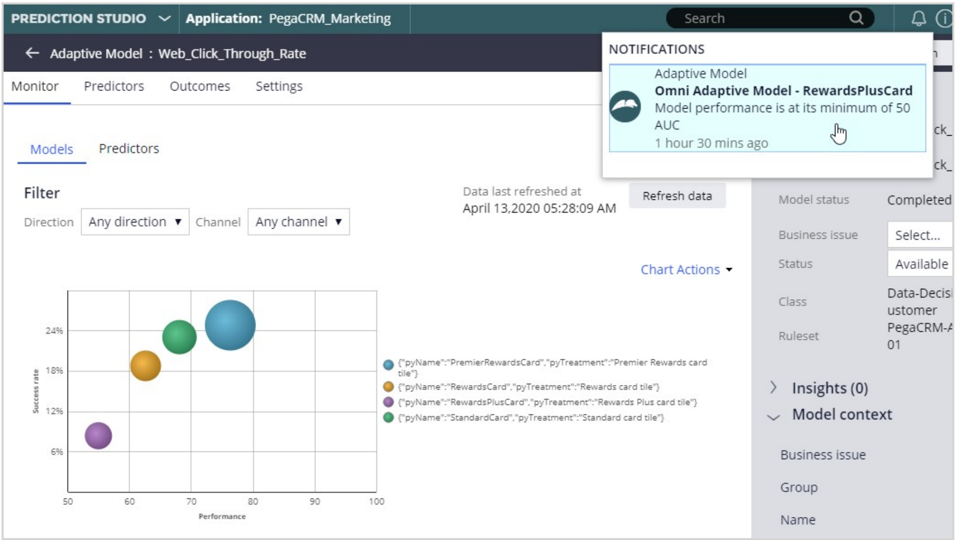
The model context includes the channel and direction, so you have different models for the Call Center, Email, and Web channels, as well as for the inbound and outbound directions. You can apply filtering to focus on models for a particular direction or channel, or a combination of the two.



Filter

Direction: Inbound ▾ Channel: Web ▾

Actionable insights are generated for individual models when the number of responses, model performance, or success rate significantly changes over time.



NOTIFICATIONS
Adaptive Model
Omni Adaptive Model - RewardsPlusCard
Model performance is at its minimum of 50 AUC
1 hour 30 mins ago

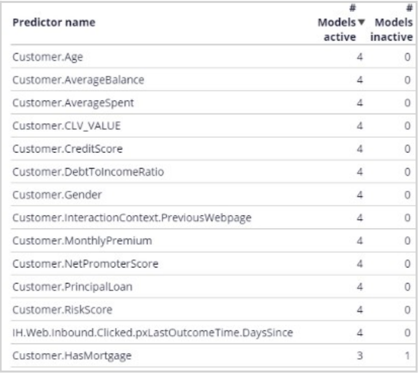
Direction: Any direction ▾ Channel: Any channel ▾

Data last refreshed at April 13,2020 05:28:09 AM

● {"pyName":"PremierRewardsCard","pyTreatment":"Premier Rewards card tile"}
● {"pyName":"RewardsCard","pyTreatment":"Rewards card tile"}
● {"pyName":"RewardsPlusCard","pyTreatment":"Rewards Plus card tile"}
● {"pyName":"StandardCard","pyTreatment":"Standard card tile"}

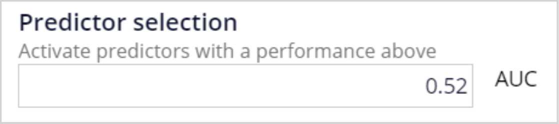
Model status: Completed
Business issue: Select...
Status: Available
Class: Data-Decisi ustomer
Ruleset: PegaCRM-A 01

On the Predictors tab, the number of models in which a predictor is active, and the performance of the predictor is displayed.



Predictor name	# Models ▾ active	# Models inactive
Customer.Age	4	0
Customer.AverageBalance	4	0
Customer.AverageSpent	4	0
Customer.CLV_VALUE	4	0
Customer.CreditScore	4	0
Customer.DebtToIncomeRatio	4	0
Customer.Gender	4	0
Customer.InteractionContext.PreviousWebpage	4	0
Customer.MonthlyPremium	4	0
Customer.NetPromoterScore	4	0
Customer.PrincipalLoan	4	0
Customer.RiskScore	4	0
IH.Web.Inbound.Clicked.pxLastOutcomeTime.DaysSince	4	0
Customer.HasMortgage	3	1

In this case, the Age predictor is used in all four models. The HasMortgage predictor is active in three models and inactive in one model, where its predictive power is below the threshold. The default value for this threshold is 52 percent.



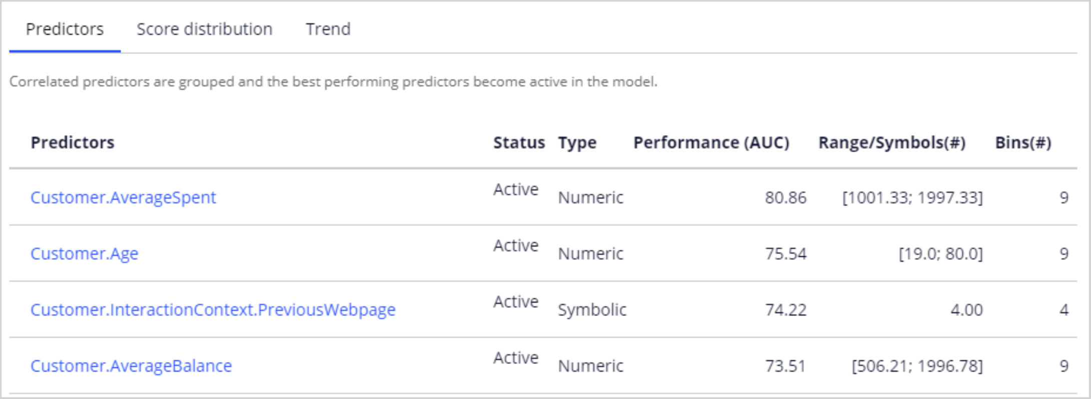
Predictor selection

Activate predictors with a performance above

0.52 AUC

The system continuously monitors the predicting power of every predictor. If the predicting power of a predictor drops below the threshold value that predictor is deactivated. The data that is used to visualize the models in the bubble chart is displayed in a table below the chart. For each model number of responses, success rate, and performance are shown.

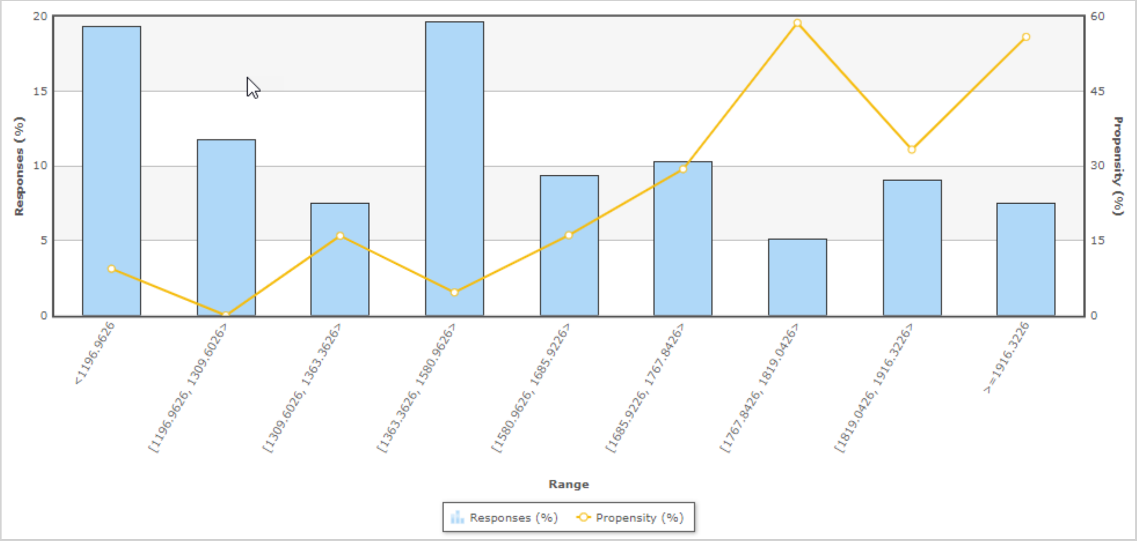
From the adaptive model table, you can drill down into a model report for a specific adaptive model.



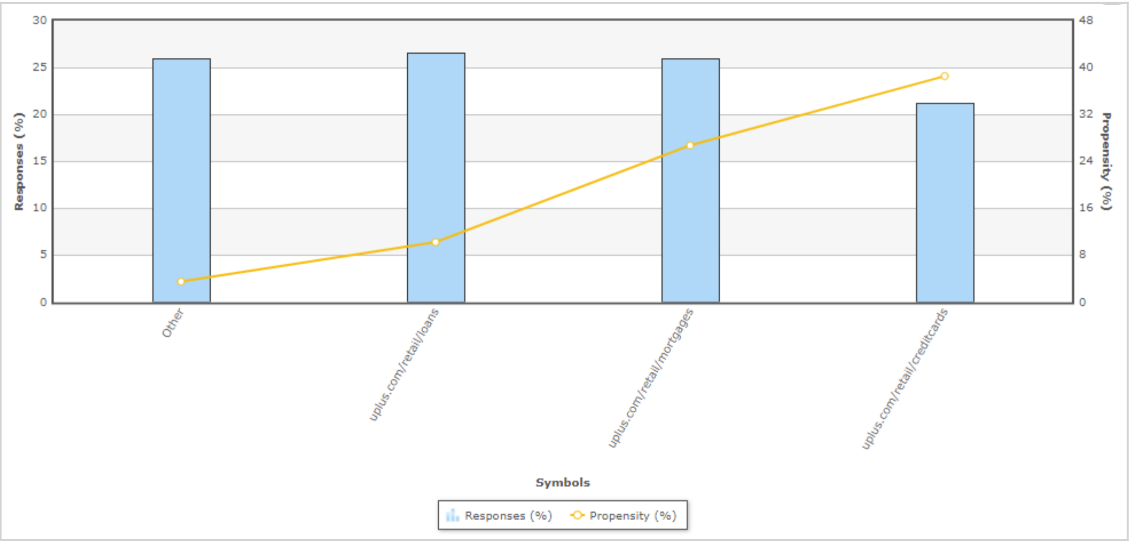
Correlated predictors are grouped and the best performing predictors become active in the model.

Predictors	Status	Type	Performance (AUC)	Range/Symbols(#)	Bins(#)
Customer.AverageSpent	Active	Numeric	80.86	[1001.33; 1997.33]	9
Customer.Age	Active	Numeric	75.54	[19.0; 80.0]	9
Customer.InteractionContext.PreviousWebpage	Active	Symbolic	74.22	4.00	4
Customer.AverageBalance	Active	Numeric	73.51	[506.21; 1996.78]	9

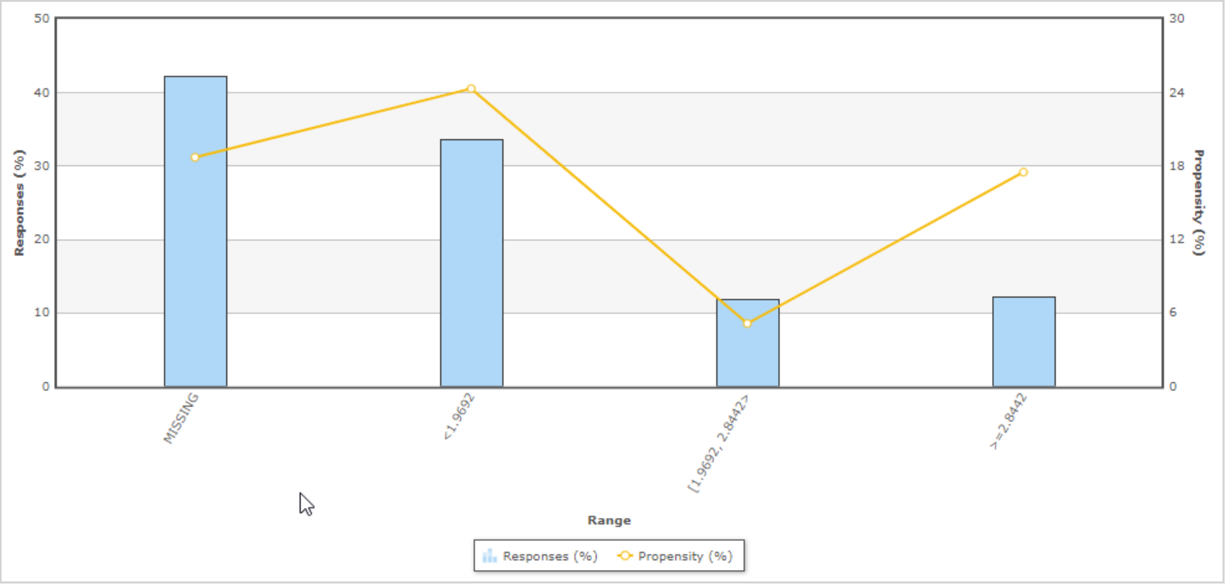
There are three tabs, reporting on predictors, the model score distribution, and the trend. In the predictors report, you can examine the performance of individual predictors. Let’s examine a couple of them. In this case, the best-performing predictor is AverageSpent. This is a predictor of type numeric. The system split the AverageSpent predictor into 9 bins. Each bin has its own offer propensity.



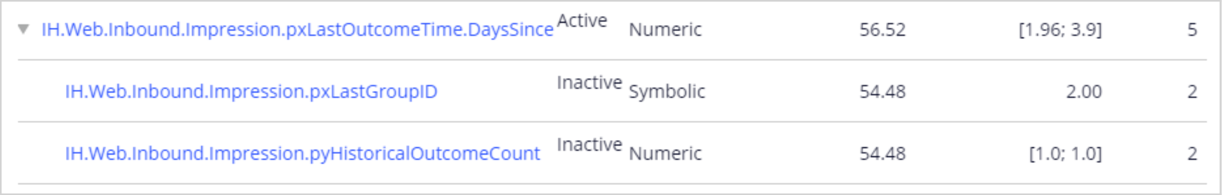
Propensity is the likelihood of positive customer behavior, which in this example is clicking on a web banner. Now, let's examine the symbolic PreviousWebpage predictor. The system split this predictor into 4 bins. The context of an interaction, in this case, the previous web page visited by the customer, can be highly predictive.



To further improve the predictive power of the models the system uses Interaction History summaries. In this example, the adaptive system established that the number of days since the offer was accepted is a well-performing predictor.

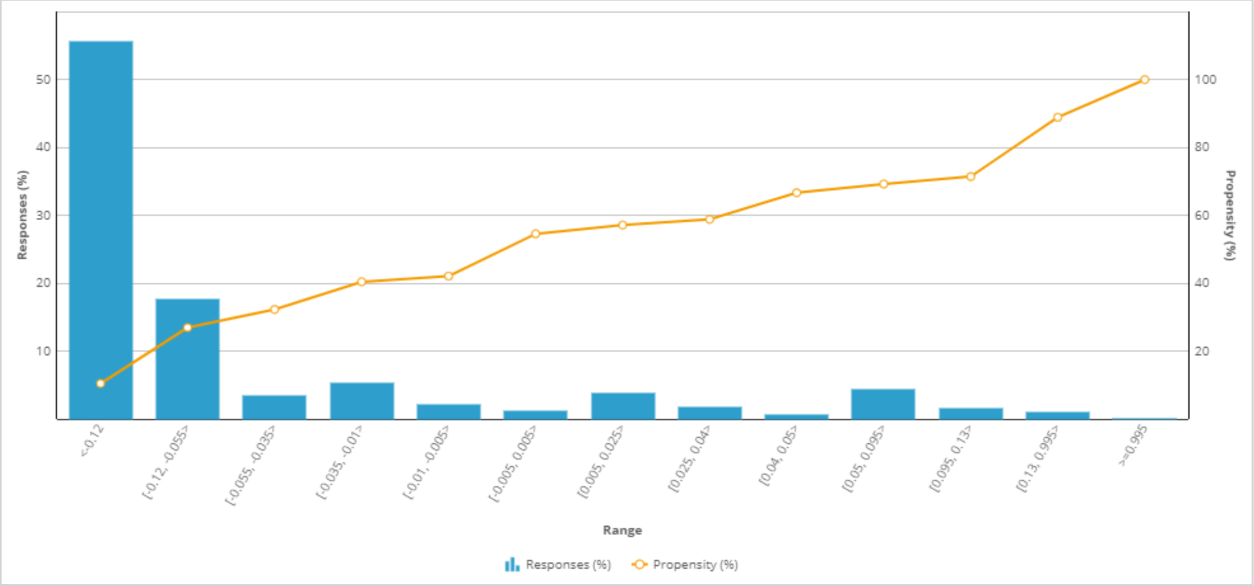


The system grouped three predictors that are correlated. It then marked two of them as inactive. Inactive predictors are not used in the propensity calculation.

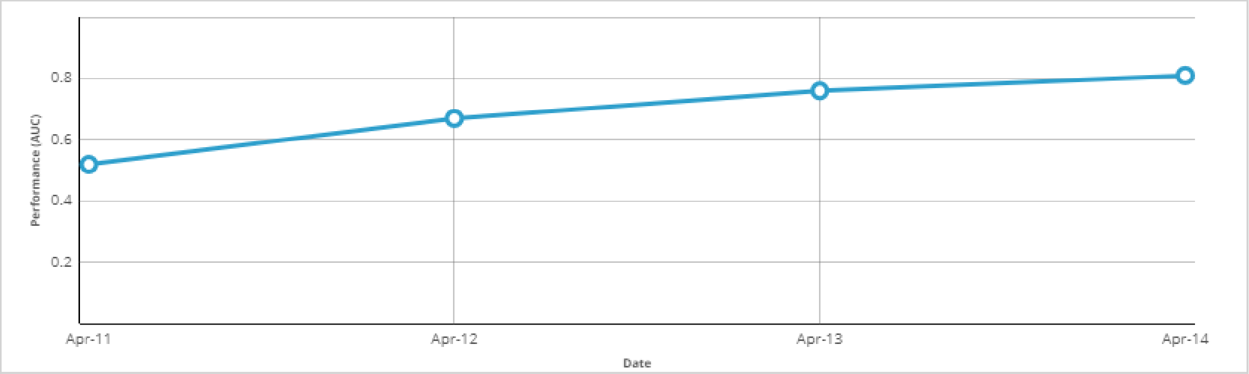


▼ IH.Web.Inbound.Impression.pxLastOutcomeTime.DaysSince	Active	Numeric	56.52	[1.96; 3.9]	5
IH.Web.Inbound.Impression.pxLastGroupID	Inactive	Symbolic	54.48	2.00	2
IH.Web.Inbound.Impression.pyHistoricalOutcomeCount	Inactive	Numeric	54.48	[1.0; 1.0]	2

The Score distribution report enables you to examine the propensity distribution for the whole model.



And in the trend report, you can see the performance of the model over time.



This demo has concluded. What did it show you?

- How the key metrics of adaptive models are visualized in a bubble chart.
- How you can customize the bubble chart by filtering.
- How to inspect active and inactive predictors.
- How to inspect individual predictors.

- How predictors with similar predictive performance are grouped.
- How to examine the propensity distribution for the whole model.
- How to examine the trend for the whole model.

Exporting adaptive model data

Description

The reporting datamart of Pega Adaptive Decision Manager (ADM) is an open data model. As a result, data scientists that work on Pega Customer Decision Hub™ projects can export adaptive model data, predictor binning data, and historical data (predictor and outcome values) for further analysis. Learn how to configure data flows to export the model data of interest.

You can build meaningful plots and more with the open-source GitHub repository Pega Data Scientist Tools.

Learning objectives

- Export the historical data from a Customer Decision Hub system.
- Configure data flows to export selected adaptive model data and predictor data.
- Visualize adaptive model data and predictor data with Pega Data Scientist Tools.

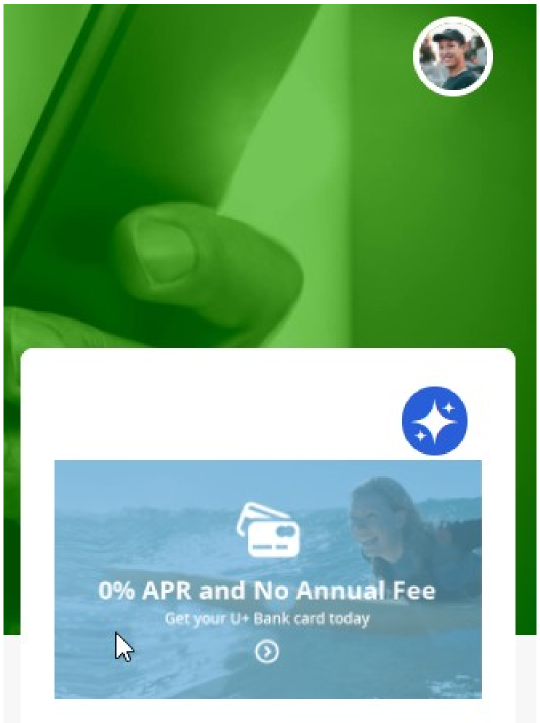
Exporting historical data

Learn how to extract historical data (predictors and outcomes) from adaptive models in your application to perform offline analysis or use the data to build models using the machine learning service of your choice.

Transcript

This demo shows you how to export the customer interaction data that is used by adaptive models to make predictions, including all predictor data and associated outcomes, for offline analysis.

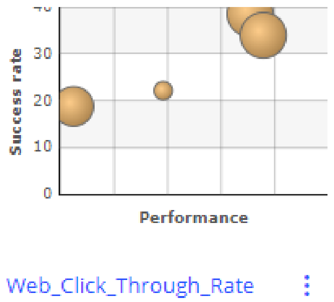
U+ Bank has implemented Pega Customer Decision Hub™ to show a personalized banner on their website that advertises credit card offers.



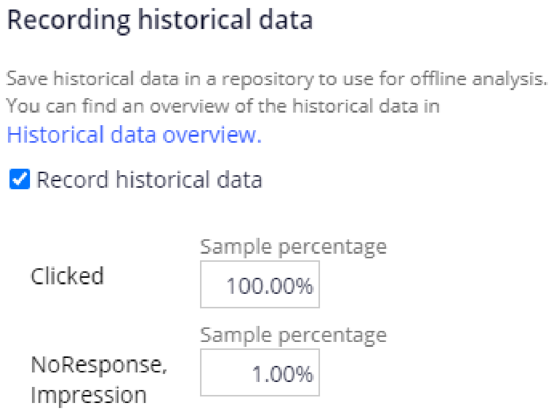
When a customer is eligible for multiple credit cards, adaptive models decide which card to show. When the customer ignores the banner, the adaptive model that drives the decision regards this as negative behavior. When a customer clicks on the banner, the model regards this as a positive behavior.

As a data scientist, you may want to inspect the raw predictor data used by an adaptive model and the customer interaction outcome to validate data assumptions and check for concept drift. You can also use the data to build various predictive models externally.

All models are managed in Prediction Studio. The adaptive model that drives the decision over which banner to display is the **Web Click Through Rate** model.



To extract the data, you enable the recording of historical data for a selected adaptive model. A web banner typically has a low click-through rate and a significantly lower number of positive responses than negative responses. In such cases, you can sample all positive outcomes and just one percent of the negative outcomes to limit the storage space needed.



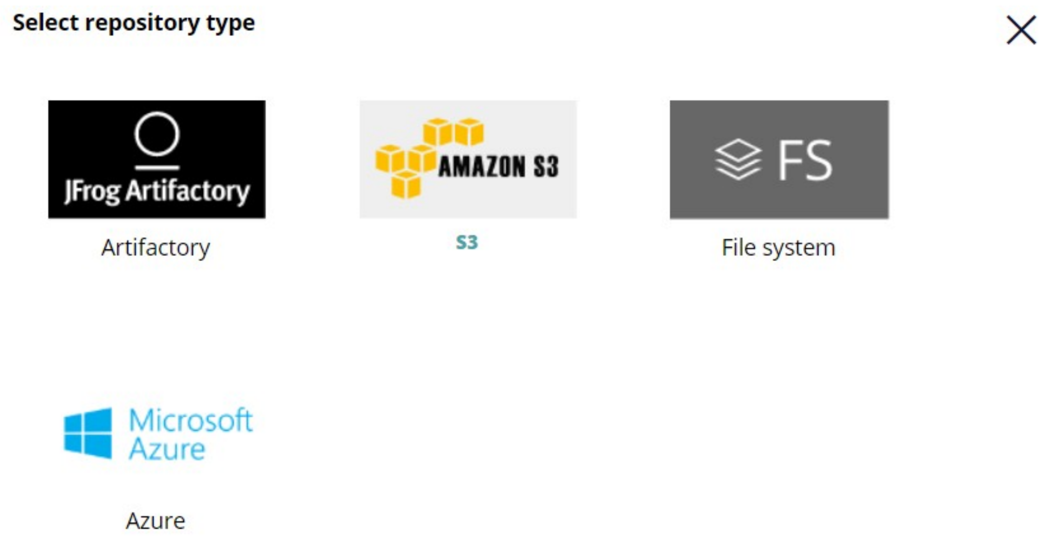
Recording historical data

Save historical data in a repository to use for offline analysis.
You can find an overview of the historical data in
Historical data overview.

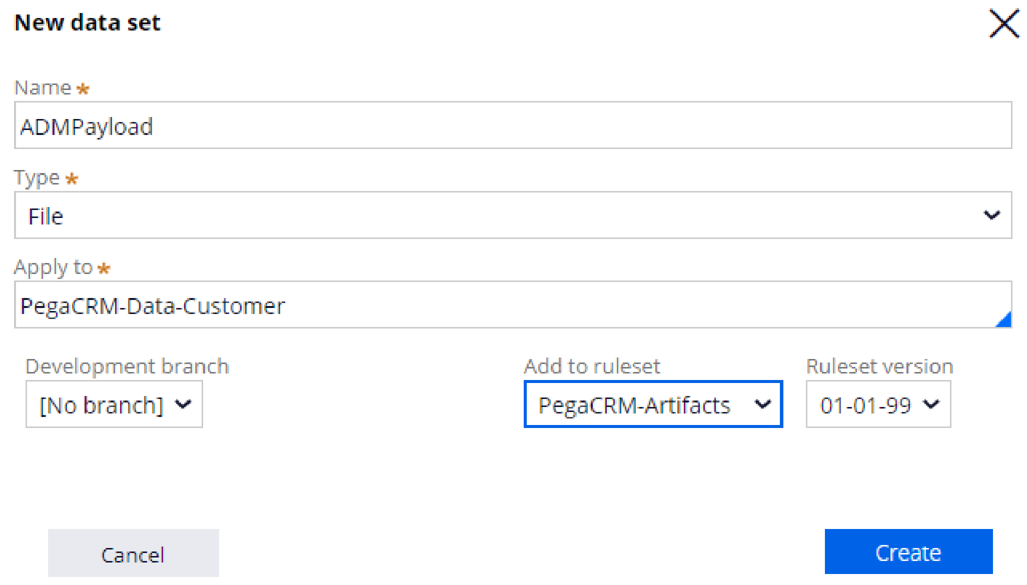
☑ Record historical data

Clicked	Sample percentage 100.00%
NoResponse, Impression	Sample percentage 1.00%

The sample percentages determine the likelihood that a customer response is recorded. The system stores the predictor data and outcome as a JSON file in a repository of your choice. By default, the data is stored for 30 days in the **defaultstore** repository. However, this repository points to a temporary directory, and a system architect should switch to a resilient repository to avoid data loss. Supported repository types include Microsoft Azure and Amazon S3.



For this demo, we use the default store repository and create a data set to export the data.



New data set

Name * ADMPayload

Type * File

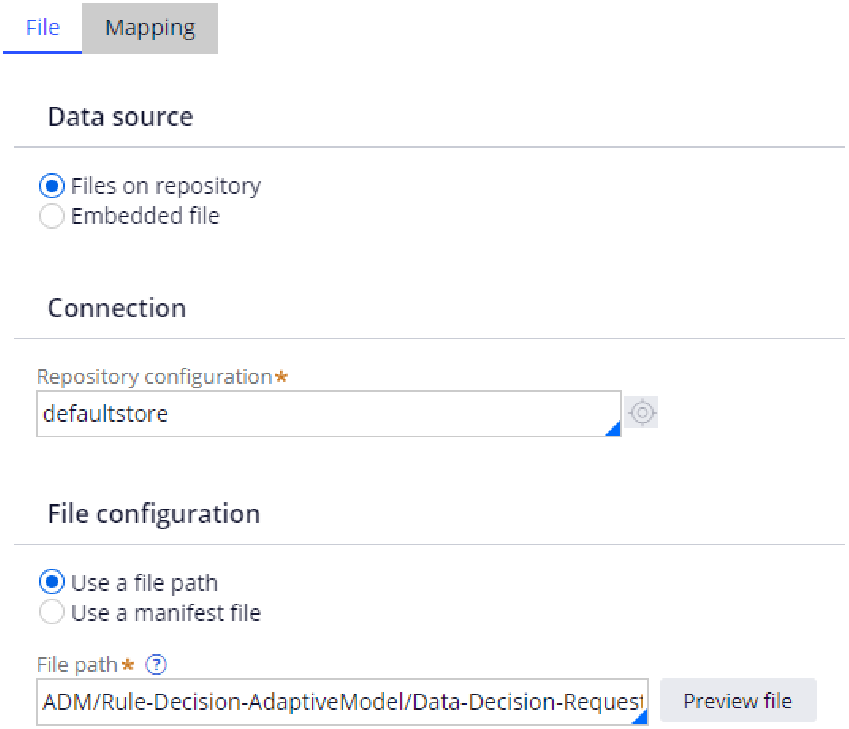
Apply to * PegaCRM-Data-Customer

Development branch [No branch]

Add to ruleset PegaCRM-Artifacts

Ruleset version 01-01-99

The data set is mapped to the file that contains the recorded historical data.



Data source

● Files on repository
○ Embedded file

Connection

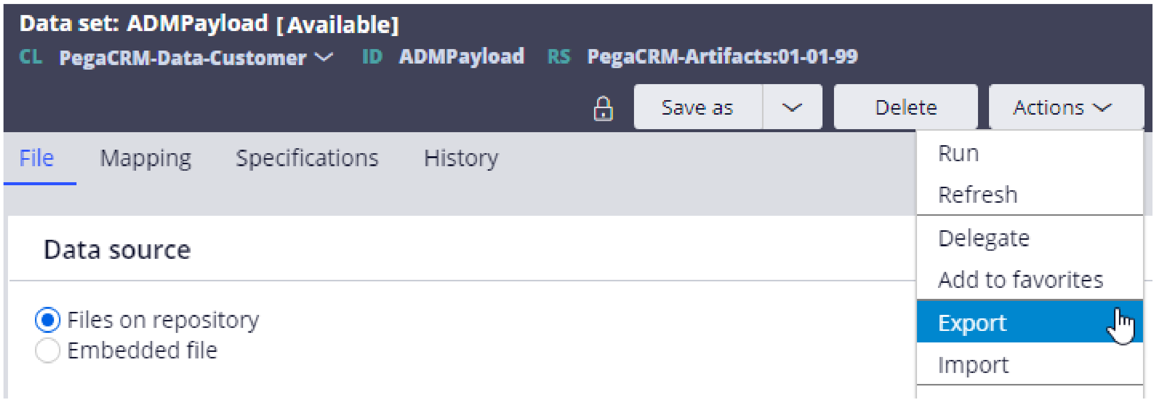
Repository configuration*: defaultstore

File configuration

● Use a file path
○ Use a manifest file

File path*: ADM/Rule-Decision-AdaptiveModel/Data-Decision-Request

With these settings in place, the input data used for the prediction and associated outcome are stored in the configured data set when customers see an offer and click on an offer. The system architect can download the data set in DEV Studio.



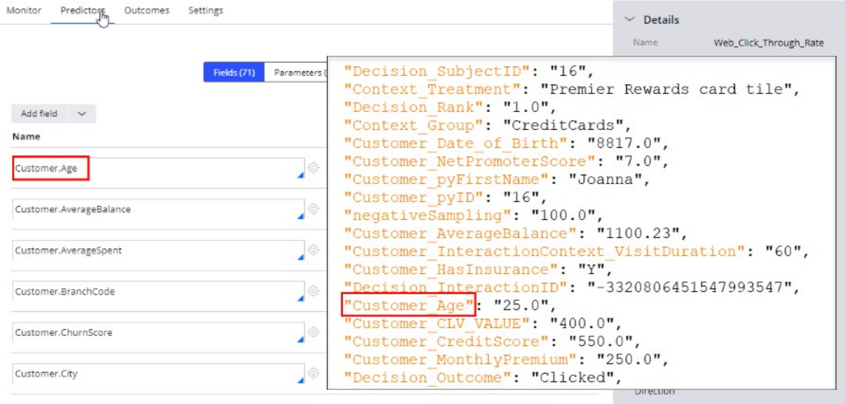
Data set: ADMPayload [Available]

CL PegaCRM-Data-Customer ID ADMPayload RS PegaCRM-Artifacts:01-01-99

Data source

● Files on repository
○ Embedded file

Every record contains the predictor values used for the prediction, as well as the context and the decision properties, including the outcome of the interaction. All property names are automatically converted to comply with the JSON format.



```
"Decision_SubjectID": "16",
"Context_Treatment": "Premier Rewards card tile",
"Decision_Rank": "1.0",
"Context_Group": "CreditCards",
"Customer_Date_of_Birth": "8817.0",
"Customer_NetPromoterScore": "7.0",
"Customer_pyFirstName": "Joanna",
"Customer_pyID": "16",
"negativeSampling": "100.0",
"Customer_AverageBalance": "1100.23",
"Customer_InteractionContext_VisitDuration": "60",
"Customer_HasInsurance": "Y",
"Decision_InteractionID": "-3320806451547993547",
"Customer_Age": "25.0",
"Customer_CLV_VALUE": "400.0",
"Customer_CreditScore": "550.0",
"Customer_MonthlyPremium": "250.0",
"Decision_Outcome": "Clicked",
```

To use the JSON file for further analysis, import the file into a third-party analytics tool. Keep in mind that when many customers visit the website, the file size becomes very large in a short time. To limit the storage space needed, you can lower the sample percentages.

You have reached the end of this demo. What did it show you?

- How to export the raw data that is used by adaptive models.
- What data is captured during a customer interaction.

Exporting selected model data for external analysis

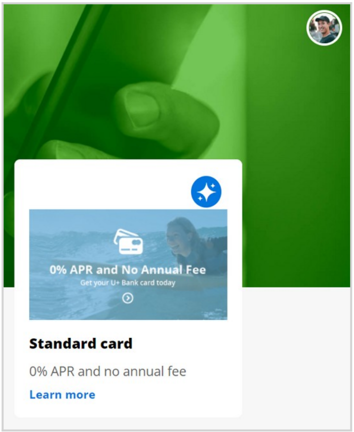
Data scientists can export snapshots of the adaptive models that drive Pega Customer Decision Hub™ predictions from the Adaptive Decision Manager (ADM) data mart for further analysis in their favorite analytical tools.

To limit the scope of the export to the data of interest, learn how to customize the export to populate data sets that contain only data that are relevant to you.

Transcript

This video shows you how to customize the export of adaptive model data for offline analysis in analytical tools such as Python and R. The publicly available GitHub repository Pega Data Scientist Tools helps to build meaningful plots and more with the exported data.

U+ Bank uses Customer Decision Hub to determine which credit card offer to show on their website when a customer logs in. For each offer, an adaptive model determines the likelihood that the customer will click on the web banner.

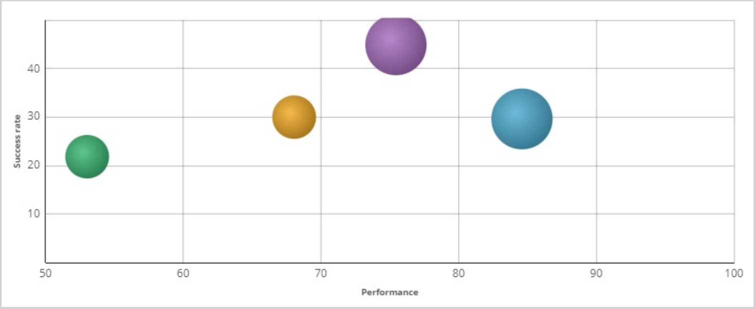


Standard card

0% APR and no annual fee

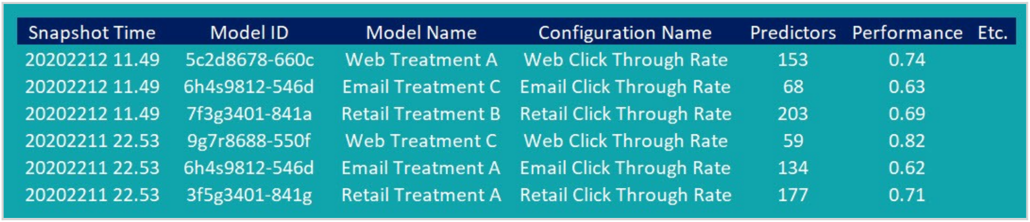
Learn more

In Prediction Studio, Data Scientists continuously monitor the state of their predictions and the adaptive models that drive them.



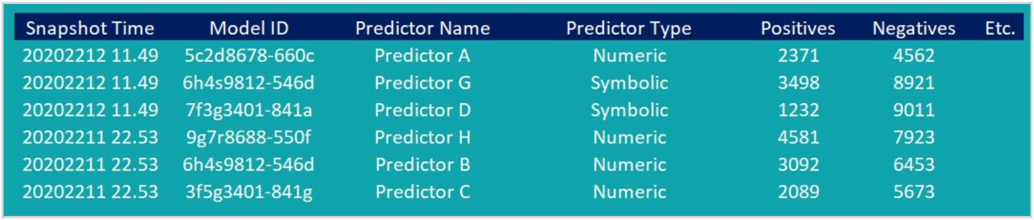
For offline analysis of the adaptive model data, they can export the data from the Adaptive Decision Manager data mart. Two database tables contain the required monitoring information, and these tables populate two data sets.

The *ADM snapshot* data set contains snapshots that include the model ID, the model name, the configuration name, and model attributes such as the number of predictors, the model performance, and many others.



Snapshot Time	Model ID	Model Name	Configuration Name	Predictors	Performance	Etc.
20202212 11.49	5c2d8678-660c	Web Treatment A	Web Click Through Rate	153	0.74	
20202212 11.49	6h4s9812-546d	Email Treatment C	Email Click Through Rate	68	0.63	
20202212 11.49	7f3g3401-841a	Retail Treatment B	Retail Click Through Rate	203	0.69	
20202211 22.53	9g7r8688-550f	Web Treatment C	Web Click Through Rate	59	0.82	
20202211 22.53	6h4s9812-546d	Email Treatment A	Email Click Through Rate	134	0.62	
20202211 22.53	3f5g3401-841g	Retail Treatment A	Retail Click Through Rate	177	0.71	

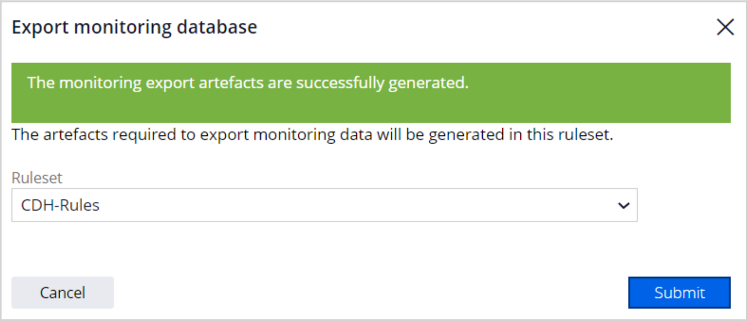
The *ADM predictor* data set contains snapshots of the binning of individual predictors. The data sets have the Model ID key in common.



Snapshot Time	Model ID	Predictor Name	Predictor Type	Positives	Negatives	Etc.
20202212 11.49	5c2d8678-660c	Predictor A	Numeric	2371	4562	
20202212 11.49	6h4s9812-546d	Predictor G	Symbolic	3498	8921	
20202212 11.49	7f3g3401-841a	Predictor D	Symbolic	1232	9011	
20202211 22.53	9g7r8688-550f	Predictor H	Numeric	4581	7923	
20202211 22.53	6h4s9812-546d	Predictor B	Numeric	3092	6453	
20202211 22.53	3f5g3401-841g	Predictor C	Numeric	2089	5673	

Both tables can grow very large, but you typically need only the data for a selection of the models. For example, you may only be interested in the models for the application you are working on, or in just a particular channel. This demo shows you how to customize the export of the two data sets to your repository.

In the implementation phase of the project, you generate the artifacts required for the export of adaptive and predictive model snapshots, including the required data flows in Prediction Studio.



Export monitoring database

The monitoring export artefacts are successfully generated.

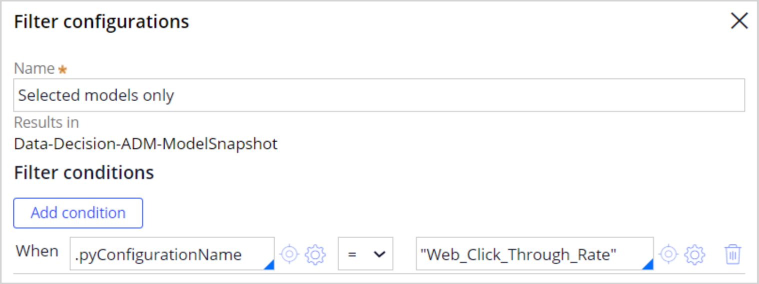
The artefacts required to export monitoring data will be generated in this ruleset.

Ruleset: CDH-Rules

Data flows are scalable and resilient data pipelines that you can use to ingest, process, and move data to one or more destinations.

To limit the size of the exports, you may want to select only the data that you are interested in before exporting. For example, to select the data of models based on the *Web Click Through Rate* model configuration, adjust the two relevant data flows.

The *ADM snapshot data export* data flow exports the model snapshots to the repository. To configure the data flow to only export the relevant snapshots, add a filter component that only passes on snapshots on the condition that the model configuration name equals *Web Click Through Rate*.



Filter configurations

Name*

Selected models only

Results in

Data-Decision-ADM-ModelSnapshot

Filter conditions

Add condition

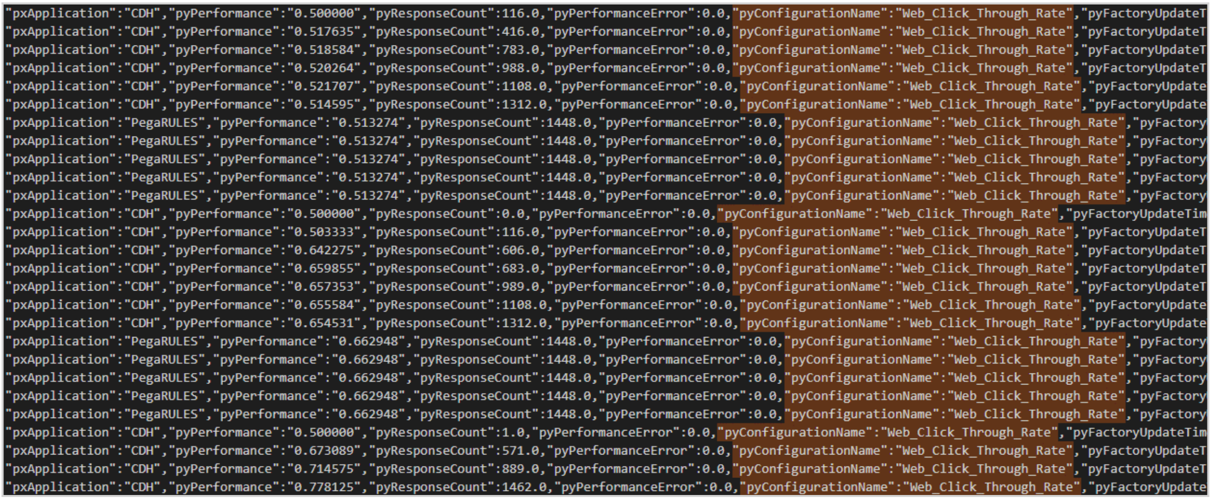
When .pyConfigurationName = "Web_Click_Through_Rate"

The *ADM predictor* data set does not contain the model configuration name, but it does contain the model ID. To select the predictor binning snapshots for the selected models, create a data set that only contains the selected model IDs. Add the new data set as a second destination to the *ADM snapshot data export* data flow.

The *ADM predictor data export* data flow exports the predictor binning snapshots. To filter only the relevant snapshots, merge the predictor binning snapshots and the data set that contains the selected model IDs that you created for this purpose.

Add a Convert component to match the two classes of the source data set, which is a prerequisite for the merge operation. The data flow merges the data sets on the condition that the model IDs match, and the destination data set only contains the relevant predictor binning snapshots.

The actual export of the monitoring data is typically done in the Business Operations Environment. The exported data in the repository contains only the model snapshots of interest.



```
"pxApplication":"CDH","pyPerformance":"0.500000","pyResponseCount":116.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.517635","pyResponseCount":416.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.518584","pyResponseCount":783.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.520264","pyResponseCount":988.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.521707","pyResponseCount":1108.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdate
"pxApplication":"CDH","pyPerformance":"0.514595","pyResponseCount":1312.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdate
"pxApplication":"PegaRULES","pyPerformance":"0.513274","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"PegaRULES","pyPerformance":"0.513274","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"PegaRULES","pyPerformance":"0.513274","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"PegaRULES","pyPerformance":"0.513274","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"PegaRULES","pyPerformance":"0.513274","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"CDH","pyPerformance":"0.500000","pyResponseCount":0.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateTim
"pxApplication":"CDH","pyPerformance":"0.503333","pyResponseCount":116.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.642275","pyResponseCount":606.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.659855","pyResponseCount":683.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.657353","pyResponseCount":989.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.655584","pyResponseCount":1108.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdate
"pxApplication":"CDH","pyPerformance":"0.654531","pyResponseCount":1312.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdate
"pxApplication":"PegaRULES","pyPerformance":"0.662948","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"PegaRULES","pyPerformance":"0.662948","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"PegaRULES","pyPerformance":"0.662948","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"PegaRULES","pyPerformance":"0.662948","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"PegaRULES","pyPerformance":"0.662948","pyResponseCount":1448.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactory
"pxApplication":"CDH","pyPerformance":"0.500000","pyResponseCount":1.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateTim
"pxApplication":"CDH","pyPerformance":"0.673089","pyResponseCount":571.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.714575","pyResponseCount":889.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdateT
"pxApplication":"CDH","pyPerformance":"0.778125","pyResponseCount":1462.0,"pyPerformanceError":0.0,"pyConfigurationName":"Web_Click_Through_Rate","pyFactoryUpdate
```

This demo has concluded. What did it show you?

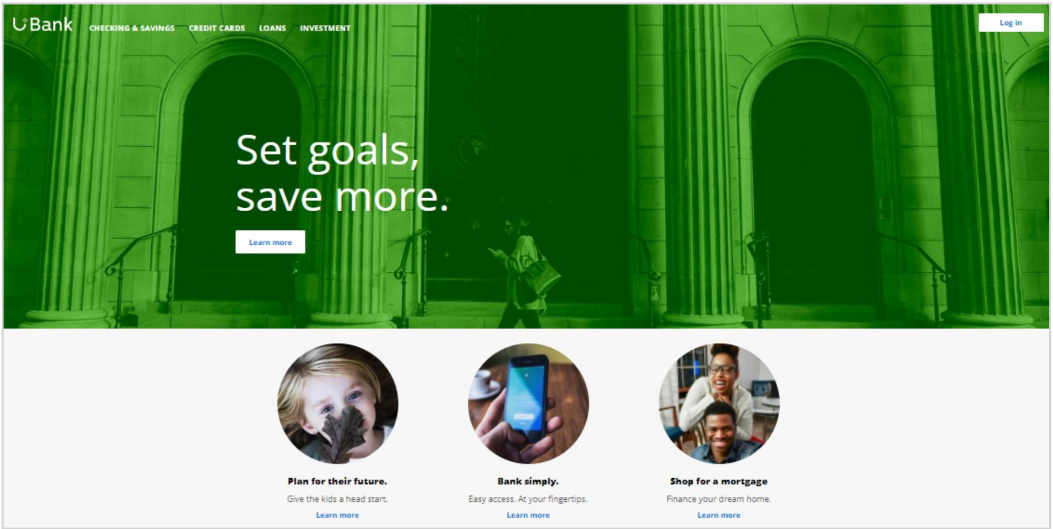
- How to generate the monitoring database export artifacts.
- How to configure the auto-generated data flows to export a subset of the ADM data mart.
- How to trigger the monitoring database export.

Data scientist tools for Customer Decision Hub

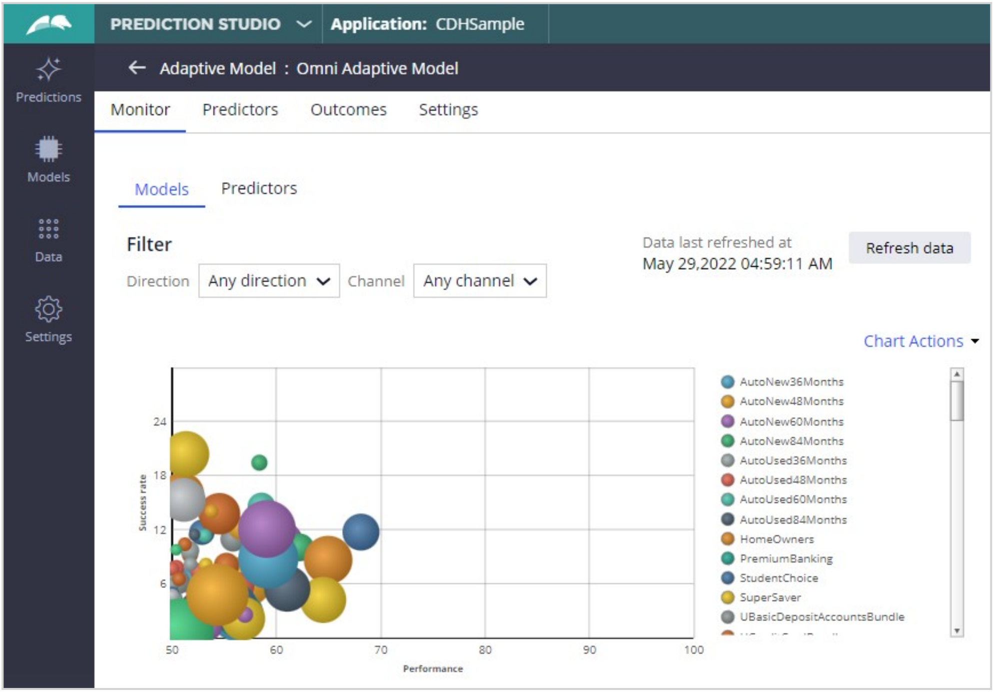
Pega has an open data model. Consequently, data scientists can export adaptive model data, predictor binning data, and historical data for further analysis. The open-source GitHub repository **Pega Data Scientist Tools** helps you to build meaningful plots and more.

Transcript

This video introduces you to the open-source tools that are available to data scientists that work on Pega Customer Decision Hub™ projects. Consider the following scenario: U+ Bank uses CDH to optimize customer interactions across multiple channels.

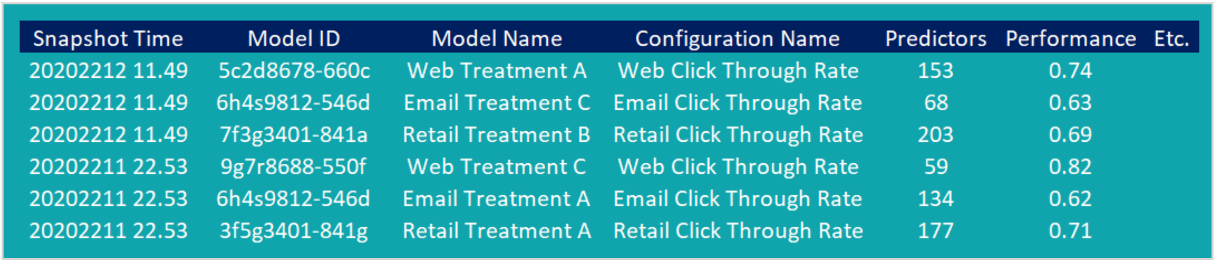


As a data scientist, you can monitor your adaptive models in Prediction Studio. The bubble chart shows you which models perform well, and which models do not.

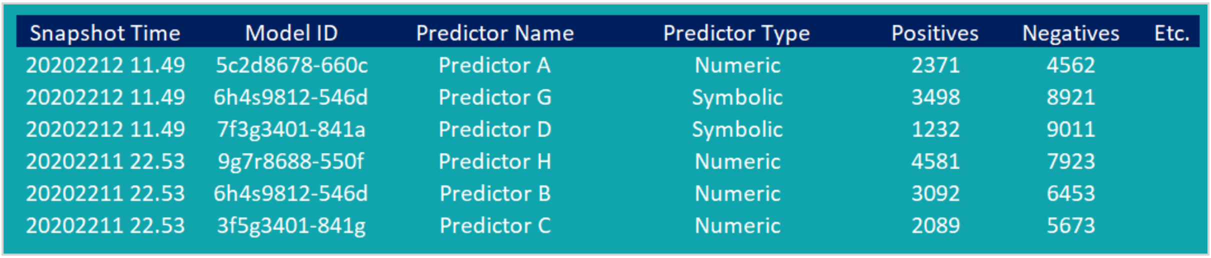


Direction: Any direction
Channel: Any channel
Data last refreshed at May 29,2022 04:59:11 AM

However, you might want to use a third-party analytical tool to do an in-depth analysis of the performance of the models and predictors. To do so, you export snapshots of the model data and the predictor data from your CDH system.



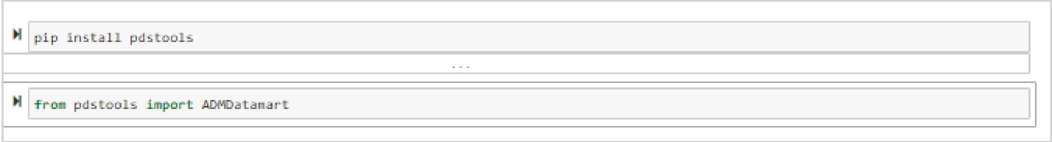
Snapshot Time	Model ID	Model Name	Configuration Name	Predictors	Performance	Etc.
20202212 11.49	5c2d8678-660c	Web Treatment A	Web Click Through Rate	153	0.74	
20202212 11.49	6h4s9812-546d	Email Treatment C	Email Click Through Rate	68	0.63	
20202212 11.49	7f3g3401-841a	Retail Treatment B	Retail Click Through Rate	203	0.69	
20202211 22.53	9g7r8688-550f	Web Treatment C	Web Click Through Rate	59	0.82	
20202211 22.53	6h4s9812-546d	Email Treatment A	Email Click Through Rate	134	0.62	
20202211 22.53	3f5g3401-841g	Retail Treatment A	Retail Click Through Rate	177	0.71	



Snapshot Time	Model ID	Predictor Name	Predictor Type	Positives	Negatives	Etc.
20202212 11.49	5c2d8678-660c	Predictor A	Numeric	2371	4562	
20202212 11.49	6h4s9812-546d	Predictor G	Symbolic	3498	8921	
20202212 11.49	7f3g3401-841a	Predictor D	Symbolic	1232	9011	
20202211 22.53	9g7r8688-550f	Predictor H	Numeric	4581	7923	
20202211 22.53	6h4s9812-546d	Predictor B	Numeric	3092	6453	
20202211 22.53	3f5g3401-841g	Predictor C	Numeric	2089	5673	

The **Pega Data Scientist Tools** GitHub repository provides utensils to analyze analytical data from a Pega decisioning system in R and Python.

To showcase the tools in Python, we'll use a Jupyter notebook. After installing the **PDS Tools** package, you can import the ADMDatamart class.



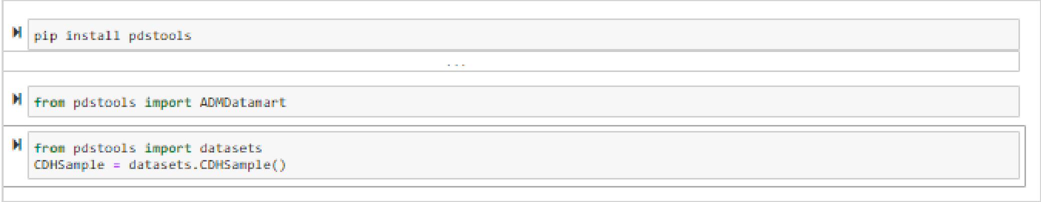
```
pip install pdstools
```

...

```
from pdstools import ADMDatamart
```

The ADMDatamart class orchestrates reading, preprocessing, and visualizing the data.

Import the **Model Snapshots** and **ADM Predictor Snapshots** data sets that you exported from your Pega system to your directory, and then initialize them in an ADMDatamart class. For this demo, we use sample data.



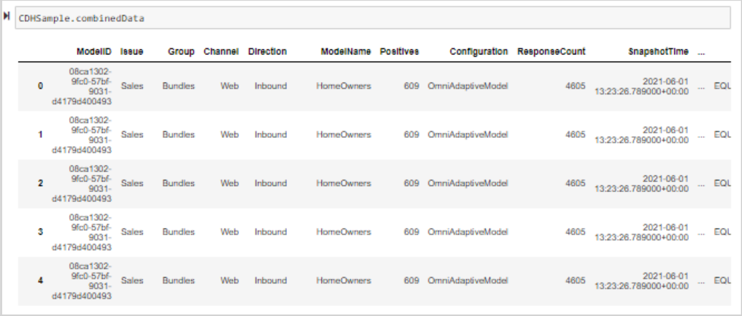
```
pip install pdstools
```

...

```
from pdstools import ADMDatamart
```

```
from pdstools import datasets
CDHSample = datasets.CDHSample()
```

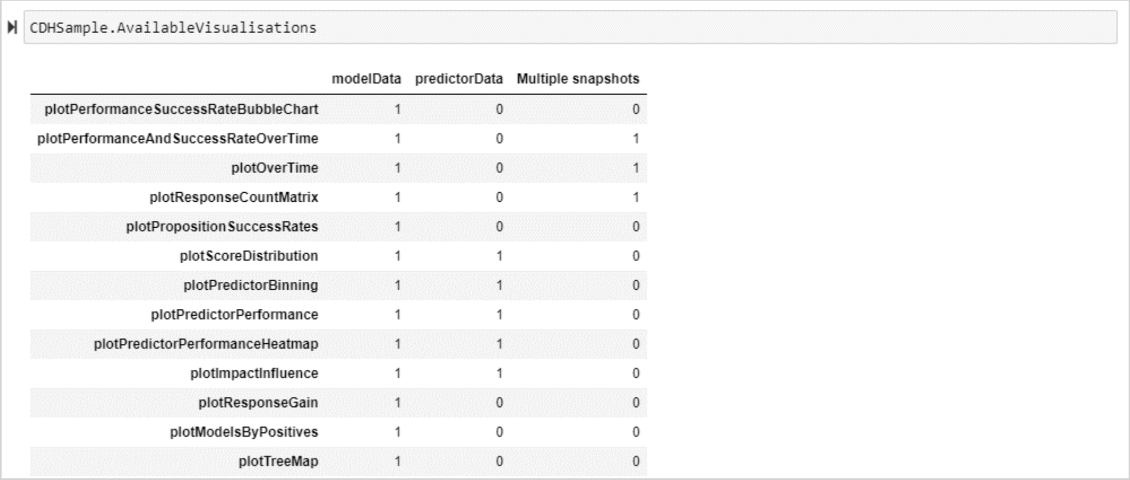
You can access your model data and your predictor data as data frames. When both data sources are present, the ADMDatamart class combines them in the background, and you can inspect the resulting data frame.



```
CDHSample.combinedData
```

	ModelID	Issue	Group	Channel	Direction	ModelName	Positives	Configuration	ResponseCount	SnapshotTime	...	
0	08ca1302-9fc0-57bf-9031-d4179d400493	Sales	Bundles	Web	Inbound	HomeOwners	609	OmniAdaptiveModel	4605	2021-06-01 13:23:26.789000+00:00	...	EQL
1	08ca1302-9fc0-57bf-9031-d4179d400493	Sales	Bundles	Web	Inbound	HomeOwners	609	OmniAdaptiveModel	4605	2021-06-01 13:23:26.789000+00:00	...	EQL
2	08ca1302-9fc0-57bf-9031-d4179d400493	Sales	Bundles	Web	Inbound	HomeOwners	609	OmniAdaptiveModel	4605	2021-06-01 13:23:26.789000+00:00	...	EQL
3	08ca1302-9fc0-57bf-9031-d4179d400493	Sales	Bundles	Web	Inbound	HomeOwners	609	OmniAdaptiveModel	4605	2021-06-01 13:23:26.789000+00:00	...	EQL
4	08ca1302-9fc0-57bf-9031-d4179d400493	Sales	Bundles	Web	Inbound	HomeOwners	609	OmniAdaptiveModel	4605	2021-06-01 13:23:26.789000+00:00	...	EQL

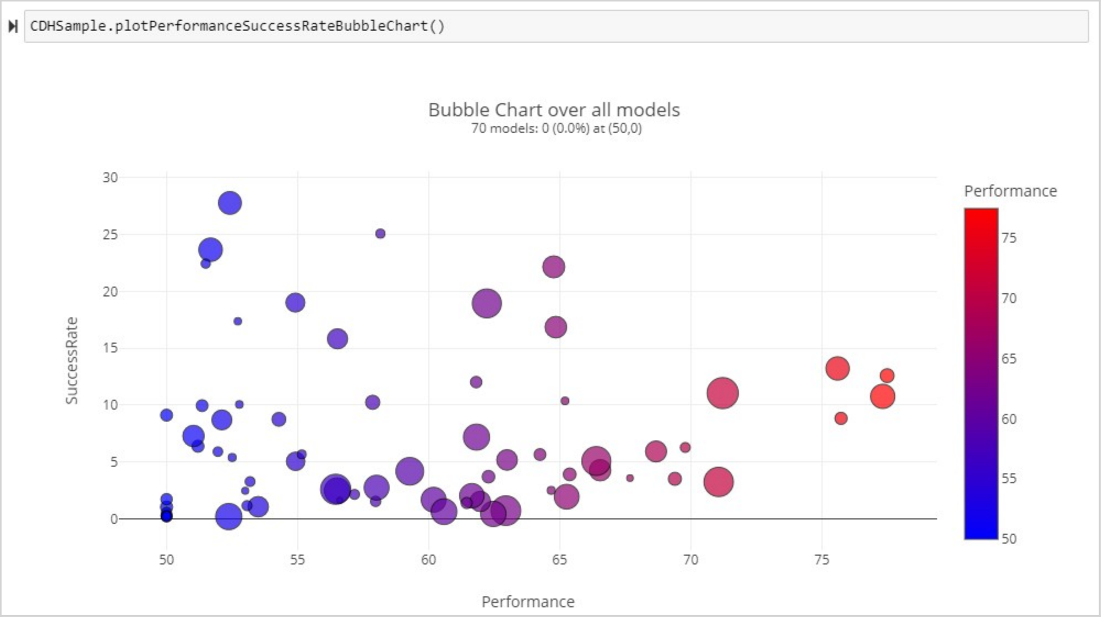
A collection of sample plots and graphs are available for analysis of the adaptive models operating in CDH.



```
CDHSample.AvailableVisualisations
```

	modelData	predictorData	Multiple snapshots
plotPerformanceSuccessRateBubbleChart	1	0	0
plotPerformanceAndSuccessRateOverTime	1	0	1
plotOverTime	1	0	1
plotResponseCountMatrix	1	0	1
plotPropositionSuccessRates	1	0	0
plotScoreDistribution	1	1	0
plotPredictorBinning	1	1	0
plotPredictorPerformance	1	1	0
plotPredictorPerformanceHeatmap	1	1	0
plotImpactInfluence	1	1	0
plotResponseGain	1	0	0
plotModelsByPositives	1	0	0
plotTreeMap	1	0	0

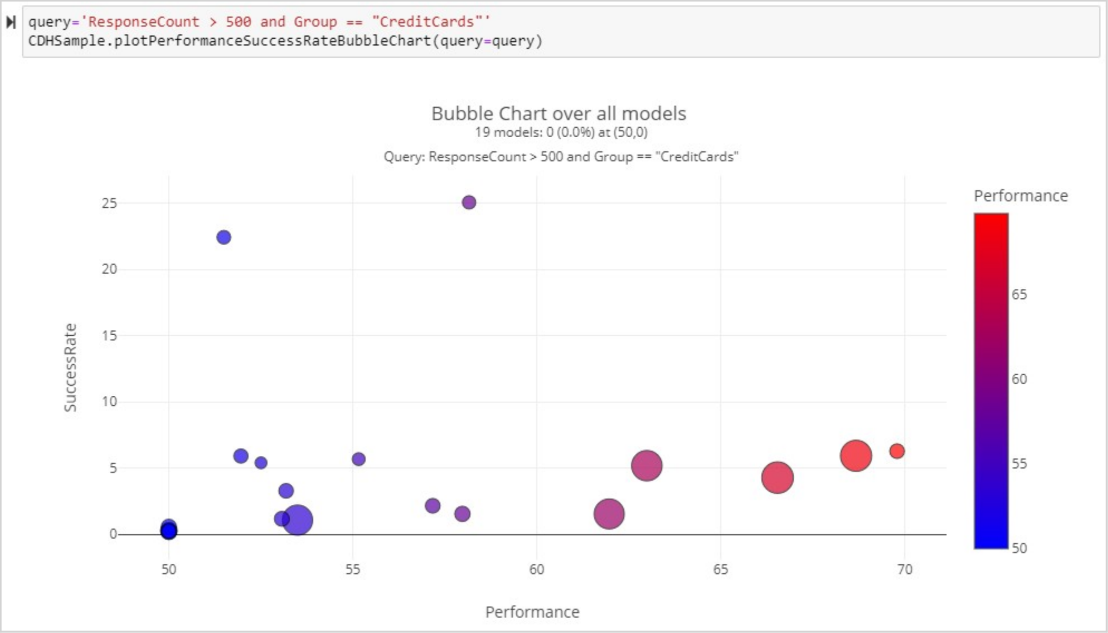
All visualizations need the model data. Some visualizations also need predictor data, and still others need multiple snapshots to create timelines. One of the visualizations is a bubble chart that plots performance versus success rate, similar to the bubble chart available in Prediction Studio. The visualization considers the latest snapshot by default.



```
CDHSample.plotPerformanceSuccessRateBubbleChart()
```

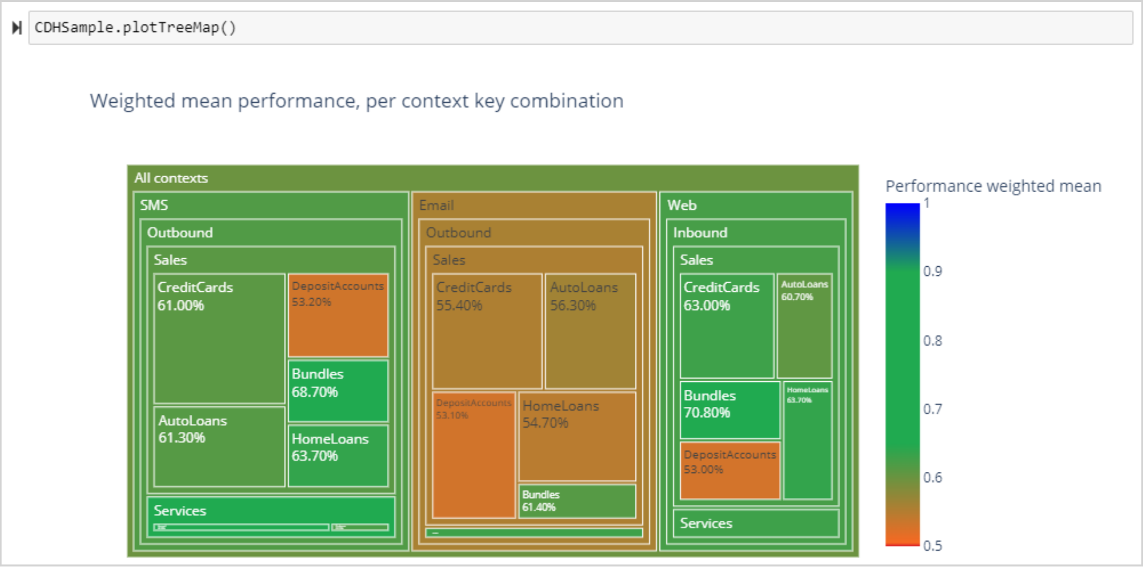
Zoom in on models that perform very well but have a low success rate, to report actions that need attention to the business.

You can visualize a subset of the data by supplying a query argument. Let's only consider the models with a high response count within the CreditCards group.



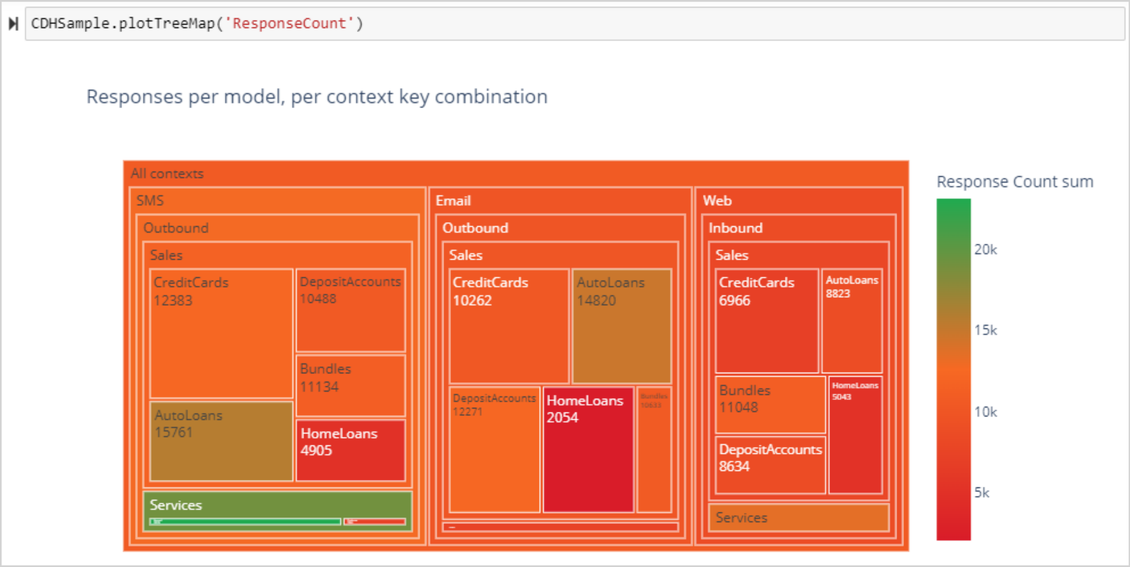
```
query='ResponseCount > 500 and Group == "CreditCards"'
CDHSample.plotPerformanceSuccessRateBubbleChart(query=query)
```

The bubble chart shows you which models perform well. You might, however, want to know if performance issues occur in a specific channel, issue, or group. The Treemap visualization offers insight in this situation.



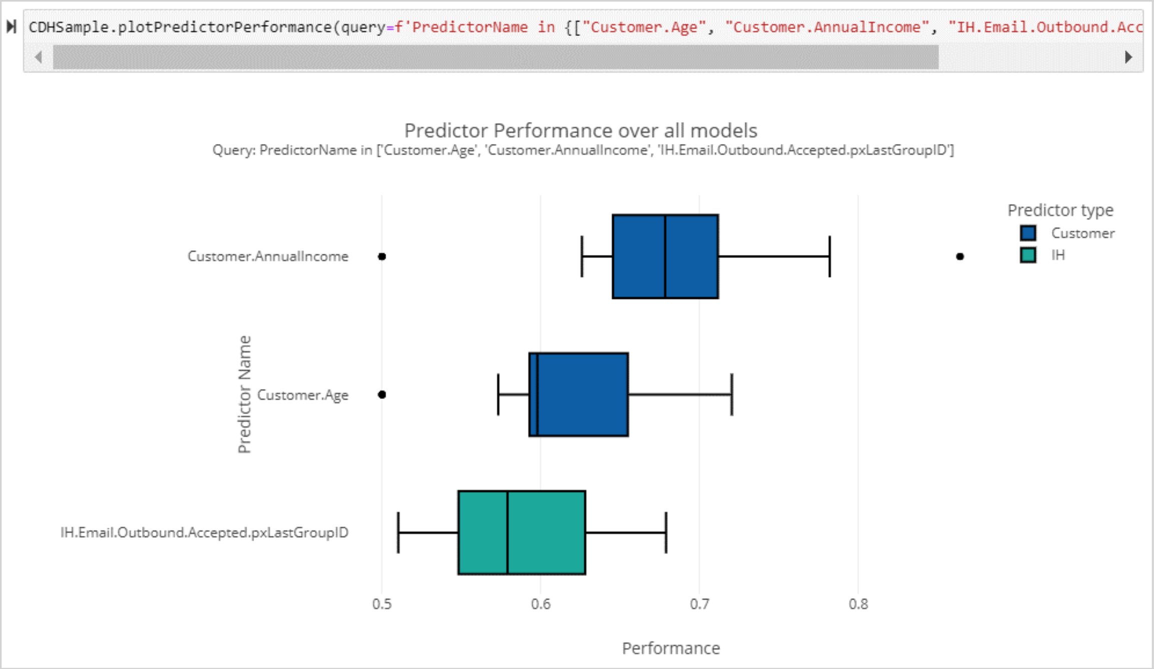
```
CDHSample.plotTreeMap()
```

By default, the Treemap shows the performance, weighted by the response count. The number of model IDs within a combination of context keys determines the size of the squares. Besides performance, you can also use another variable, such as the SuccessRate or ResponseCount.



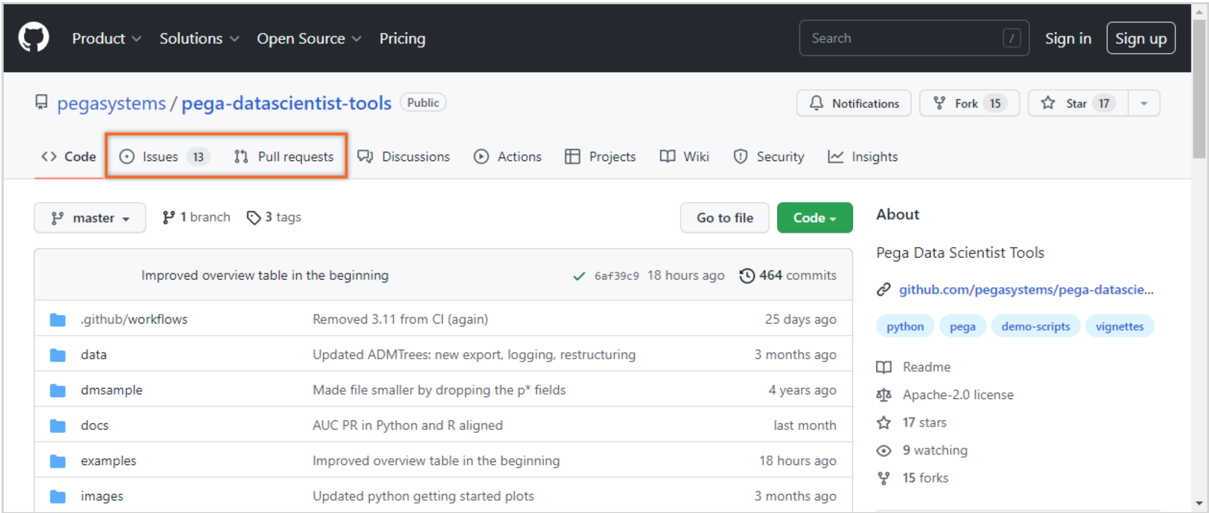
```
CDHSample.plotTreeMap('ResponseCount')
```

You might want to look at the performance of specific predictors over multiple models. To make the visualization more legible, limit the number of predictors.



```
CDHSample.plotPredictorPerformance(query=f'PredictorName in {["Customer.Age", "Customer.AnnualIncome", "IH.Email.Outbound.Acc
```

The GitHub repository PDS Tools is open source. You can therefore contribute to the repository by creating a pull request. You can also report problems by creating an issue on the main GitHub page.



pegasystems / pega-datascientist-tools Public

Notifications | Fork 15 | Star 17

Code | Issues 13 | Pull requests | Discussions | Actions | Projects | Wiki | Security | Insights

master | 1 branch | 3 tags | Go to file | Code

Improved overview table in the beginning ✓ 6af39c9 18 hours ago 464 commits

.github/workflows	Removed 3.11 from CI (again)	25 days ago
data	Updated ADMTrees: new export, logging, restructuring	3 months ago
dmsample	Made file smaller by dropping the p* fields	4 years ago
docs	AUC PR in Python and R aligned	last month
examples	Improved overview table in the beginning	18 hours ago
images	Updated python getting started plots	3 months ago

About

Pega Data Scientist Tools

github.com/pegasystems/pega-datascie...

python pega demo-scripts vignettes

Readme
Apache-2.0 license
17 stars
9 watching
15 forks

This demo has concluded. What did it show you?

- How to use model data and predictor data from your CDH system with PDS Tools.
- How to inspect the data frame that combines model and predictor data.
- How to visualize a subset of the data with PDS Tools.

Creating predictions

Description

Predicting customer churn is a crucial challenge, as losing customers can significantly impact profitability. To address this requirement, you can use a scorecard or a predictive model to drive a churn prediction. Learn how to connect to models in Amazon SageMaker and Google Vertex AI machine learning services, as well as how to challenge the active model with a candidate model.

Learning objectives

- Build a scorecard to drive a churn prediction in Prediction Studio.
- Create a new prediction that uses a predictive model.
- Connect to machine learning services.
- Challenge a predictive model.

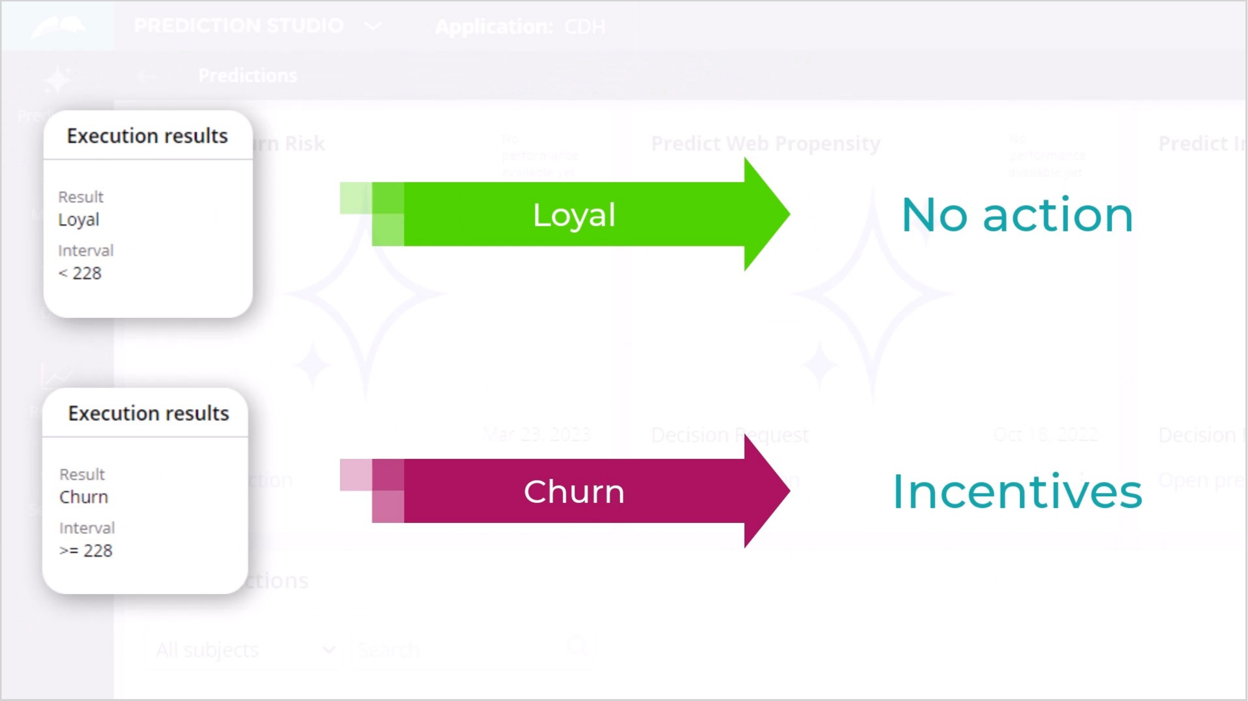
Creating a churn prediction using a scorecard

Predicting customer churn is a crucial challenge, as losing customers can significantly impact profitability. To address this requirement, you can use a scorecard. A scorecard is a transparent predictive model that can drive a churn prediction.

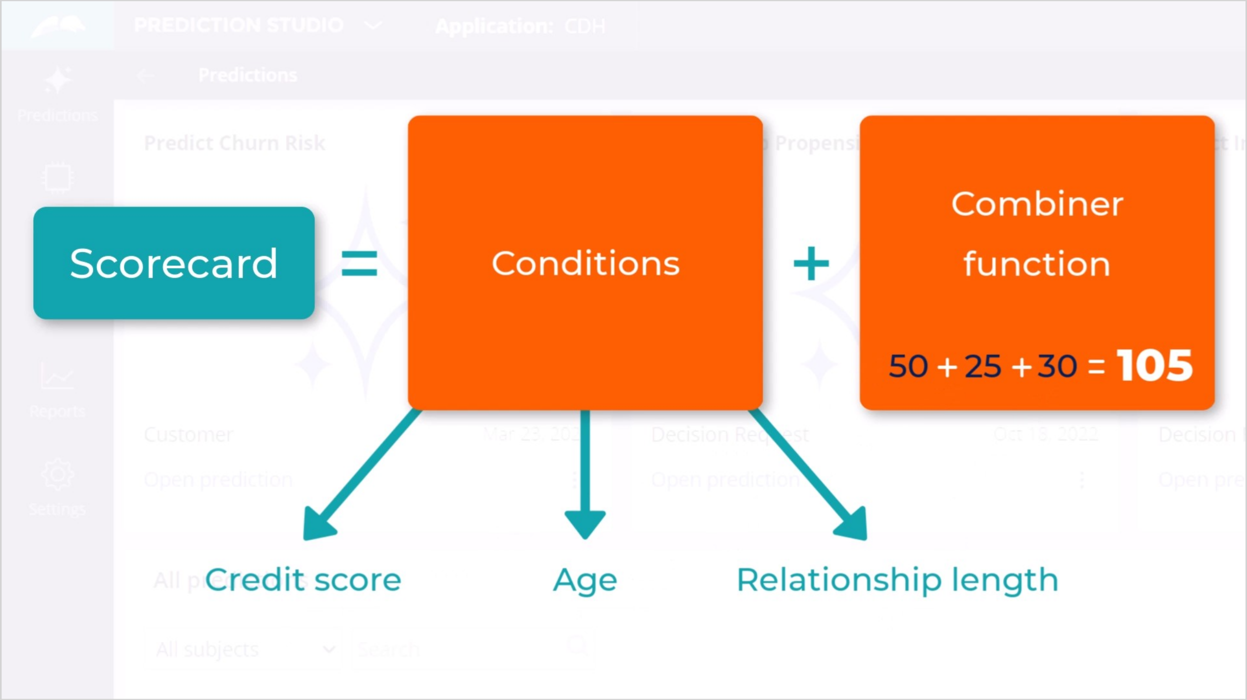
Transcript

In this video, you will explore the use of scorecards to drive predictions in Pega Customer Decision Hub™.

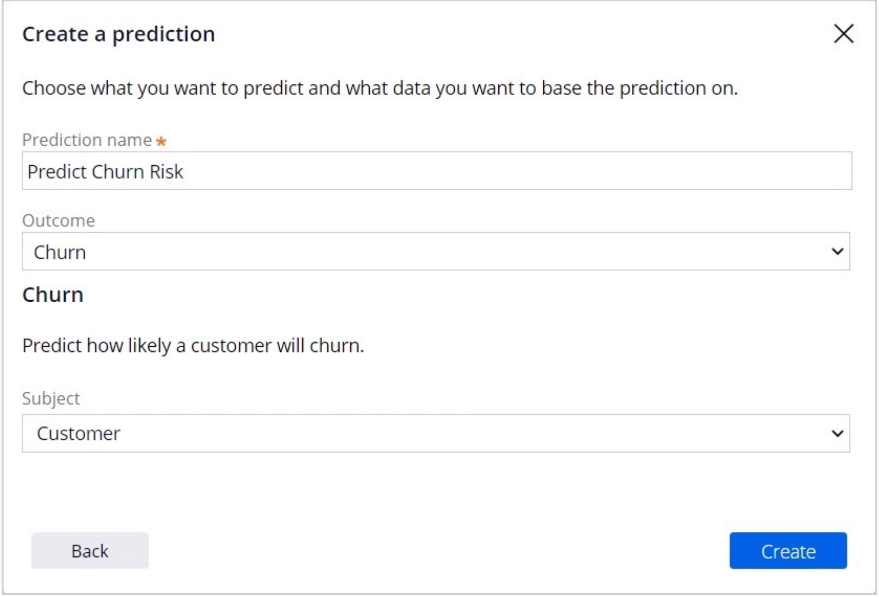
U+Bank implements Pega Customer Decision Hub to optimize customer engagement on the banks' website. The way they reduce the number of customers leaving the bank is through a churn prediction created by a data scientist in Prediction Studio. This prediction is then used in engagement policies, allowing the bank to proactively offer incentives to customers that are likely to leave.



Predictive models, pre-calculated fields, and scorecards can all drive a prediction. A scorecard is a transparent predictive model that assigns a score to each customer based on specific conditions for each predictor. Customers receive points based on these conditions, and the score is calculated using a combiner function, for example, by summing up the points.



To use a scorecard, you create a Customer Decision Hub prediction in Prediction Studio.



Create a prediction

Choose what you want to predict and what data you want to base the prediction on.

Prediction name *: Predict Churn Risk

Outcome: Churn

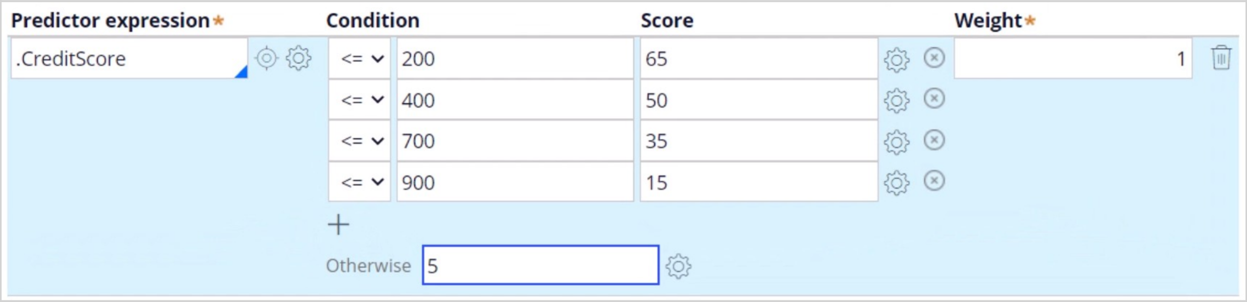
Churn

Predict how likely a customer will churn.

Subject: Customer

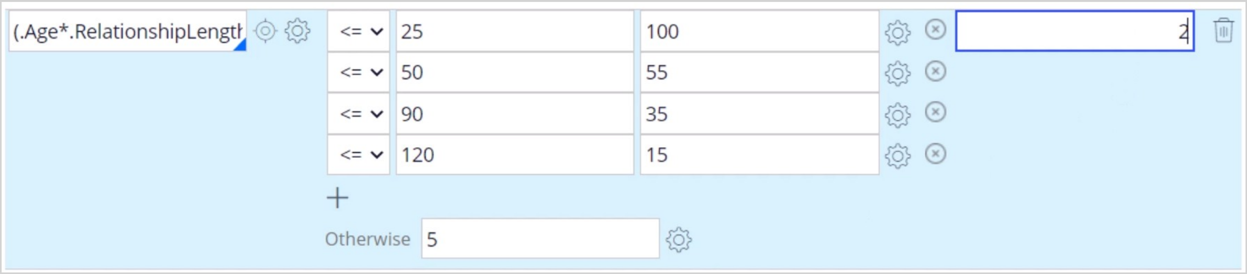
Default churn prediction includes an out-of-the-box template for a scorecard, that you can edit from the Models tab. Let's add the first predictor field - **.CreditScore**.

For the numerical predictor **.CreditScore**, customers with a credit score below or equal to 200 will get 65 score points, between 200 and 400, or equal to 400, will get 50 score points, and so on. Lower credit scores may indicate a higher risk of leaving, as customers may have more difficulty obtaining loans or other financial products from the bank.



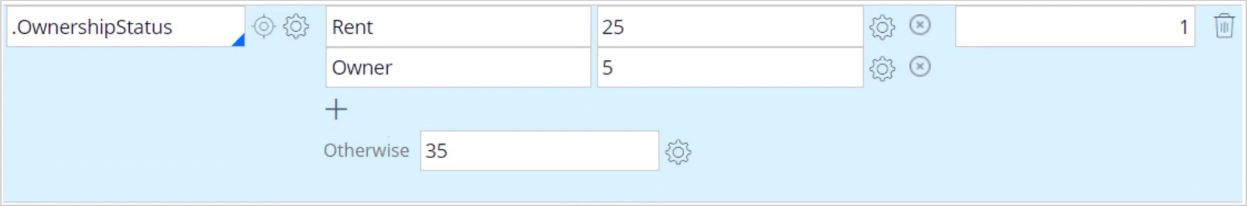
Predictor expression*	Condition		Score	Weight*
.CreditScore	<=	200	65	1
	<=	400	50	
	<=	700	35	
	<=	900	15	
	Otherwise	5		

Age and length of the customer relationship are important factors in credit risk assessment. The scorecard allows for complex expressions that can involve multiple predictors and calculations. We use both the **.Age** and **.RelationshipLengthDays** predictors to create a predictor field expression. The score is calculated by multiplying the predictor values and dividing the result by 100. Younger customers and those with shorter relationships receive more points, as they may be less financially stable and more likely to churn. You can set a weight to this predictor to assign its relative importance.



(.Age*.RelationshipLengtl	<=	25	100	2
	<=	50	55	
	<=	90	35	
	<=	120	15	
	Otherwise	5		

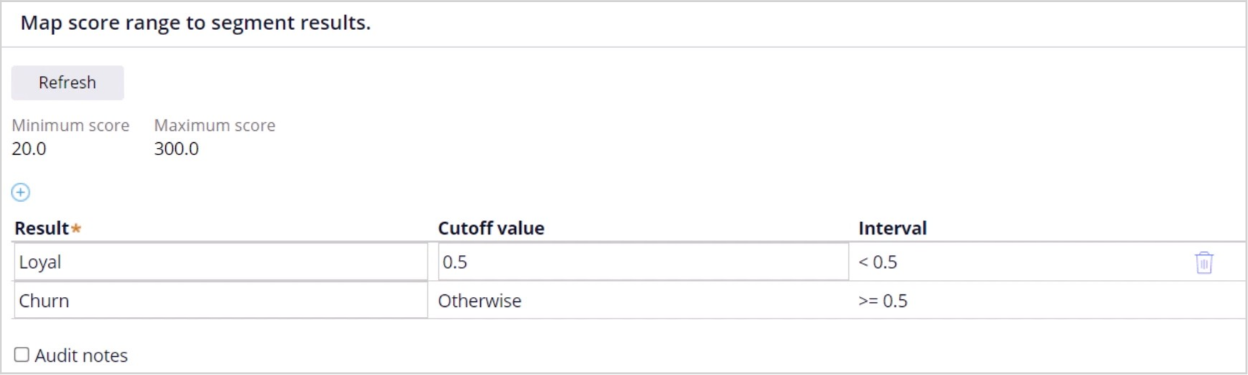
The scorecard also allows you to add categorical predictors, where you can assign a score for an individual value. For the categorical predictor **.OwnershipStatus**, a customer that meets the condition to be a house owner scores less points, as house ownership may indicate financial maturity, and a low propensity to churn.



.OwnershipStatus	Rent	25	1
	Owner	5	
	Otherwise 35		

The Combiner function enables you to select a method for combining scores, where in this scorecard, the points assigned for each predictor are summed.

On the Results tab, you can map the Cutoff value to distinguish potential churners from loyal customers. In this case, the scorecard predicts that customers with less than 122 points are likely to remain loyal to U+Bank, while customers with higher points are likely to churn.



Map score range to segment results.

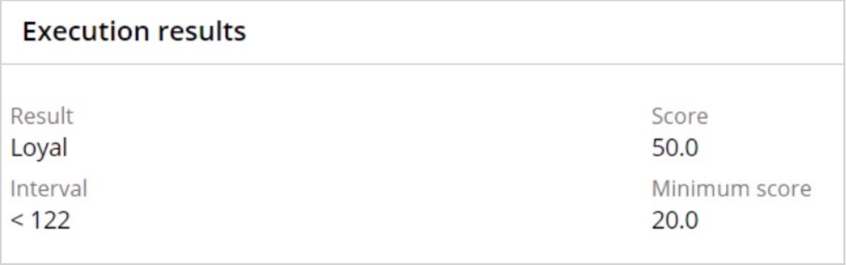
Minimum score
20.0

Maximum score
300.0

Result*	Cutoff value	Interval
Loyal	0.5	< 0.5
Churn	Otherwise	>= 0.5

□ Audit notes

To test the scorecard, you can apply a data transform to run it for different customers. For example, Barbara. In the execution details section, you can see the points assigned for each predictor field. Note the **.Age** and **.RelationshipLengthDays** predictor expression. The final points double the score because of the weight value. The combiner function sums up the points to give Barbara a score of 50, indicating that she is a loyal customer.



Execution results

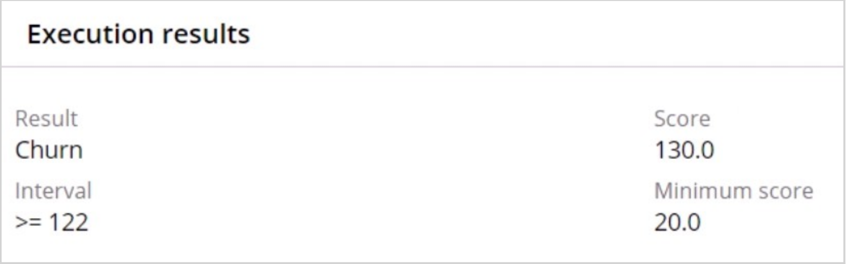
Result
Loyal

Score
50.0

Interval
< 122

Minimum score
20.0

Next, we run the scorecard for Robert. Robert's score is 130, which suggests that he is likely to churn and should be targeted with retention offers.



Execution results

Result
Churn

Score
130.0

Interval
>= 122

Minimum score
20.0

You have reached the end of this video. You have learned:

- How to create a prediction in Pega Customer Decision Hub to predict the risk of churning.
- How to build a scorecard to drive the churn prediction in Prediction Studio.

Predictive models

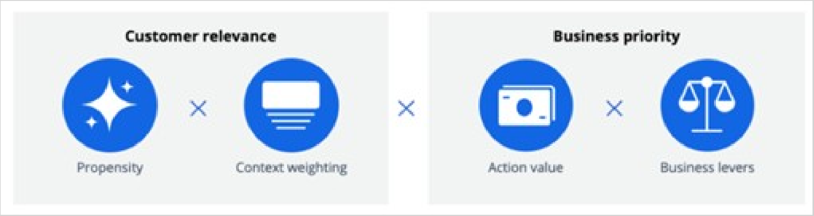
Introduction

Enhance decision strategies with predictive models built on customer interaction data and let Pega Customer Decision Hub™ bring even more relevance to every customer engagement. Build models using Pega's machine learning capabilities, import models built with third-party tools, and incorporate the latest AI algorithms into the Pega AI engine by connecting to the Google Vertex AI and Amazon SageMaker machine learning services.

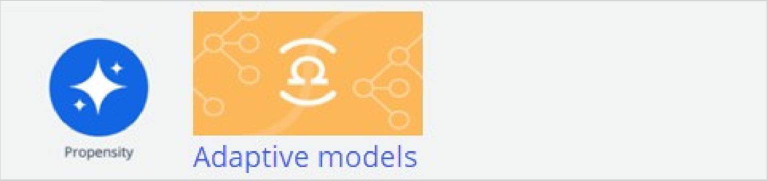
Transcript

This video will describe the use of predictive models to enhance the next best actions that Customer Decision Hub generates.

Next best actions balance customer relevance and business priorities by selecting the actions with the highest priority. The priority is calculated by multiplying the values for propensity, context weighting, action value, and business levers.



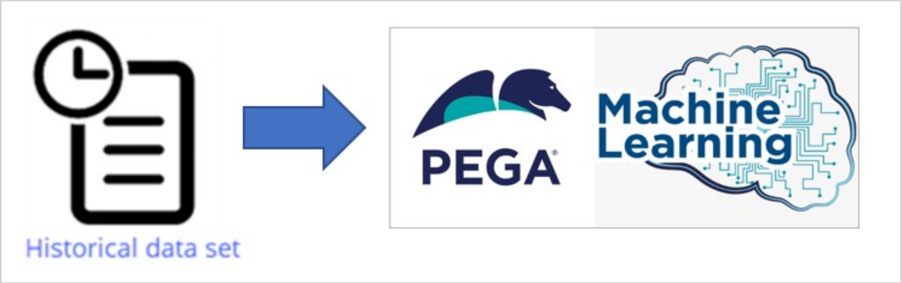
Propensity is the likelihood of a customer responding positively to an action by, for example, clicking on a web banner or accepting an offer. This is calculated by predictive models. In Pega, self-learning Naive Bayes models, which are generated for each action, are a key feature.



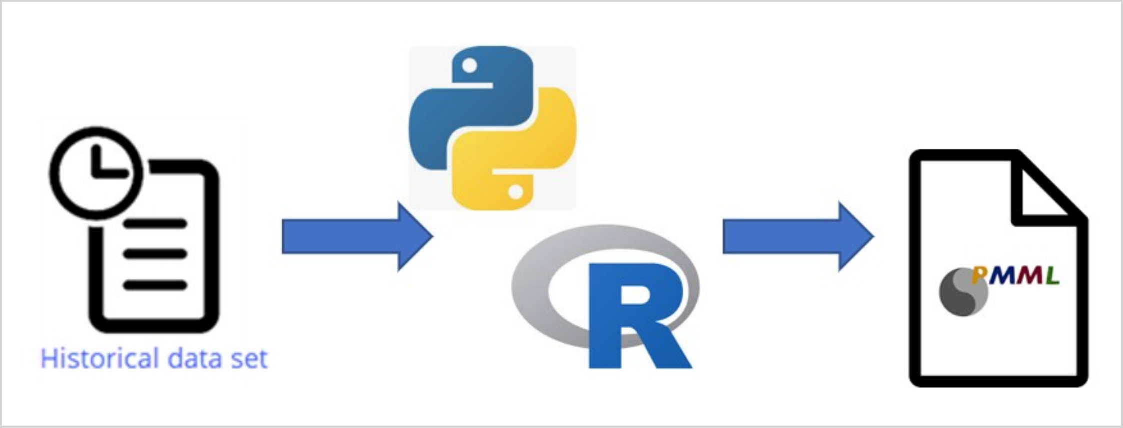
These adaptive models are automatically updated after new responses have been received and can start without any historical information because they learn on the fly.

When the use case requires a more advanced modeling technique, for example to predict customer churn or to estimate credit risk, Prediction Studio offers several methods to create the artifacts that represent an actual predictive model or that reference a predictive model.

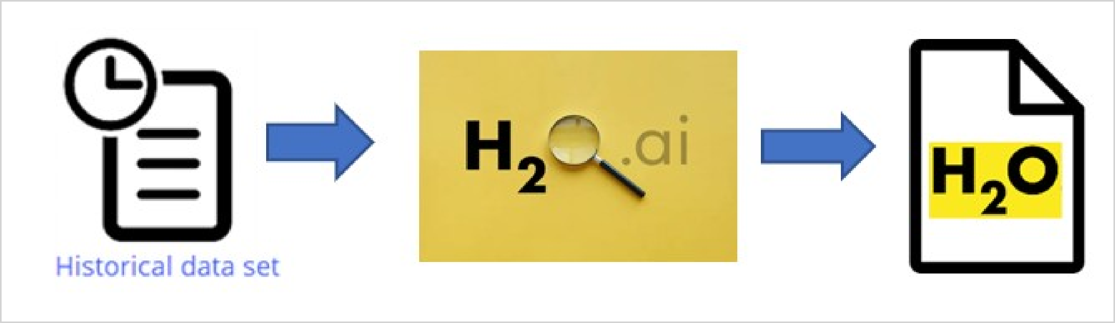
The first method is to use Pega machine learning. You can import a file containing the historical customer interaction data set and build a model in Prediction Studio. This model can then be used in decision strategies. When the decision strategies execute, the models are executed inside the Pega platform.



The second option is to import an existing model. You can build a model using a third-party tool like R or Python and export it as a PMML file. PMML is an XML-based standard that is designed to facilitate the exchange of models between applications. Import the PMML file into Prediction Studio and map its predictors to the fields in the customer data model.



Similarly, you can import model files that have been generated in H2O.ai. H2O is a modeling platform, and the procedure for using the generated model file is identical to that for a PMML file.



Just like with Pega machine learning models, the imported model can then be used in decision strategies. When decision strategies using the imported models execute, the models are executed inside the Pega platform.

The third option is to reference a model on an external platform like the Google Vertex AI. Just like with Pega machine learning models, the referenced model can then be used in decision strategies. In this case, when the decision strategy requires a prediction, a request is sent to the external model, which calculates the outcome and sends it back to Pega. Like with the Google Vertex AI platform, you can connect to AWS SageMaker and run your model remotely.

To summarize, you have three options for leveraging predictive models built on customer data. You can build models using Pega machine learning, you can import models built with third-party tools, and you can use machine learning services to reference predictive models. When the decision strategies using predictive models execute, the models are executed inside Pega or externally by Google Vertex AI and the Amazon SageMaker platform.

Creating a Prediction using an ML model

Introduction

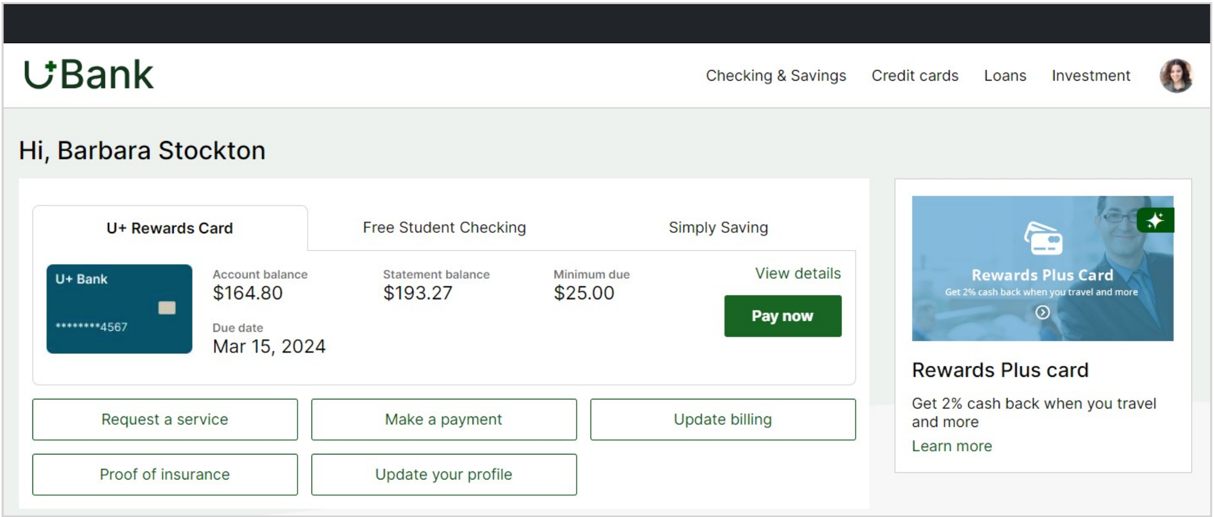
U+ Bank uses Pega Customer Decision Hub™ to handle their customer engagement. Acquiring new customers can be more costly than retaining current customers, so now they aim to reduce the churn rate.

Learn how to create a new Prediction in Prediction Studio that calculates the likelihood that a customer might churn soon.

Transcript

This demo shows you how to create a new Prediction in Prediction Studio.

U+ Bank uses Pega Customer Decision Hub™ to personalize the credit card offered to customers on their website.



Hi, Barbara Stockton

Account balance $164.80

Statement balance $193.27

Minimum due $25.00

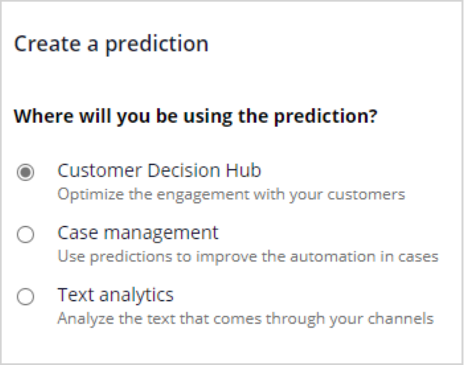
Due date Mar 15, 2024

Rewards Plus card

Get 2% cash back when you travel and more

For customers who are likely to leave the bank soon, like Troy, the bank wants to make a proactive retention offer instead of a credit card offer. To predict the likelihood that a customer may leave the bank soon, you use a predictive model to drive a new Prediction. Predictions are managed in Prediction Studio.

You use Customer Decision Hub Predictions to improve customer engagement. Predictions for case automation and text analytics are also available.



Create a prediction

Where will you be using the prediction?

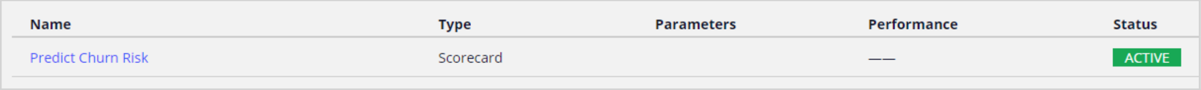
● Customer Decision Hub
Optimize the engagement with your customers

○ Case management
Use predictions to improve the automation in cases

○ Text analytics
Analyze the text that comes through your channels

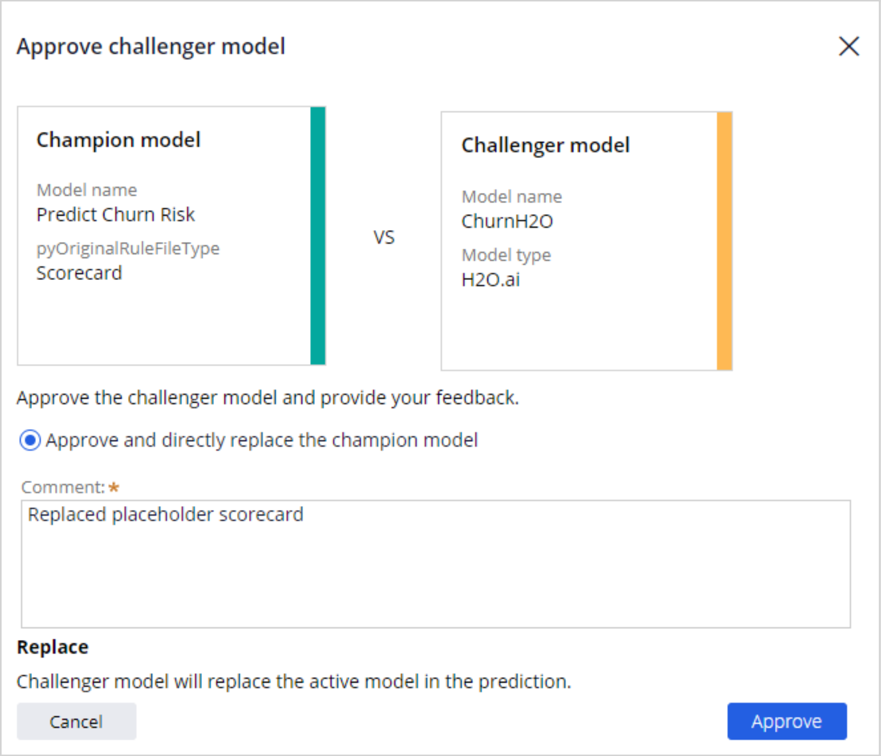
To create a Prediction that aims to calculate the likelihood that a customer might churn soon, set the outcome to **Churn** and the subject of the Prediction to **Customer**.

Notice that initially, a placeholder scorecard is generated and used to drive the new Prediction. This placeholder is useful in case you do not have a predictive model yet, as it allows the Next-Best-Action Designer to continue work while a predictive model is built.



Name	Type	Parameters	Performance	Status
Predict Churn Risk	Scorecard		——	ACTIVE

As you already have a predictive churn model, the next step is to replace the placeholder scorecard with your model. When the replacement is ready for review, approve the candidate model.



Approve challenger model

Champion model

Model name
Predict Churn Risk

pyOriginalRuleFileType
Scorecard

VS

Challenger model

Model name
ChurnH2O

Model type
H2O.ai

Approve the challenger model and provide your feedback.

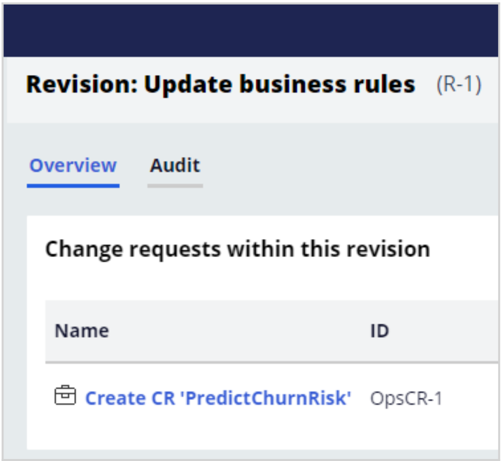
● Approve and directly replace the champion model

Comment:*
Replaced placeholder scorecard

Replace

Challenger model will replace the active model in the prediction.

When you submit the new Prediction for deployment, a change request is automatically created in the current revision.



Revision: Update business rules (R-1)

Change requests within this revision

Name	ID
Create CR 'PredictChurnRisk'	OpsCR-1

The changes come into effect after the Revision Manager deploys the revision. Changes from the Business Operations environment are pushed to the Development environment, and from there to other environments through the Business Change pipeline.

You have reached the end of this demo. What did it show you?

- How to create a new Prediction.
- How to replace the generated scorecard with a predictive model.
- How the new Prediction is deployed.

Using machine learning services

Introduction

Enhance the Pega AI engine with the latest AI algorithms by connecting to models in Amazon SageMaker and Google Vertex AI machine learning services. Learn how to leverage a model, created in and running on Amazon SageMaker, in Pega's Prediction Studio.

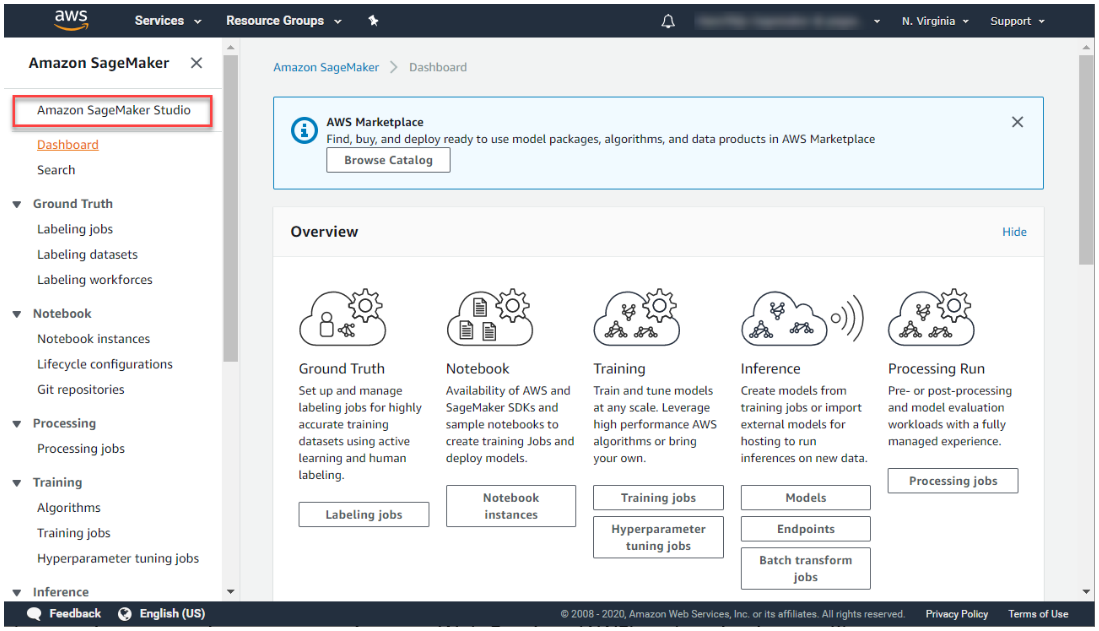
Transcript

This demo will show you how to leverage a machine learning service by running a churn model created externally and using its outputs in Pega Prediction Studio.

We will showcase this using Amazon SageMaker. Using a machine learning service instead of a model that runs locally may involve costs and possible down time of the service.

However, for certain use cases such as churn or credit risk models, machine learning services can be the optimal choice. To showcase how to use a churn model created in Amazon SageMaker, let's first consider the high-level steps involved in creating a machine learning model.

Amazon SageMaker allows you to build, train and deploy machine learning models in a fully managed service. The Autopilot feature automates this process and trains and tunes the best machine learning models for classification or regression, based on your data. After setting up your AWS environment, you can open Amazon SageMaker Studio to create a new Autopilot experiment.



AWS Marketplace
Find, buy, and deploy ready to use model packages, algorithms, and data products in AWS Marketplace

Ground Truth: Set up and manage labeling jobs for highly accurate training datasets using active learning and human labeling.

Notebook: Availability of AWS and SageMaker SDKs and sample notebooks to create training Jobs and deploy models.

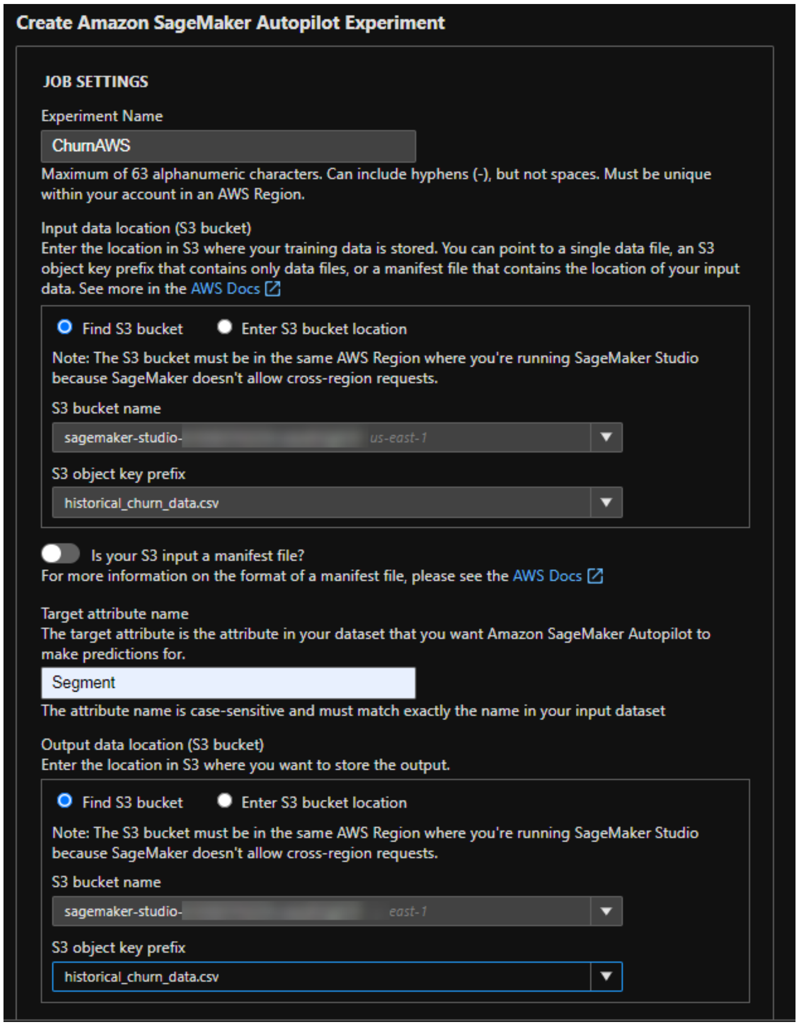
Training: Train and tune models at any scale. Leverage high performance AWS algorithms or bring your own.

Inference: Create models from training jobs or import external models for hosting to run inferences on new data.

Processing Run: Pre- or post-processing and model evaluation workloads with a fully managed experience.

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

In the Job settings, select the data file you want to build the model on, specify the outcome field, choose the location where the output should be stored and create the experiment.



Create Amazon SageMaker Autopilot Experiment

JOB SETTINGS

Experiment Name

ChurnAWS

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Input data location (S3 bucket)
Enter the location in S3 where your training data is stored. You can point to a single data file, an S3 object key prefix that contains only data files, or a manifest file that contains the location of your input data. See more in the AWS Docs

● Find S3 bucket ○ Enter S3 bucket location

Note: The S3 bucket must be in the same AWS Region where you're running SageMaker Studio because SageMaker doesn't allow cross-region requests.

S3 bucket name

sagemaker-studio-[redacted]us-east-1 ▼

S3 object key prefix

historical_churn_data.csv ▼

○ Is your S3 input a manifest file?
For more information on the format of a manifest file, please see the AWS Docs

Target attribute name
The target attribute is the attribute in your dataset that you want Amazon SageMaker Autopilot to make predictions for.

Segment

The attribute name is case-sensitive and must match exactly the name in your input dataset

Output data location (S3 bucket)
Enter the location in S3 where you want to store the output.

● Find S3 bucket ○ Enter S3 bucket location

Note: The S3 bucket must be in the same AWS Region where you're running SageMaker Studio because SageMaker doesn't allow cross-region requests.

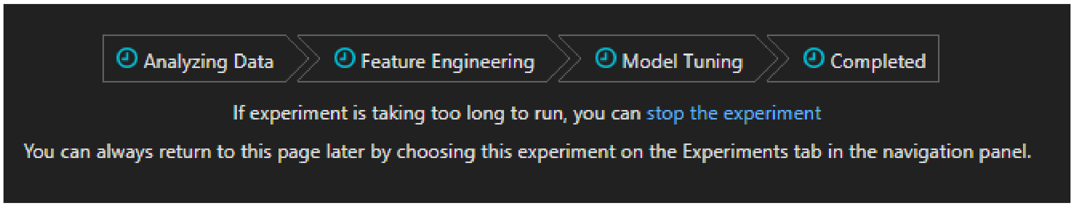
S3 bucket name

sagemaker-studio-[redacted]east-1 ▼

S3 object key prefix

historical_churn_data.csv ▼

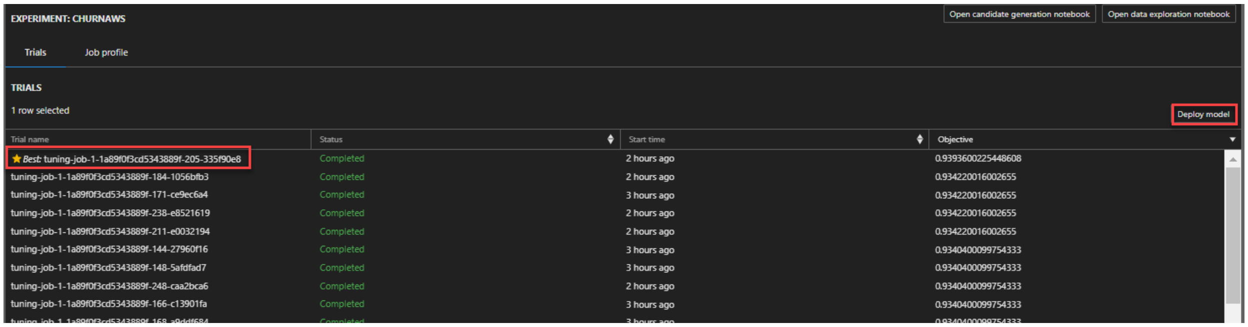
The Autopilot process analyzes the data, performs a feature engineering step, and tunes the candidate models.



If experiment is taking too long to run, you can stop the experiment

You can always return to this page later by choosing this experiment on the Experiments tab in the navigation panel.

To deploy the best candidate model, select the tuning job with the highest Objective value. This value indicates the predictive power of the model.


Open candidate generation notebook
Open data exploration notebook
Trials
Job profile
Deploy model

EXPERIMENT: CHURNAWS

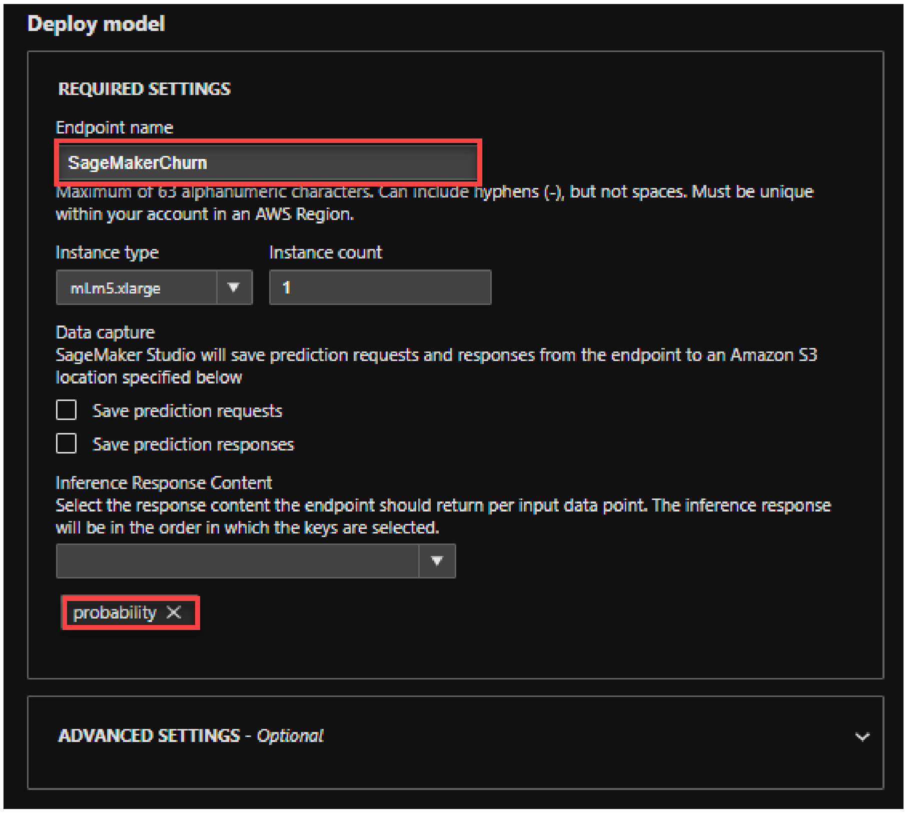
TRIALS

1 row selected

Trial name	Status	Start time	Objective
★ *Best:* tuning-job-1-1a89f0f3cd5343889f-205-335f90e8	Completed	2 hours ago	0.9393600225448608
tuning-job-1-1a89f0f3cd5343889f-184-1056bfb3	Completed	2 hours ago	0.934220016002655
tuning-job-1-1a89f0f3cd5343889f-171-ce9ec6a4	Completed	3 hours ago	0.934220016002655
tuning-job-1-1a89f0f3cd5343889f-238-e8521619	Completed	2 hours ago	0.934220016002655
tuning-job-1-1a89f0f3cd5343889f-211-e0032194	Completed	2 hours ago	0.934220016002655
tuning-job-1-1a89f0f3cd5343889f-144-27960f16	Completed	3 hours ago	0.9340400099754333
tuning-job-1-1a89f0f3cd5343889f-148-5afdfad7	Completed	3 hours ago	0.9340400099754333
tuning-job-1-1a89f0f3cd5343889f-248-caa2bca6	Completed	2 hours ago	0.9340400099754333
tuning-job-1-1a89f0f3cd5343889f-166-c13901fa	Completed	3 hours ago	0.9340400099754333
tuning-job-1-1a89f0f3cd5343889f-168-a9ddf684	Completed	3 hours ago	0.9340400099754333

An endpoint that can be reached from Pega is automatically created. A binary classification, as in this example, predicts if an event will happen or not, based on a cut-off value. By default, the response content for a binary model is set to this 'predicted_label'.

However, it is best practice to include a value for the probability that the event will happen in the response content as it contains the most information and allows the cutoff value to be adjusted in Pega. Also, it allows for monitoring of the probability with respect to observed outcomes over time.



Deploy model

REQUIRED SETTINGS

Endpoint name: SageMakerChurn

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Instance type: ml.m5.xlarge

Instance count: 1

Data capture

SageMaker Studio will save prediction requests and responses from the endpoint to an Amazon S3 location specified below

☐ Save prediction requests

☐ Save prediction responses

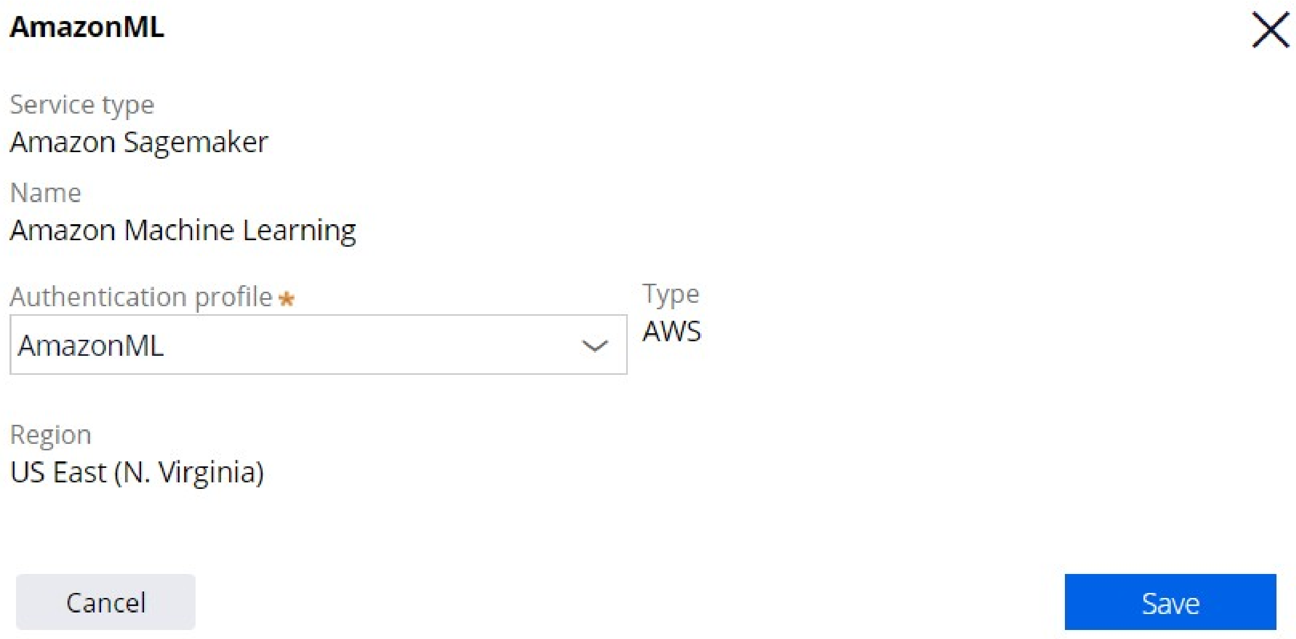
Inference Response Content

Select the response content the endpoint should return per input data point. The inference response will be in the order in which the keys are selected.

probability ✕

ADVANCED SETTINGS - *Optional*

In Prediction Studio, you can define a machine learning service to connect to your cloud service instance. To move messages securely to and from Pega, the system architect has set up an authentication profile.



AmazonML

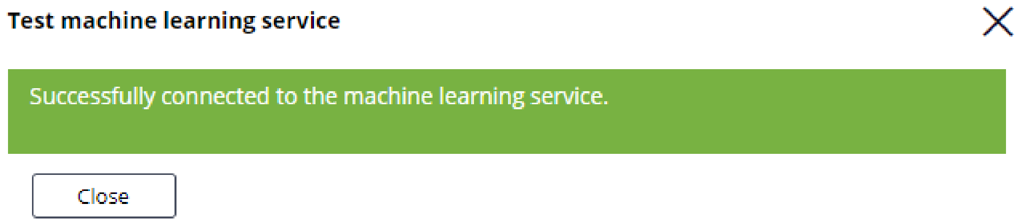
Service type
Amazon Sagemaker

Name
Amazon Machine Learning

Authentication profile *
AmazonML

Type
AWS

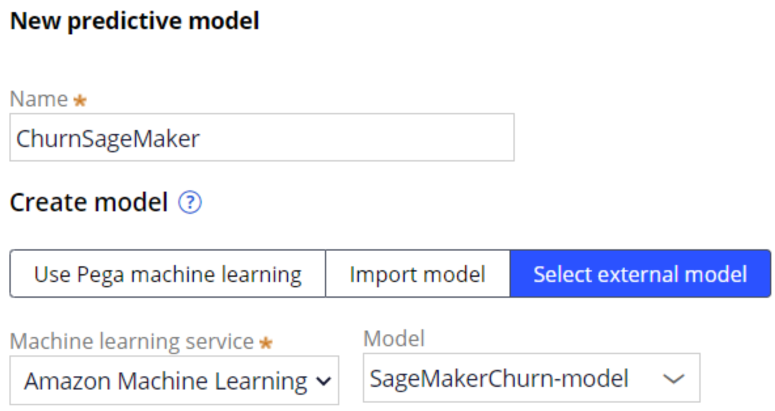
Region
US East (N. Virginia)



Test machine learning service

Successfully connected to the machine learning service.

Once the connection to the machine learning is established, start by creating a new predictive model to leverage the service. Select the machine learning service and the model that you want to reference.



New predictive model

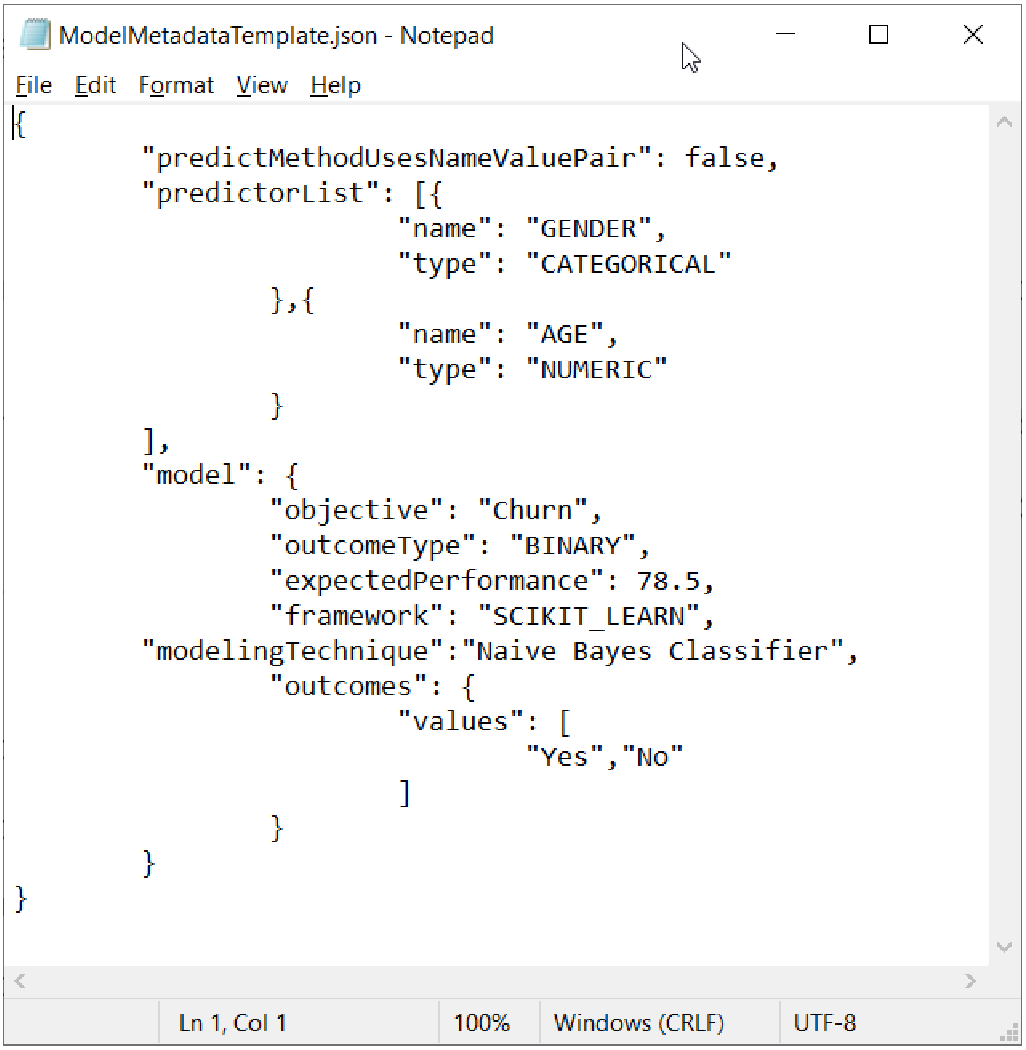
Name *
ChurnSageMaker

Create model

Machine learning service *
Amazon Machine Learning

Model
SageMakerChurn-model

Next, upload the required model metadata file. A template for this JSON file, containing example values, is available for download.


ModelMetadataTemplate.json - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8

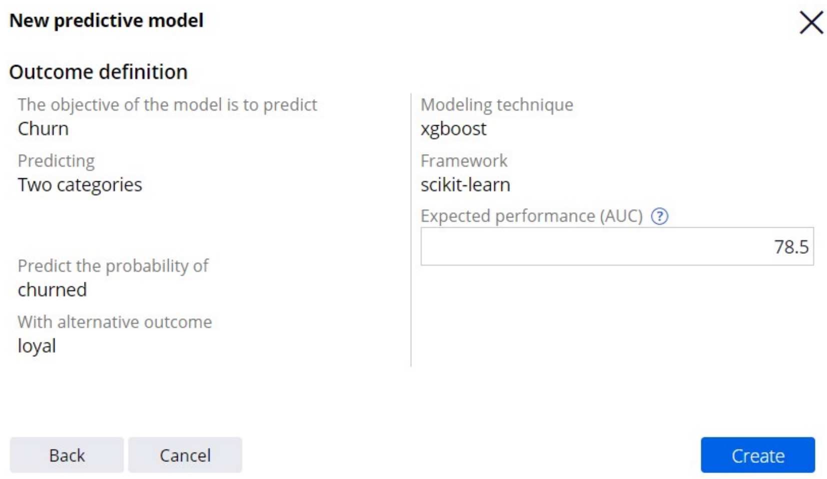
```
{
        "predictMethodUsesNameValuePair": false,
        "predictorList": [{
                        "name": "GENDER",
                        "type": "CATEGORICAL"
                },{
                        "name": "AGE",
                        "type": "NUMERIC"
                }
        ],
        "model": {
                "objective": "Churn",
                "outcomeType": "BINARY",
                "expectedPerformance": 78.5,
                "framework": "SCIKIT_LEARN",
        "modelingTechnique":"Naive Bayes Classifier",
                "outcomes": {
                        "values": [
                                "Yes","No"
                        ]
                }
        }
}
```

The JSON file must contain the list of predictors in the data set and their property type. It must also contain the objective of the model and the outcome type. Available outcome types are binary, categorical, and continuous. Optionally, include the expected performance. The metric for binary models is AUC, F-score for categorical models and RMSE for continuous models.

For SageMaker, the file must include the framework property. This property determines the input format and output format of the model.

Finally, the metadata file must include the modeling technique and the outcome values. For binary outcome models, enter the values for the outcome for which you want to predict the probability, and the alternative outcome. For categorical outcome models, enter all values that represent the possible outcomes. For continuous outcome models, enter minimum and maximum outcome values. Best practice is to generate the file as part of the model-building process to avoid human errors.

Next, set the correct context of the model if required. The default context is the customer class. You can review the model metadata, such as the objective of the model and the type of problem to solve, before proceeding.



New predictive model

Outcome definition

The objective of the model is to predict
Churn

Predicting
Two categories

Predict the probability of
churned

With alternative outcome
loyal

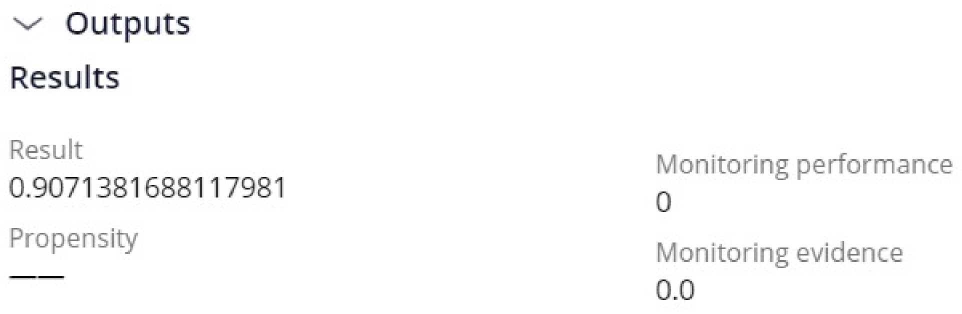
Modeling technique
xgboost

Framework
scikit-learn

Expected performance (AUC)
78.5

All predictors must be mapped to the corresponding fields in the data model. After saving the model, you can run it through the new service connection.

Customer Troy has a high risk of churning; the model returned a high probability to churn for him.



Outputs

Results

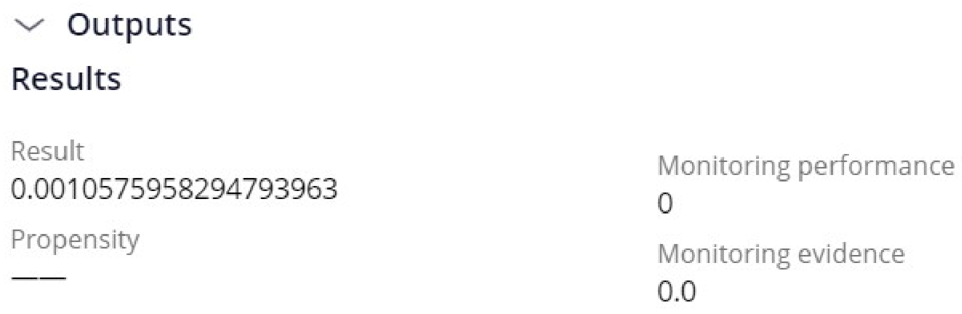
Result
0.9071381688117981

Propensity
——

Monitoring performance
0

Monitoring evidence
0.0

Customer Barbara will probably remain loyal; the model returned a low probability to churn for her.



Outputs

Results

Result
0.0010575958294793963

Propensity
——

Monitoring performance
0

Monitoring evidence
0.0

By default, the results of the model are shown in the Results field. Model results are unique for each framework type on which a model is built. Pega offers full support for the xgboost, tensorflow, kmeanclustering, knn, linearlearner and randomcutforest frameworks.

Once the predictive model rule is created, it can be used in next-best-action strategies in a similar way as native Pega machine learning models and third-party models imported using PMML or H2O.ai. But there is an important difference to keep in mind. Native and imported models, using the required input data, execute inside Pega. In the case of machine learning services, the input data required by the model is sent to the external platform, the model is executed externally, outside of Pega, and the result is sent back to Pega using a secured connection.

You’ve reached the end of this demo. What did it show you?

- The high level steps involved in creating a model using Amazon SageMaker Autopilot.
- How to connect to external machine learning services and run a model externally.

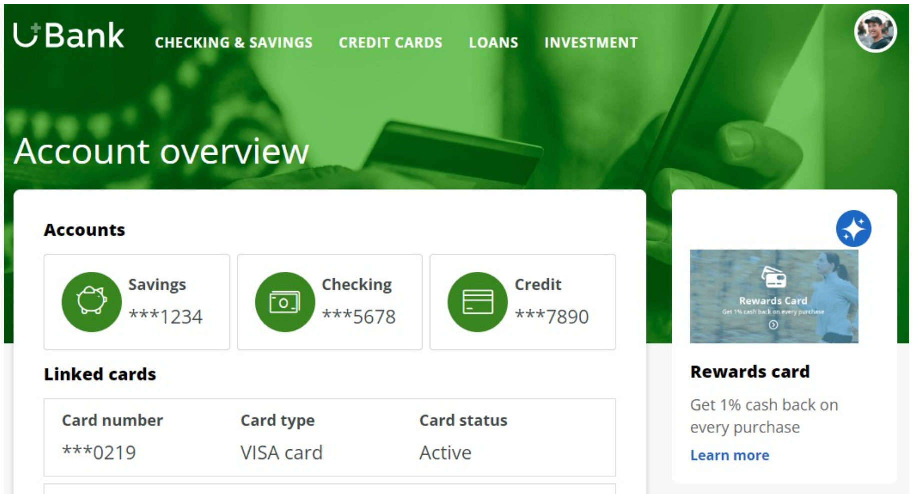
Challenging a predictive model

In the rapidly evolving landscape of data-driven decision-making, organizations are continually seeking ways to enhance their predictive models and ensure optimal outcomes. Learn how to shadow and challenge a Champion model, and ultimately promote the Challenger model to Active status in Pega Customer Decision Hub™.

Transcript

Hi, I'm Bella, a Lead Decisioning Architect working on a Customer Decision Hub project. In this demo, I will show you how to introduce a candidate model for a churn prediction.

U+ Bank uses Customer Decision Hub to optimize customer engagement on the bank's website.



Account overview

Accounts

Savings ***1234

Checking ***5678

Credit ***7890

Linked cards

Card number	Card type	Card status
***0219	VISA card	Active

Rewards card

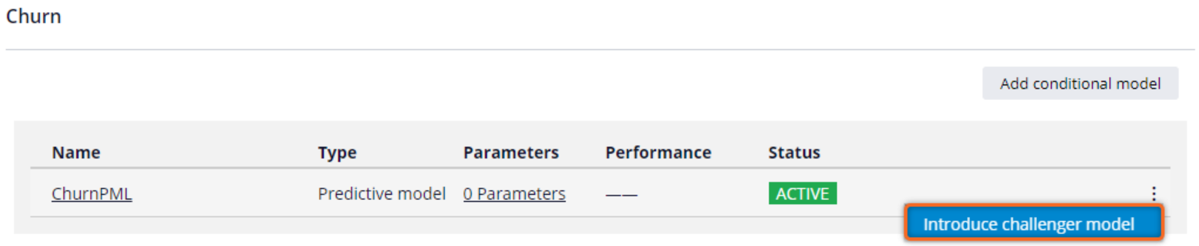
Get 1% cash back on every purchase

To reduce the number of customers leaving the bank, Customer Decision Hub uses the Predict Churn Propensity prediction. This prediction calculates the likelihood that a customer will switch to a competitor soon. A predictive churn model built on historical data drives the prediction.

The bank personalizes interactions with customers that have a high churn risk, like Troy, and prioritizes retention offers. In this case, Troy receives a 'free extra miles' offer in the hope of retaining him. In contrast, customers with a low churn risk, like Barbara, receive a credit card offer. The predictive power of the churn model declines over time as customer behavior changes, and the model needs to be updated regularly.

You can gently introduce an updated model in a series of deployment cycles. Begin by placing the candidate model in shadow mode, so it does not impact business outcomes. When you submit your changes for deployment, an automatic change request is created in the open revision. The Revision Manager deploys the revision, and the changes become effective in production. Next, you can utilize a Champion Challenger pattern to gradually increase the utilization of the Challenger model instead of the Champion model over time. Lastly, promote the Challenger model to Active status to replace the Champion model.

When you introduce a candidate model, you have the option to validate the candidate model and compare it with the Active model. For this, you need a validation dataset. This validation data set contains the predictor values and the outcome for a set of customers.



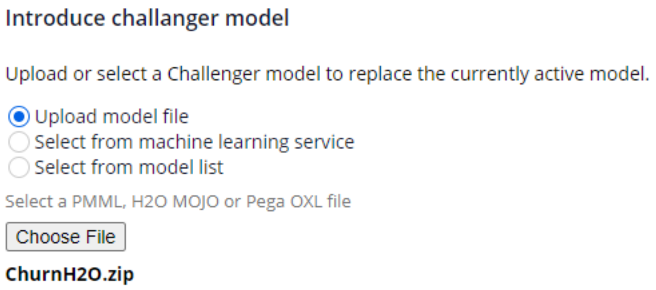
Churn

Add conditional model

Name	Type	Parameters	Performance	Status
ChurnPML	Predictive model	0 Parameters	——	ACTIVE

Introduce challenger model

When introducing a Challenger model, you have different options to select the model. You can use a file-based model, connect to a machine learning service like Google AI or Amazon SageMaker, or select a model that is available in your application.



Introduce challanger model

Upload or select a Challenger model to replace the currently active model.

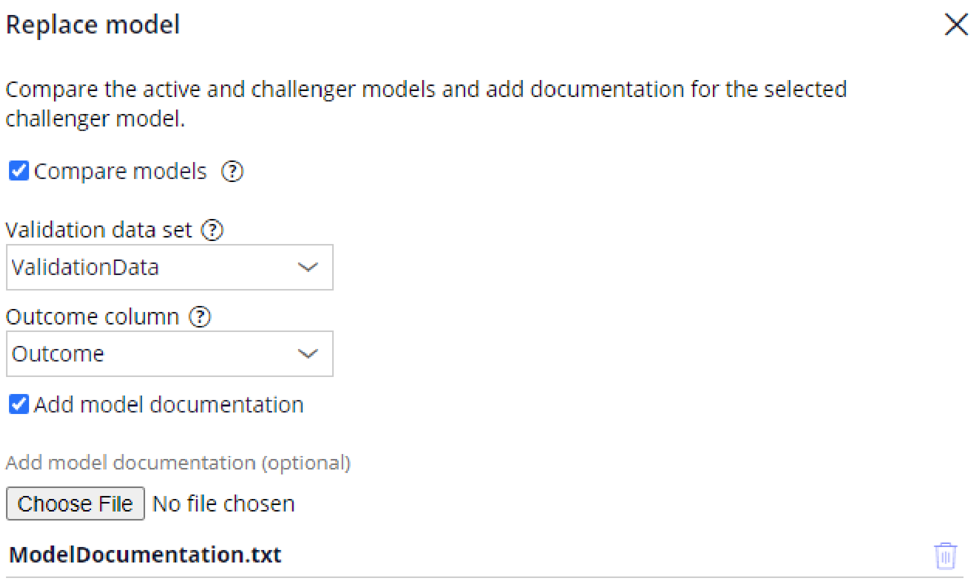
● Upload model file
○ Select from machine learning service
○ Select from model list

Select a PMML, H2O MOJO or Pega OXL file

Choose File

ChurnH2O.zip

To compare the Active and Challenger models, use your validation data set. Optionally, add supporting documents that provide detailed information about the model, its purpose, and its components. These documents help stakeholders understand the structure and functionality of the model, making it easier to maintain, update, and share with others.



Replace model

Compare the active and challenger models and add documentation for the selected challenger model.

☑ Compare models ⓘ

Validation data set ⓘ

ValidationData

Outcome column ⓘ

Outcome

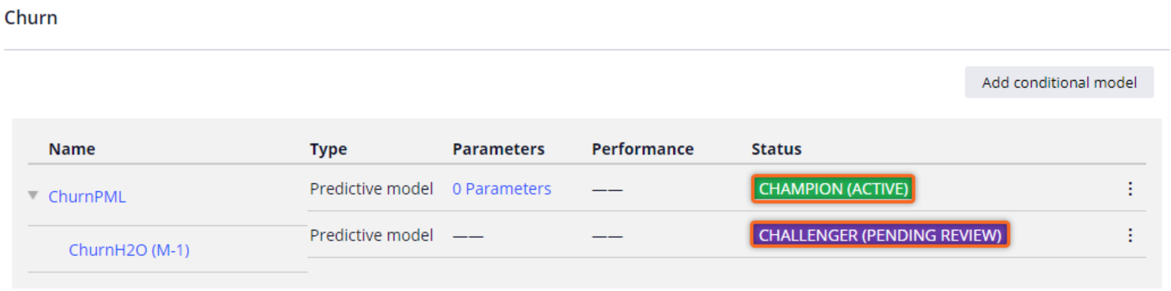
☑ Add model documentation

Add model documentation (optional)

Choose File No file chosen

ModelDocumentation.txt

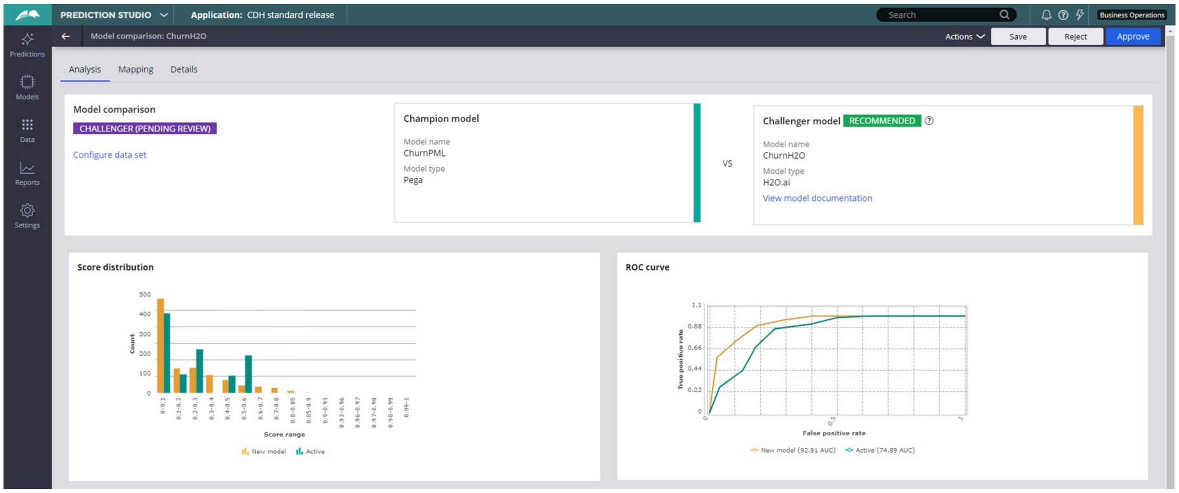
The system configures the challenger model in a pending review state.



Churn

Name	Type	Parameters	Performance	Status
ChurnPML	Predictive model	0 Parameters	——	CHAMPION (ACTIVE)
ChurnH2O (M-1)	Predictive model	——	——	CHALLENGER (PENDING REVIEW)

Review the predictive power of the Challenger model compared to the current Champion model, and then approve or reject the Challenger model.



Model comparison: ChurnH2O

Model comparison

CHALLENGER (PENDING REVIEW)

Configure data set

Champion model

Model name
ChurnPML

Model type
Pega

VS

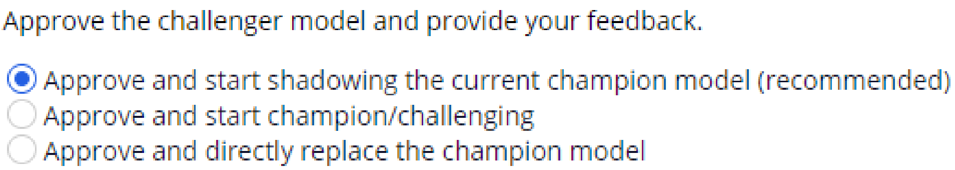
Challenger model RECOMMENDED

Model name
ChurnH2O

Model type
H2O.ai

View model documentation

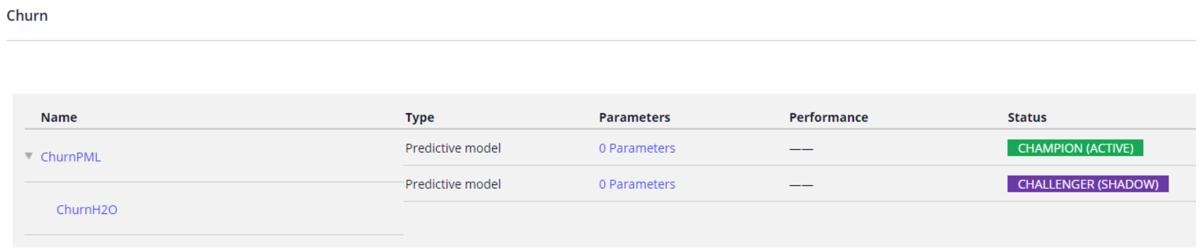
You have the option to shadow, challenge, or directly replace the Active model with the Challenger model.



Approve the challenger model and provide your feedback.

● Approve and start shadowing the current champion model (recommended)
○ Approve and start champion/challenging
○ Approve and directly replace the champion model

It is recommended to start a Challenger model in shadow mode. In shadow mode, the challenger model runs alongside the current model, receiving production data and generating outcomes without impacting business decisions.

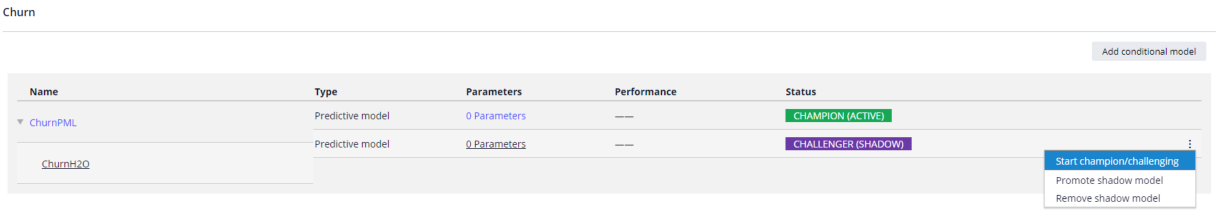


Churn

Name	Type	Parameters	Performance	Status
▾ ChurnPML	Predictive model	0 Parameters	——	CHAMPION (ACTIVE)
ChurnH2O	Predictive model	0 Parameters	——	CHALLENGER (SHADOW)

When you submit for deployment, a change request containing all changes made to the prediction is created in the open revision. When the Revision Manager deploys the revision, the changes take effect in production. Note that your validation data set can be included in the revision.

After running the Challenger model in shadow mode for some time, and the model performs without errors and the range of scores is as expected, you switch to a Champion Challenger pattern.

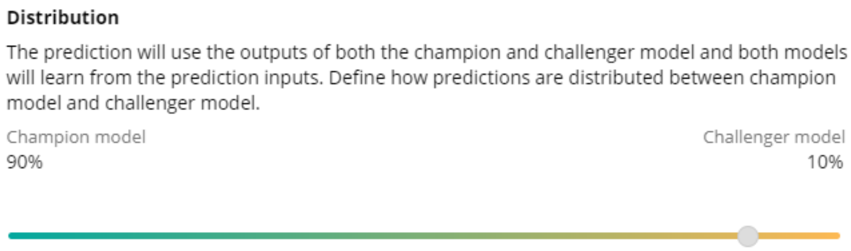


Churn

Add conditional model

Name	Type	Parameters	Performance	Status
▾ ChurnPML	Predictive model	0 Parameters	——	CHAMPION (ACTIVE)
ChurnH2O	Predictive model	0 Parameters	——	CHALLENGER (SHADOW)

When you challenge the Active model, in a small percentage of cases the Challenger model is used instead of the Active model.



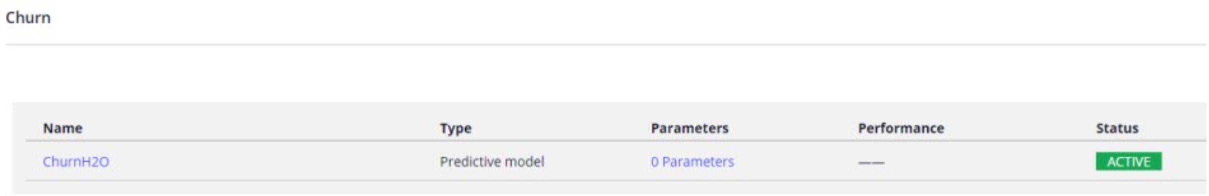
Distribution

The prediction will use the outputs of both the champion and challenger model and both models will learn from the prediction inputs. Define how predictions are distributed between champion model and challenger model.

Champion model 90%

Challenger model 10%

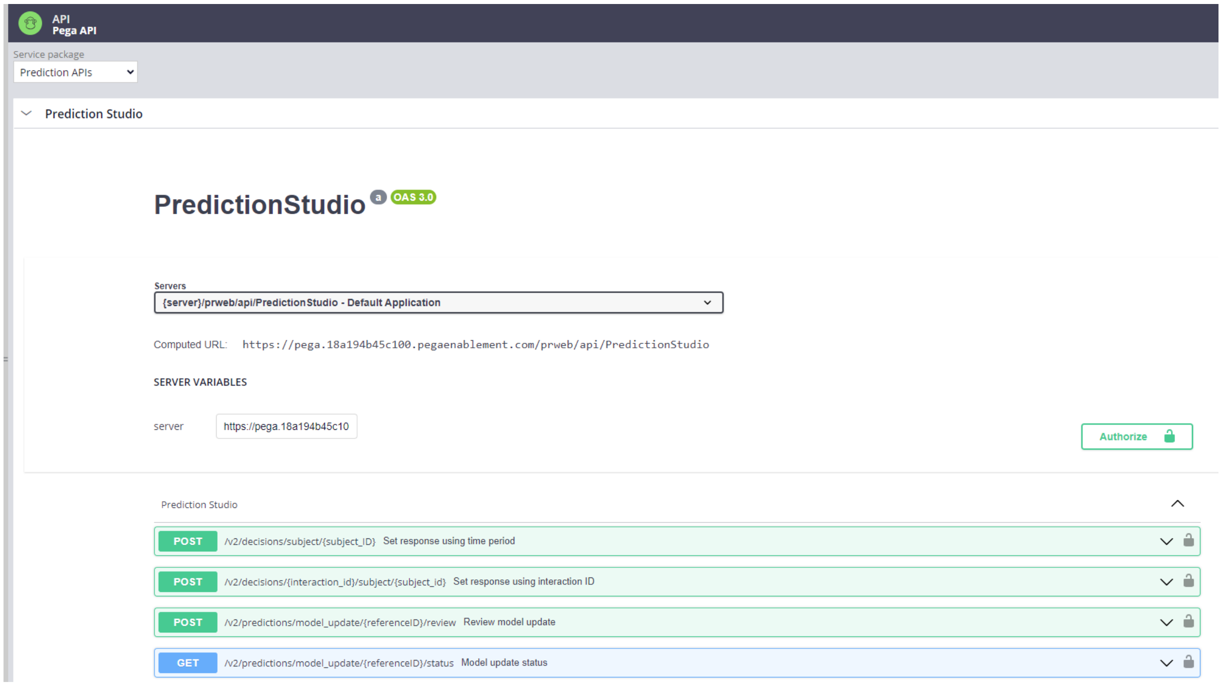
You can edit the distribution to gradually increase this percentage to give the Challenger model more exposure over time and then initiate the promotion of the Challenger model to Active status. By promoting the challenger model, you replace the Champion model with the Challenger model.



Churn

Name	Type	Parameters	Performance	Status
ChurnH2O	Predictive model	0 Parameters	——	ACTIVE

To introduce a Challenger model, you can also use the available Prediction APIs.



API
Pega API

Service package
Prediction APIs

Prediction Studio

PredictionStudio a OAS 3.0

Servers
{server}/prweb/api/PredictionStudio - Default Application

Computed URL: `https://pega.18a194b45c100.pegaenablement.com/prweb/api/PredictionStudio`

SERVER VARIABLES

server: https://pega.18a194b45c10

Prediction Studio

POST /v2/decisions/subject/{subject_ID} Set response using time period

POST /v2/decisions/{interaction_id}/subject/{subject_id} Set response using interaction ID

POST /v2/predictions/model_update/{referenceID}/review Review model update

GET /v2/predictions/model_update/{referenceID}/status Model update status

The API provides options to add a model to your application, review and approve or reject a model update, and retrieve the status of a model update.

This allows you to integrate the model approval process in Prediction Studio with any external model deployment process.

To summarize:

- You can shadow the Active model with a candidate model in production.
- When you submit your changes for deployment, a change request is automatically created in the open revision.

- When the Challenger model functions as expected in production, you can gradually increase the use of the Challenger model.
- You can promote the Challenger model to Active status and replace the Champion model.

Decision Strategies Overview

Description

Strategy canvas provides a guided and intuitive UI to bootstrap your application development with proven best practices that help you generate the strategies. These strategies can be customized using designated extension points or by building decision strategies from scratch, depending on the business requirement.

Learning objectives

- Describe how decision strategies are used
- Explain the decision strategy canvas and its building blocks
- Create decision strategies from scratch
- Explain what’s going on inside each component when a decision strategy is executed

Creating a decision strategy

Introduction

Decision Strategies drive Next-Best-Action. They comprise a unit of reasoning represented by decision components. How these components combine determines which action will be selected for a customer: the Next-Best-Action. Learn the type of decision components and how they are used to create decision strategies. Gain hands-on experience designing and executing your own Next-Best-Action decision strategy.

Transcript

This demo will show you how to create a new decision strategy.

It will also describe three important decision components and the types of properties available for use in expressions during strategy building.

In this demo you will build a Next-Best-Label strategy. The Next-Best-Label strategy is a sample strategy, used to illustrate the mechanics of a decision strategy.

Start by creating a new strategy from scratch.

Decision strategies output actions, utilizing the so-called Strategy-Results class.

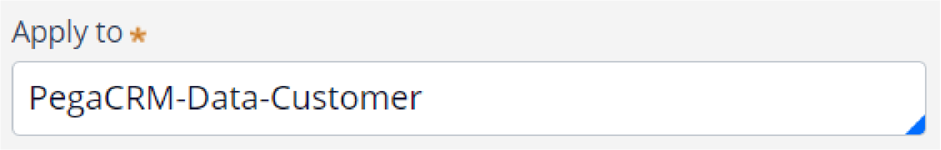
The Strategy-Results class limits the output of the strategy to the actions contained in the Business issue and Group.

The strategy you build will select a Label action from a set of predefined actions. The Label action selected will be the one with the lowest printing cost.

Notice that the complete definition of the Next-Best-Label strategy needs to include a reference to the PegaCRM-Data-Customer class.

This is the ‘Apply to’ class and it indicates the context of the strategy.

It ensures that from within the strategy, you have access to customer-related properties such as Age, Income, Address, Name, etc.



Apply to *

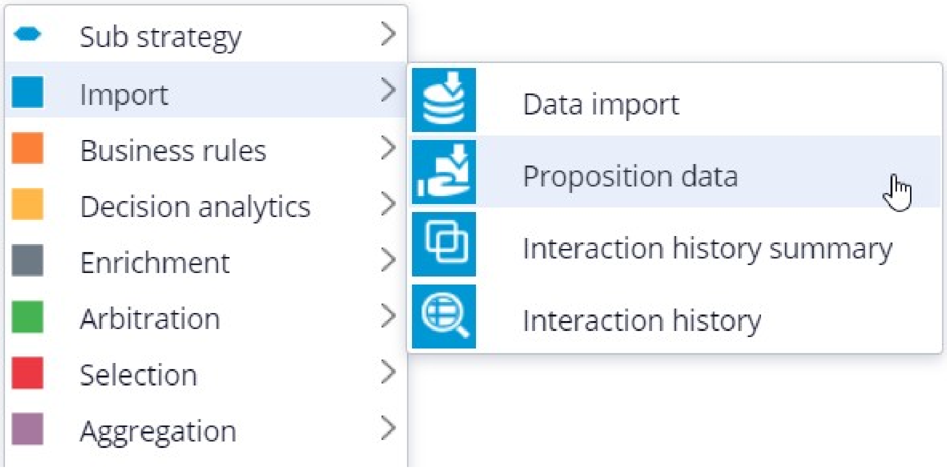
PegaCRM-Data-Customer

You can now start building the strategy. Right-click on the canvas to get the Context menu, which shows all component categories.

The first component to add is an Import component.

By expanding the Import category, you can see the Import component types available.

In this case you need a Proposition Data component to define the actions that will be considered by the strategy.



Now you need to configure the component. First, right-click to open the Proposition Data properties panel.

Notice that the Business issue and Group are grayed out.

This cannot be changed because the Enablement Business issue and Labels Group have already been selected for this decision strategy.

By default, the strategy will import all actions within that Group, unless you select a specific action.

For this component, you only want to import the Green Label, so let's select that.

Selecting the action from the drop-down menu automatically gives the component the appropriate name.

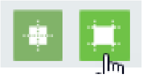
The description, which will appear under the component on the canvas, will also be generated automatically.

If you want to create your own description, you can do so by clicking the 'Use custom' radio button.

Now you want to import a second action into the strategy. You can use the Copy and Paste buttons to quickly add more Proposition Data components to the canvas.

You can use Alignment Snapping and Grid Snapping for easy placement of the components.

By turning these off, you can place a component anywhere on the canvas, but it makes it more difficult to align the shapes.



Now you need to add the next component in the strategy, which is an Enrichment component called Set Property.

You can add this component to the canvas by selecting it from the component menu.

Next, connect it to the Proposition Data components.

Ultimately, the result of this strategy should be the Label action with the lowest printing cost.

This printing cost is the sum of a base printing cost, which is specific to each label, and a variable cost, which depends on the number of letters.

The Set Property component is where you will calculate the printing cost for each of the actions.

The information in the 'Source components' tab is populated automatically by the Proposition Data components connected to this component.

Notice that the Black Label action is in the first row.

On the Target tab you can add properties for which values need to be calculated.

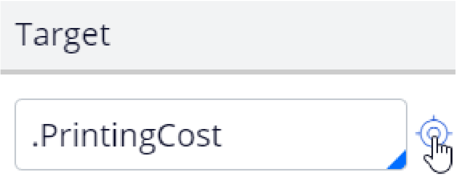
Click 'Add Item' to create the equation that will calculate the printing cost for each of the components.

Begin by setting the Target property to 'dot' PrintingCost.

In Pega, all inputs begin with a dot. This is called the dot-operator and it means that you are going to use a strategy property.

The PrintingCost property is a new strategy property that does not yet exist.

To create the new PrintingCost strategy property, click on the icon next to the Target field.



Target

.PrintingCost

By default, the property type is Text. In Pega, there are various types supported. In this case, the PrintingCost is a numeric value, so change its type to Decimal.

Next, you need to make PrintingCost equal to the calculation you create. To create the calculation, click on the icon next to the Source field.

Using the Expression builder, you can create all sorts of complex calculations, but in this use case, the computation is very basic.

PrintingCost should equal BaseCost + 5 * LetterCount.

To access the BaseCost you type a dot. Notice that when you type the dot, a list of available and relevant strategy properties appears.

This not only makes it easy to quickly find the property names you're looking for; it also avoids spelling mistakes.

In a decision strategy, you have two categories of properties available to use in Expressions.

The first category contains the strategy properties, which can be one of two types.

An Action property is defined in the Action form. Examples are the BaseCost and LetterCount properties you are using here.

These properties have a value defined in the Action form and are available in the decision strategy via the Proposition Data component.

The property values can be overridden in the decision strategy but will often be used as read only.

The second type of strategy property is a calculation like the one you just created, PrintingCost. Such calculations are often created and set in the decision strategy.

These types of properties are either used as transient properties, for temporary calculations, or for additional information you want the strategy to output.

The second category contains properties from the strategy context, also called customer properties.

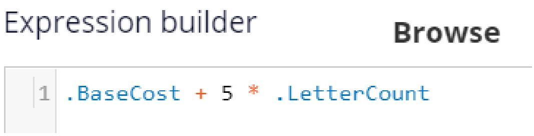
Suppose you want to use a customer property in your Expression, such as Age or Income.

In that case, you would have to type the prefix 'Customer dot', instead of just dot.

This is the list of available properties from the strategy context, also known as Customer properties.

For now, you calculate the printing cost for each action that does not use customer properties.

Finalize the Expression.

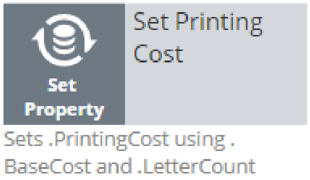


```
.BaseCost + 5 * .LetterCount
```

Even though you used the dot-operator to build your Expression, it's best practice to validate it, so click Test.

If the Expression isn't valid, you will receive an error message on screen.

On the canvas, you can see the automatically generated description for the component: Sets PrintingCost using BaseCost and LetterCount.



Set Printing Cost

Sets .PrintingCost using .BaseCost and .LetterCount

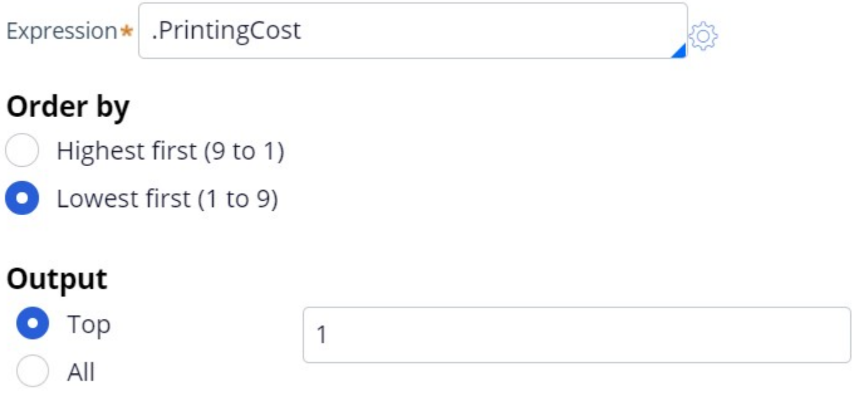
Now you want to ensure that the actions will be prioritized based on the lowest printing cost. So, you need to add the Prioritize component from the Arbitration category.

The prioritization can either be based on an existing property, or it can be based on an equation. Let's select an existing property using the dot construct.

Here you can select the order in which the top actions are presented. Since you are interested in the lowest printing costs, configure it accordingly.

You can also select the number of actions that will be returned by the strategy.

If you want to output only one label, select Top 1 here.



Expression* .PrintingCost

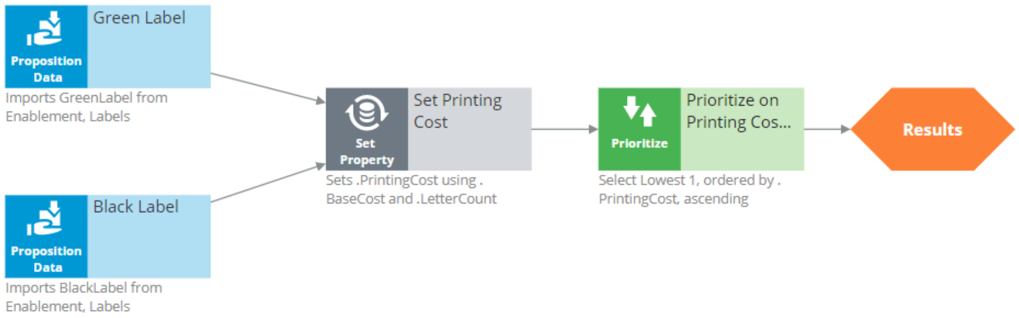
Order by

○ Highest first (9 to 1)
● Lowest first (1 to 9)

Output

● Top 1
○ All

Now you can connect the components and save the strategy.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Set Property
Set Printing Cost
Sets .PrintingCost using .BaseCost and .LetterCount
Prioritize
Prioritize on Printing Cos...
Select Lowest 1, ordered by .PrintingCost, ascending
Results

To test the strategy, first check it out. Then, expand the right-hand side test panel and click 'Save & Run' to examine the results.

You can view results for any of the components by selecting that component.

If more than one action is present, each one is presented as a Page.

For the Set Property component, the Results contain a page for the Black Label and one for the Green Label.

For the Black Label the PrintingCost is 70.

For the Green Label the PrintingCost is 60.

On the canvas, you can show values for strategy properties such as Printing Cost.

For this exercise, you execute this strategy against a Data Transform called UseCase1.

If you open UseCase1, you can see the customer data the strategy uses when you run it.

To test the strategy on a different use case, you can create a Data Transform with different properties.

You can also select a Data Set that points to an actual live database table.

This demo has concluded. What did it show you?

- How to create a decision strategy from scratch.
- How to configure Proposition Data, Set Property and Prioritize decision components.
- How to build expressions in strategies.
- The two categories of properties available for expressions.
- How to test a decision strategy using a use case stored in a data transform.

Decision strategy execution

Introduction

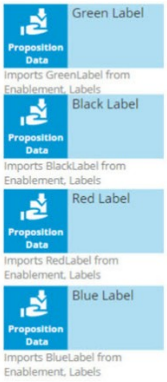
Using Pega Decision Management, you do not need to be an expert in programming, math or data science to design and execute sophisticated decision strategies that engage your customers throughout the customer journey. With its highly intuitive graphical canvas, Pega Decision Management enables you to easily embed Pega or third-party predictive models into your decision strategies. The result is customer-centric interactions that improve the customer experience while increasing customer value, retention and response rates.

Transcript

This demo explains what's going on inside each component when a Decision Strategy is executed.

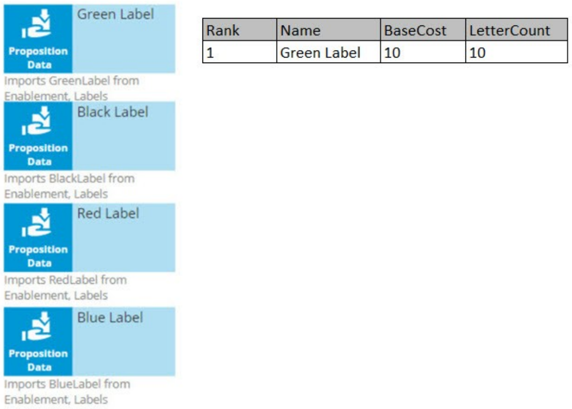
For example, what happens 'under the covers' when a Filter component is executed, and how does it interact with the components around it?

In the interest of keeping it simple, this example is limited to four actions. In reality, decision strategies will involve many more actions than that.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Proposition Data
Red Label
Imports RedLabel from Enablement, Labels
Proposition Data
Blue Label
Imports BlueLabel from Enablement, Labels

Here are our 4 actions: 'Green Label', 'Black Label', 'Red Label' and 'Blue Label'; they are represented by a Data Import or, more specifically, a Proposition Data component.

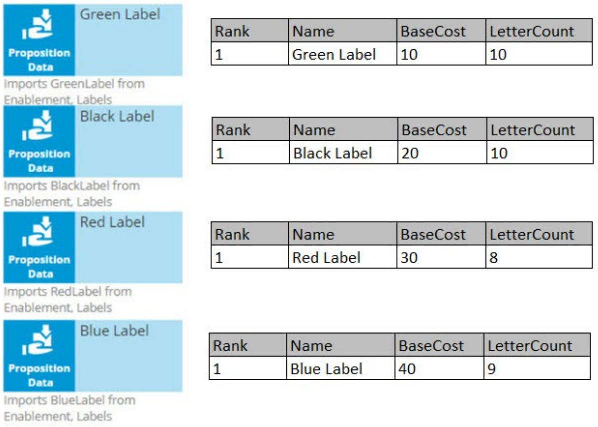
In this example, the Proposition Data components import three data properties for each action: Name, BaseCost and LetterCount.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Proposition Data
Red Label
Imports RedLabel from Enablement, Labels
Proposition Data
Blue Label
Imports BlueLabel from Enablement, Labels

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10

The first action's Name is Green Label, its BaseCost is 10, and its LetterCount is 10.

Likewise, the other actions have a Name, BaseCost and LetterCount.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Proposition Data
Red Label
Imports RedLabel from Enablement, Labels
Proposition Data
Blue Label
Imports BlueLabel from Enablement, Labels

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10

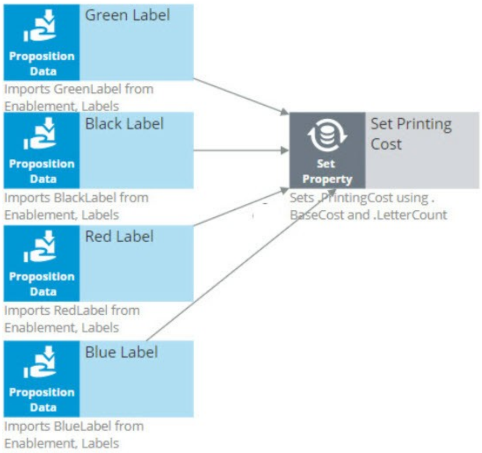
Rank	Name	BaseCost	LetterCount
1	Black Label	20	10

Rank	Name	BaseCost	LetterCount
1	Red Label	30	8

Rank	Name	BaseCost	LetterCount
1	Blue Label	40	9

One property is automatically populated for you; this is the Rank. We will come back to this later, but notice that, as separate components, each action has a Rank of 1.

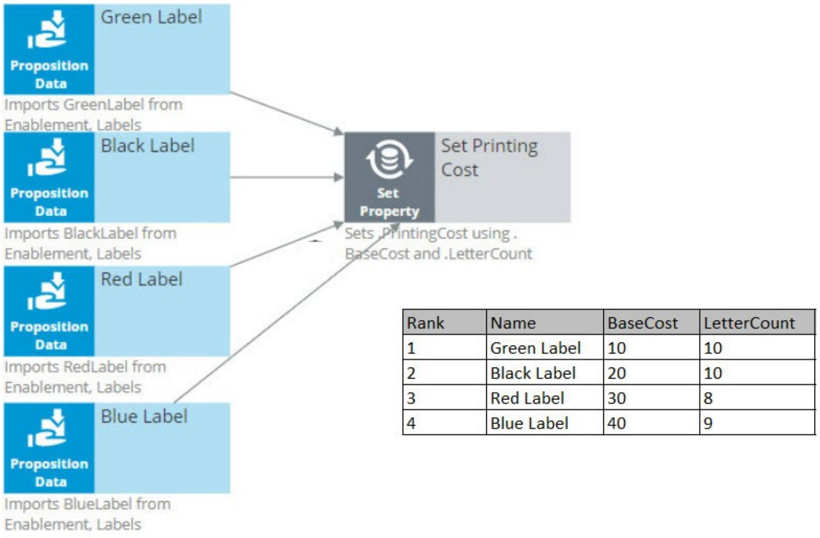
On the strategy canvas, components are connected by drawing arrows from component to component. So, what do these arrows mean exactly?


Green Label
Proposition Data
Imports GreenLabel from Enablement, Labels
Black Label
Proposition Data
Imports BlackLabel from Enablement, Labels
Red Label
Proposition Data
Imports RedLabel from Enablement, Labels
Blue Label
Proposition Data
Imports BlueLabel from Enablement, Labels
Set Property
Set Printing Cost
Sets .PrintingCost using .BaseCost and .LetterCount

Well, when you draw an arrow, what happens is that, at runtime, all information in the component you're drawing the arrow from is available as a data source to the component you're drawing the arrow to.

So now, the Name, BaseCost and LetterCount for all of the actions are available in a single Set Property component.

The only data element that changes is the row number, or as we call it in the strategies, the Rank. In each decision component, the Rank value is automatically computed.


Green Label
Proposition Data
Imports GreenLabel from Enablement, Labels
Black Label
Proposition Data
Imports BlackLabel from Enablement, Labels
Red Label
Proposition Data
Imports RedLabel from Enablement, Labels
Blue Label
Proposition Data
Imports BlueLabel from Enablement, Labels
Set Property
Set Printing Cost
Sets .PrintingCost using .BaseCost and .LetterCount

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10
2	Black Label	20	10
3	Red Label	30	8
4	Blue Label	40	9

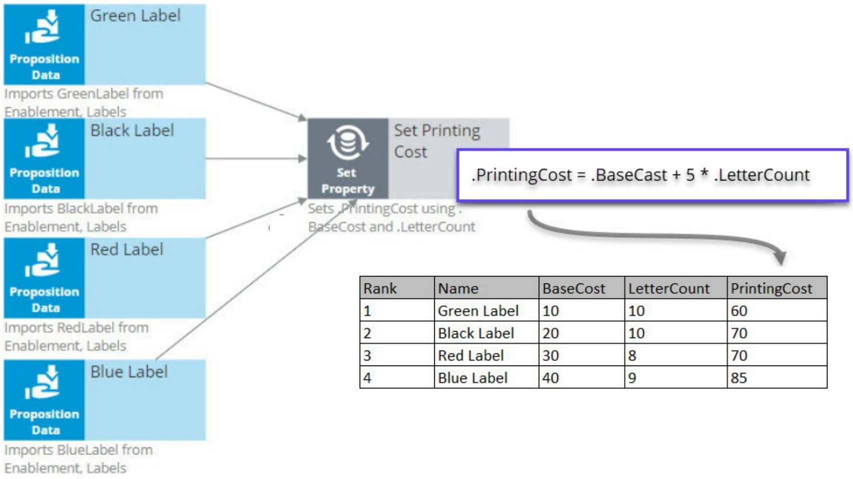
In the Set Property component, the Rank is determined by the order in which the actions are received by the component.

As a result, in this instance, the Green Label action has a Rank of 1, Black has a Rank of 2, Red has a Rank of 3, and Blue has a Rank of 4.

Ultimately, you want to select the best Label action. That is the Label with the lowest printing cost.

The printing cost of a Label is the sum of the BaseCost and a variable cost based on the LetterCount.

You configure the Set Property component to compute the printing cost of each Label action.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Proposition Data
Red Label
Imports RedLabel from Enablement, Labels
Proposition Data
Blue Label
Imports BlueLabel from Enablement, Labels
Set Property
Set Printing Cost
Sets .PrintingCost using .BaseCost and .LetterCount

.PrintingCost = .BaseCast + 5 * .LetterCount

Rank	Name	BaseCost	LetterCount	PrintingCost
1	Green Label	10	10	60
2	Black Label	20	10	70
3	Red Label	30	8	70
4	Blue Label	40	9	85

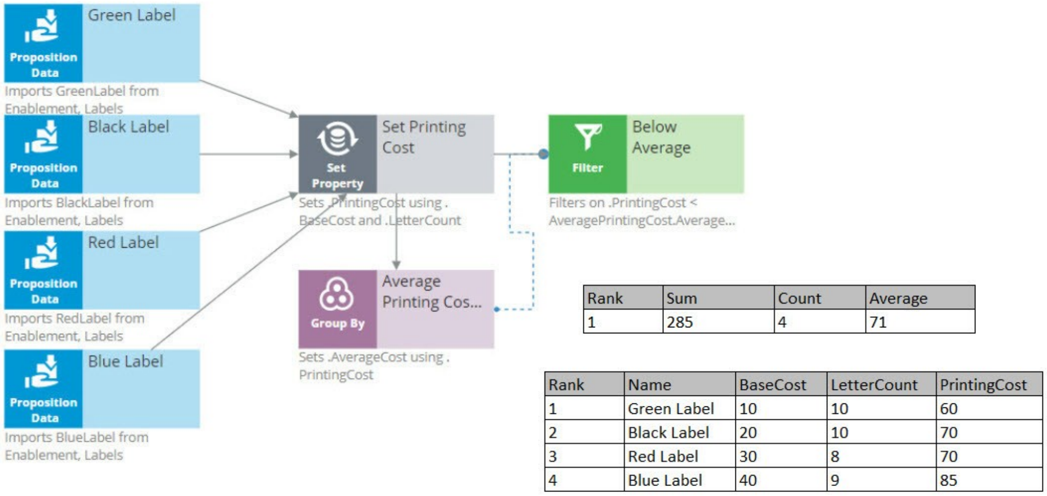
Because we are combining the data in our four Proposition Data components into one Set Property component, we only need to add one PrintingCost property to the new component, and it automatically computes the printing cost for all four actions.

For the Green Label action, PrintingCost equals a BaseCost of 10 plus 5 times the LetterCount of 10 which equals 60.

Similarly, the PrintingCost for the Black and Red Label actions is 70, and for the Blue Label action is 85.

Now, let’s say the business rule is to select only Label actions with a printing cost lower than the average printing cost of all labels. For this requirement we use a ‘Group by’/Filter component combination.

A ‘Group by’ component offers essential aggregation capabilities, like Sum and Count, that are used in many decision strategies. We will use it to calculate the average printing cost.


Green Label
Proposition Data
Imports GreenLabel from Enablement, Labels
Black Label
Proposition Data
Imports BlackLabel from Enablement, Labels
Red Label
Proposition Data
Imports RedLabel from Enablement, Labels
Blue Label
Proposition Data
Imports BlueLabel from Enablement, Labels
Set Printing Cost
Set Property
Sets .PrintingCost using .BaseCost and .LetterCount
Average Printing Cos...
Group By
Sets .AverageCost using .PrintingCost
Below Average
Filter
Filters on .PrintingCost < AveragePrintingCost.Average...

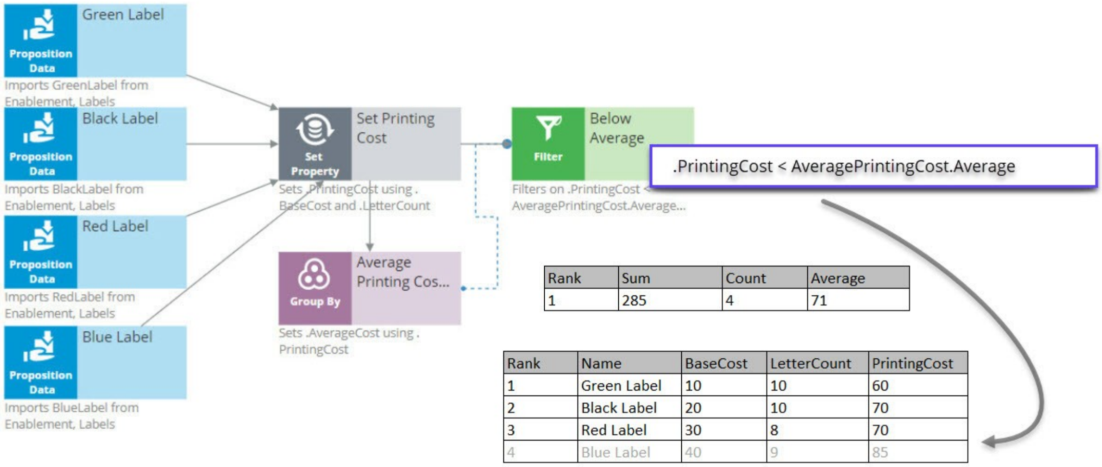
Rank	Sum	Count	Average
1	285	4	71

Rank	Name	BaseCost	LetterCount	PrintingCost
1	Green Label	10	10	60
2	Black Label	20	10	70
3	Red Label	30	8	70
4	Blue Label	40	9	85

Again, we have our set of actions, each with their own specific PrintingCost value. The 'Group by' component combines all actions into one row. How does that work?

Well, it sums the PrintingCost values for all the actions, it counts the actions, and it calculates the average printing cost by dividing the summed printing cost by the count.

In this example, the sum of the PrintingCost values is 285, and the count of the actions is 4, so the average printing cost is 71.


Green Label
Proposition Data
Imports GreenLabel from Enablement, Labels
Black Label
Proposition Data
Imports BlackLabel from Enablement, Labels
Red Label
Proposition Data
Imports RedLabel from Enablement, Labels
Blue Label
Proposition Data
Imports BlueLabel from Enablement, Labels
Set Printing Cost
Set Property
Sets .PrintingCost using .BaseCost and .LetterCount
Average Printing Cos...
Group By
Sets .AverageCost using .PrintingCost
Below Average
Filter
Filters on .PrintingCost < AveragePrintingCost.Average...

.PrintingCost < AveragePrintingCost.Average

Rank	Sum	Count	Average
1	285	4	71

Rank	Name	BaseCost	LetterCount	PrintingCost
1	Green Label	10	10	60
2	Black Label	20	10	70
3	Red Label	30	8	70
4	Blue Label	40	9	85

Now that you have calculated the average printing price using a 'Group by' component, configure the Filter component to filter out actions that have a printing cost equal to or higher than this average.

So far in this strategy, we've seen only the solid line arrows, which copy information from one component to another. But now we also see a dotted line arrow.

This tells us that a component refers to information in another component.

Here, the Filter component is referencing the average printing cost that exists inside the Aggregation component. This is an important capability to understand.

The Filter component filters out actions when the printing cost for that action is equal to or above the average printing cost and propagates the other actions.

First, via the solid arrow, the filter looks at the actions sourced from the Set Property component.

Then, it applies the filter condition, which references the average printing cost in the 'Group by' component via the dotted arrow.

The Filter Condition in the Filter component is the Expression: 'dot PrintingCost is smaller than AveragePrintingCost dot Average'.

By using this ComponentName dot Property construct, any decision component can be referenced by any other component by name.

Important to note that the Filter component lets actions through when the condition Expression evaluates to **true** and filters out actions when the condition Expression is not met.

When you refer to a component, you always refer to the first element in the component, the one with Rank 1.

In this case, you are referring to the one and only row in the 'Group by' component, which naturally has Rank 1.

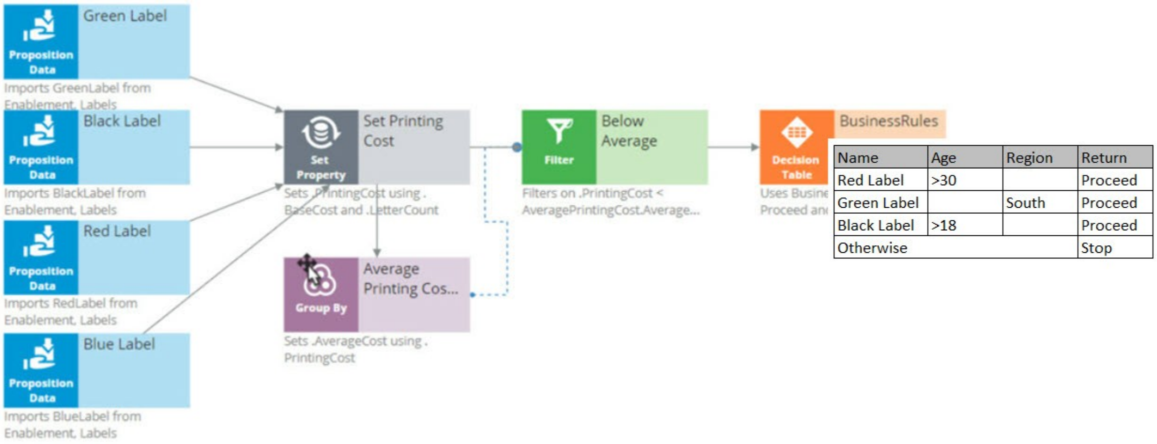
The Rank 1 average equals 71 in the 'Group by' component. This means that the filter will allow Label actions through that have a printing cost lower than 71.

By this standard, the printing cost of the Blue Label action is too high, so it is filtered out. The printing cost of the other Label actions are below 71, so they survive.

The result is that the table contains three surviving actions: Green Label with Rank 1, Black Label with Rank 2, and Red Label with Rank 3.

The next component is a Decision Table. A Decision Table in Pega is an artifact that can be used to implement business requirements in table format.

In a Decision Table, the business rules are represented by a set of conditions and a set of Return values.


Green Label
Proposition Data
Imports GreenLabel from Enablement. Labels
Black Label
Proposition Data
Imports BlackLabel from Enablement. Labels
Red Label
Proposition Data
Imports RedLabel from Enablement. Labels
Blue Label
Proposition Data
Imports BlueLabel from Enablement. Labels
Set Property
Set Printing Cost
Sets .PrintingCost using . BaseCost and .LetterCount
Group By
Average Printing Cos...
Sets .AverageCost using . PrintingCost
Filter
Below Average
Filters on .PrintingCost < AveragePrintingCost.Average...
Decision Table
BusinessRules
Uses Busine
Proceed an

Name	Age	Region	Return
Red Label	>30		Proceed
Green Label		South	Proceed
Black Label	>18		Proceed
Otherwise			Stop

The Decision Table receives information about the remaining actions via the solid arrow from the Filter component.

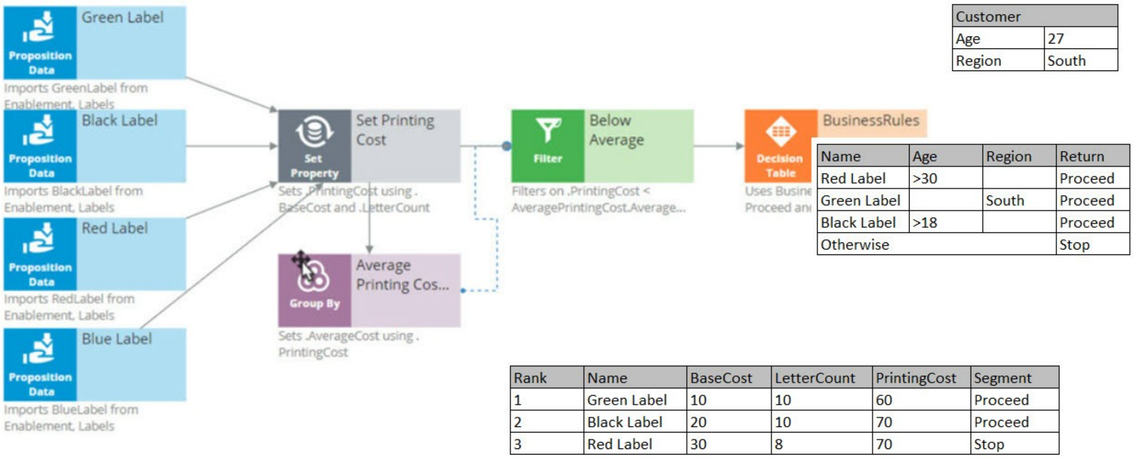
The business criteria say that the Red Label action can be offered if the customer's age is over 30 and they are from any region. If these criteria are met, the Return value is 'Proceed'.

The Decision Table also says that the Green Label action can be offered to anyone in the Southern region. So, if the Region value is South, the Return value for Green is 'Proceed'.

The Black Label action can be offered to anyone over the age of 18.

But in all other cases, or, Otherwise, no Label action meets the criteria, and the Return value is 'Stop'.

As an example, consider a customer with Age 27 and Region South.


Green Label
Proposition Data
Imports GreenLabel from Enablement. Labels
Black Label
Proposition Data
Imports BlackLabel from Enablement. Labels
Red Label
Proposition Data
Imports RedLabel from Enablement. Labels
Blue Label
Proposition Data
Imports BlueLabel from Enablement. Labels
Set Property
Set Printing Cost
Sets .PrintingCost using . BaseCost and .LetterCount
Group By
Average Printing Cos...
Sets .AverageCost using . PrintingCost
Filter
Below Average
Filters on .PrintingCost < AveragePrintingCost.Average...
Decision Table
BusinessRules
Uses Busine
Proceed an

Customer	
Age	27
Region	South

Name	Age	Region	Return
Red Label	>30		Proceed
Green Label		South	Proceed
Black Label	>18		Proceed
Otherwise			Stop

Rank	Name	BaseCost	LetterCount	PrintingCost	Segment
1	Green Label	10	10	60	Proceed
2	Black Label	20	10	70	Proceed
3	Red Label	30	8	70	Stop

Now, the Decision Table applies the business criteria for each action against the customer information and returns a value. The value returned by a Decision Table is also called a Segment.

The Decision Table checks the Green Label action with Rank 1 first, and in this case, it can proceed because the customer's Region is South.

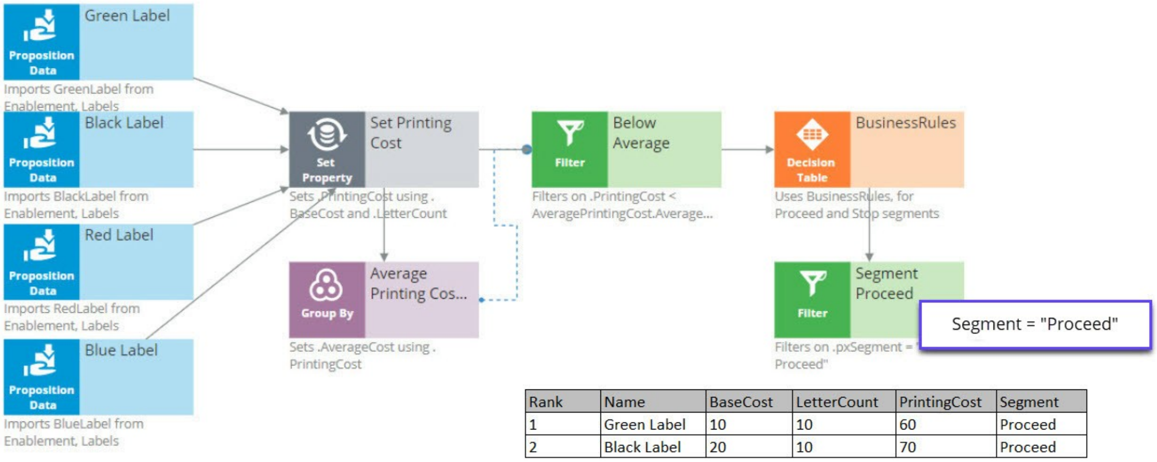
Next, it looks at the Black action and sees that the criteria for Black is that the customer's age is greater than 18. This customer is 27.

Black doesn't care about the Region, so the Segment value for the Black action is ‘Proceed’.

Finally, it looks at the Red action, and the Age criteria don't match up, so the Segment value for Red is ‘Stop’.

The result of the component is that you get a new segmentation column that flags which of the actions comply with the business rules.

You're now going to filter out the actions that do not match the business rules. This happens in the ‘Segment Proceed’ Filter component.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Proposition Data
Red Label
Imports RedLabel from Enablement, Labels
Proposition Data
Blue Label
Imports BlueLabel from Enablement, Labels
Set Property
Set Printing Cost
Sets .PrintingCost using .BaseCost and .LetterCount
Group By
Average Printing Cos...
Sets .AverageCost using .PrintingCost
Filter
Below Average
Filters on .PrintingCost < AveragePrintingCost.Average...
Decision Table
BusinessRules
Uses BusinessRules, for Proceed and Stop segments
Filter
Segment Proceed
Filters on .pxSegment = "Proceed"
Segment = "Proceed"

Rank	Name	BaseCost	LetterCount	PrintingCost	Segment
1	Green Label	10	10	60	Proceed
2	Black Label	20	10	70	Proceed

Again, via the solid arrow, the strategy copies the data over from the Decision Table component into the Filter component.

Now each action has a Rank, Name, BaseCost, LetterCount, PrintingCost and Segment. The filter condition is applied to this data.

The filter condition says: allow this action through if the Segment value equals ‘Proceed’.

What this Filter component now does is go through the list of actions to find the actions with value ‘Proceed’ in their Segment property.

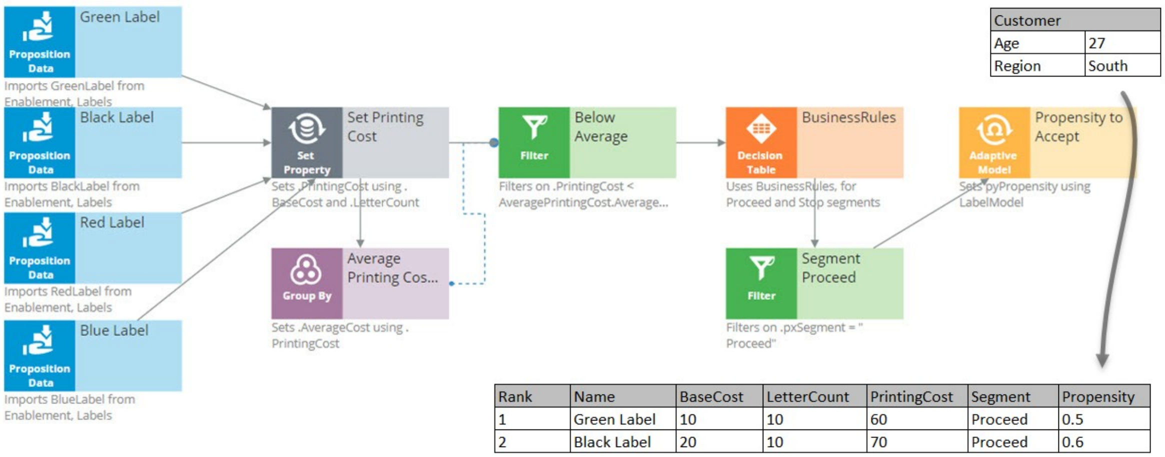
First is the Green Label. Green is allowed through, which means its properties will be available in the new component.

Then the Black Label. It is also allowed through because it also has ‘Proceed’ in its Segment property.

But the Red Label action is not allowed through, because Red has ‘Stop’ in its Segment property. Therefore, Red is not part of the output.

The strategy so far has selected two of our original actions, Green and Black.

Now, in the Adaptive Model component, you will use predictive analytics to determine the propensity of each of the remaining actions.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Proposition Data
Red Label
Imports RedLabel from Enablement, Labels
Proposition Data
Blue Label
Imports BlueLabel from Enablement, Labels
Set Property
Set Printing Cost
Sets .PrintingCost using .BaseCost and .LetterCount
Group By
Average Printing Cos...
Sets .AverageCost using .PrintingCost
Filter
Below Average
Filters on .PrintingCost < AveragePrintingCost.Average...
Decision Table
BusinessRules
Uses BusinessRules, for Proceed and Stop segments
Filter
Segment Proceed
Filters on .pxSegment = " Proceed"
Adaptive Model
Propensity to Accept
Sets pyPropensity using LabelModel

Customer	
Age	27
Region	South

Rank	Name	BaseCost	LetterCount	PrintingCost	Segment	Propensity
1	Green Label	10	10	60	Proceed	0.5
2	Black Label	20	10	70	Proceed	0.6

Propensity is the probability that a customer will accept an action, or, their likelihood of interest in it.

In order to calculate the propensity, we use an Adaptive Model component. The referenced model is configured to monitor customer characteristics such as Age and Region.

In this case our test customer has an Age of 27 and is from the South Region.

Again, just to keep it simple, we are using a model that makes predictions based on only this information. In reality, models will take into account many more properties.

The Adaptive Model determines the propensity.

First, we supply the action and the customer profile to the Adaptive Model, and the model says: ‘Oh, it’s the Green Label action; we have some evidence that young people like the Green Label action, but people from the South don’t like it.’

Combining both factors, we get an overall propensity of 0.5 for the Green Label action.

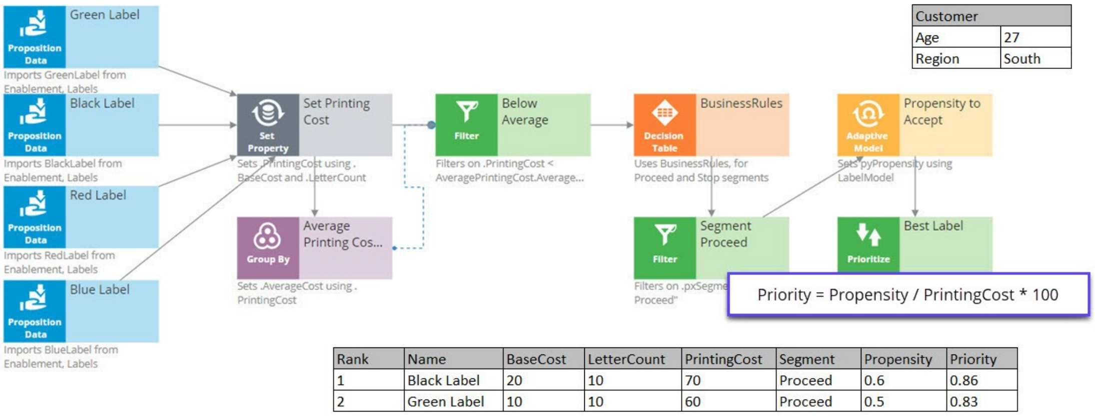
For the Black Label action, the likelihood turns out to be 0.6.

After consulting the Adaptive Model, the Propensity to Accept component sets the Propensity property value for each action.

Remember, the propensity is always a number between zero and 1.

It shows something along the lines of, half of the customers that are like this customer accepted the Green Label action in the past, and 3 out of 5 customers like this customer accepted the Black action last month.

The next component in our chain, called Best Label, is the Prioritize component. This component determines the priority of each action and ranks them. Let’s see how this works.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Proposition Data
Red Label
Imports RedLabel from Enablement, Labels
Proposition Data
Blue Label
Imports BlueLabel from Enablement, Labels
Set Property
Set Printing Cost
Sets .PrintingCost using . BaseCost and .LetterCount
Group By
Average Printing Cos...
Sets .AverageCost using . PrintingCost
Filter
Below Average
Filters on .PrintingCost < AveragePrintingCost.Average...
Decision Table
BusinessRules
Uses BusinessRules, for Proceed and Stop segments
Filter
Segment Proceed
Filters on .pxSegme
Proceed"
Adaptive Model
Propensity to Accept
Sets pyPropensity using LabelModel
Prioritize
Best Label

Customer	
Age	27
Region	South

Priority = Propensity / PrintingCost * 100

Rank	Name	BaseCost	LetterCount	PrintingCost	Segment	Propensity	Priority
1	Black Label	20	10	70	Proceed	0.6	0.86
2	Green Label	10	10	60	Proceed	0.5	0.83

A key element of this component is the priority Expression, which calculates a priority value for each action. According to this Expression, the higher the value, the higher the priority and rank.

In this case, the priority calculation weighs likelihood of acceptance in its equation: ‘Propensity divided by PrintingCost times 100’.

When performing this calculation on the Black Label action, we can see that it has a PrintingCost of 70 and a Propensity of 0.6, therefore its Priority is 0.86.

The Green Label action has a lower PrintingCost and a lower Propensity, resulting in a Priority of 0.83.

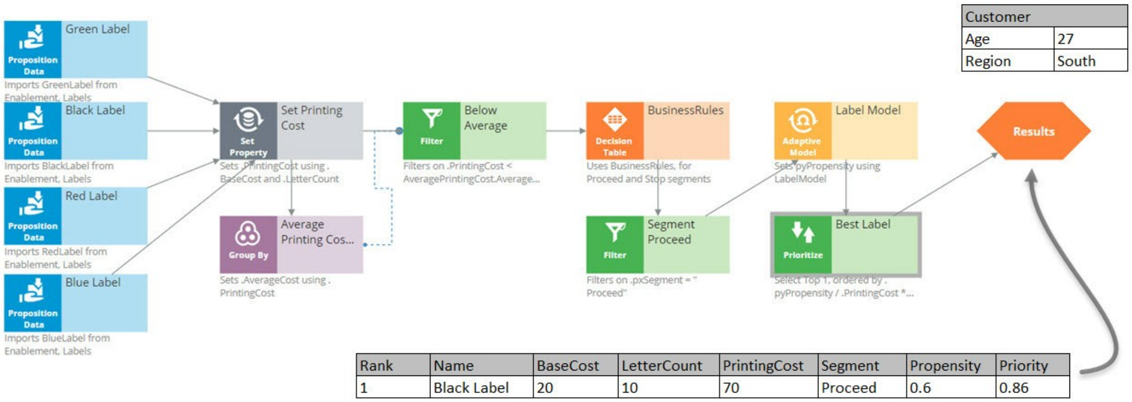
Because 0.86 is higher than 0.83, the Black Label action is now ranked number one.

So, even though the printing cost of the Black Label action is higher than that of the Green Label action, the Black Label action still comes out on top.

In this case, the Priority component reversed the Ranks of the two actions. Black is now the primary action and Green is the secondary action.

The same Prioritization component is also configured to output only the top action.

Therefore, it filters out the Green action altogether, and at the end of our strategy chain, the Black Label is left as our best action.


Proposition Data
Green Label
Imports GreenLabel from Enablement, Labels
Proposition Data
Black Label
Imports BlackLabel from Enablement, Labels
Proposition Data
Red Label
Imports RedLabel from Enablement, Labels
Proposition Data
Blue Label
Imports BlueLabel from Enablement, Labels
Set Property
Set Printing Cost
Sets .PrintingCost using .BaseCost and .LetterCount
Group By
Average Printing Cos...
Sets .AverageCost using .PrintingCost
Filter
Below Average
Filters on .PrintingCost < AveragePrintingCost.Average...
Decision Table
BusinessRules
Uses BusinessRules, for Proceed and Stop segments
Filter
Segment Proceed
Filters on .pxSegment = " Proceed"
Adaptive Model
Label Model
Sets pyPropensity using LabelModel
Prioritize
Best Label
Select Top 1, ordered by .pyPropensity / .PrintingCost *...
Results

Customer	
Age	27
Region	South

Rank	Name	BaseCost	LetterCount	PrintingCost	Segment	Propensity	Priority
1	Black Label	20	10	70	Proceed	0.6	0.86

Defining prediction patterns

Description

Learn how to improve the predictive power of your adaptive models by configuring additional potential predictors in Pega Customer Decision Hub™. For example, you can make input fields that are not directly available in the customer data model accessible to the models by configuring these fields as parameterized predictors.

Gain experience using a prediction in a decision strategy and learn how to arbitrate between different groups of actions to display more relevant offers to customers.

Learning objectives

- Describe how to use predictions in a decision strategy.
- Configure parameterized predictors.
- Arbitrate between business issues by using applicability rules in Next-Best-Action Designer.

Creating parameterized predictors

Pega Customer Decision Hub™ uses Adaptive Models that use a wide range of predictors to optimize customer interactions. You can make input fields that are not directly available in the customer Data Model accessible to the models by configuring these fields as parameterized predictors. Learn how to create parameterized predictors for Expressions, offline model scores, and online model scores.

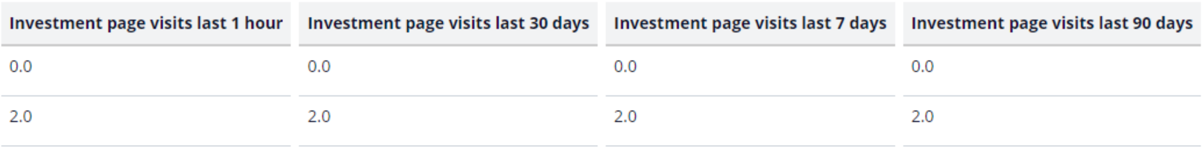
Transcript

This demo shows you how to create parameterized predictors to make input fields that are not directly available in the customer Data Model accessible to Adaptive Models. We will cover three examples: Expressions, offline model scores, and online model scores.

U+ Bank is cross-selling its credit cards on the web by using Pega Customer Decision Hub™. All customer data, including financial clickstream summary attributes, such as webpage visits, is available to the Adaptive Models that determine which offer to display for a particular customer.

The first parametrized predictor is the ratio of two clickstream summary attributes that denote the number of visits to the U+ Bank Investment webpage in the last 30 days and the last 90 days. An increasing ratio might indicate growing customer interest.

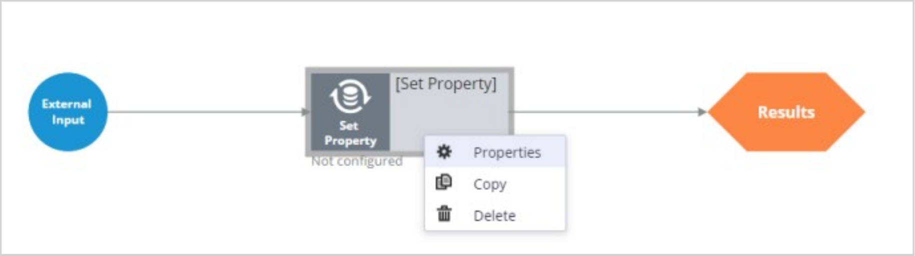
The *Financial Services Clickstream* Data Set contains the clickstream data, which is a record of the user's activity on the website. These entries reflect the *Investment* webpage visits of the customer.



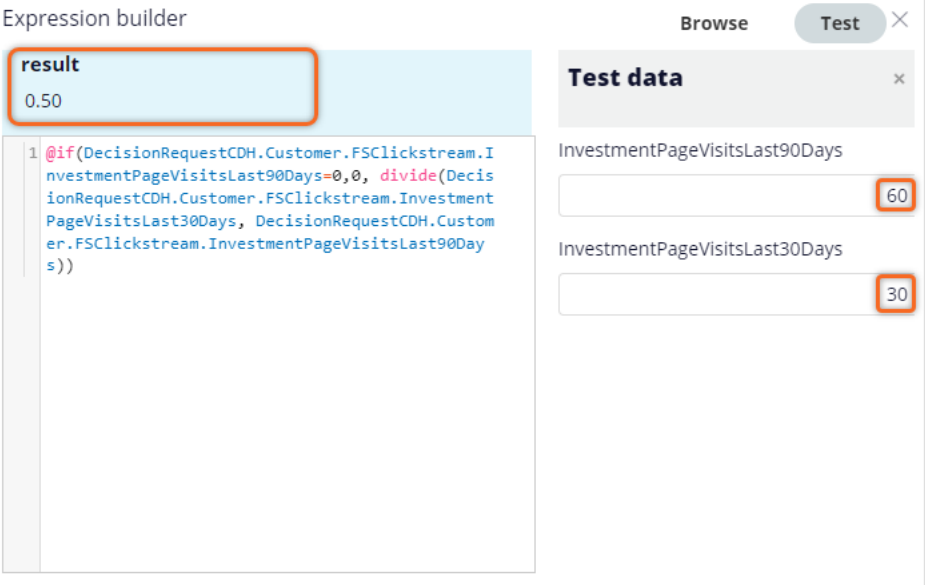
Investment page visits last 1 hour	Investment page visits last 30 days	Investment page visits last 7 days	Investment page visits last 90 days
0.0	0.0	0.0	0.0
2.0	2.0	2.0	2.0

To populate a parameterized predictor, you configure the *NBA Pre-Processing Extension Point* Decision strategy that is included in a change request. The pre-processing strategy runs before the Prediction strategy so that the output of the extension strategy is available as input to the Prediction.

The first example of a parameterized predictor is an Expression that references two clickstream fields. To add a field, you add a Set Property component to the strategy and create a new property in the *Strategy Results* class. Set the property type to **Decimal** to store the ratio of customer visits to a web page in the last 30 days and in the last 90 days.



This Expression returns a value of zero when the number of Investment page visits in the last 90 days is zero. Otherwise, it returns the ratio of the Investment page visits in the last 30 days to the last 90 days. A high value might indicate the increasing interest of the customer in the content of this page.



result
0.50

```
@if(DecisionRequestCDH.Customer.FSClickstream.I
nvestmentPageVisitsLast90Days=0,0, divide(Decis
ionRequestCDH.Customer.FSClickstream.Investment
PageVisitsLast30Days, DecisionRequestCDH.Custom
er.FSClickstream.InvestmentPageVisitsLast90Day
s))
```

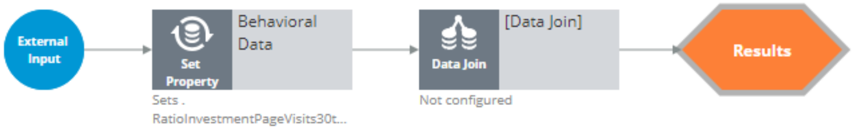
Test data

InvestmentPageVisitsLast90Days: 60

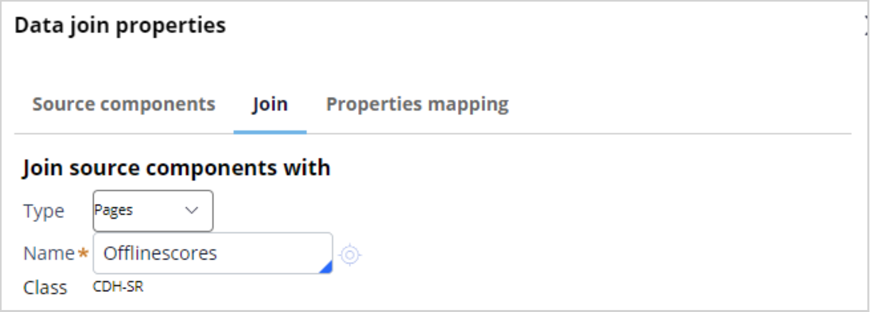
InvestmentPageVisitsLast30Days: 30

The second example of a parameterized predictor is offline model scores. A Data Set that is associated with the customer context makes the data available as a data source. Each customer can have multiple scores for different categories or different channels.

To add the model scores as potential predictors to the Adaptive Models, you add a Data Join component to the extension strategy.



The component joins the *Offlinescores* page that contains the model scores to the available data.



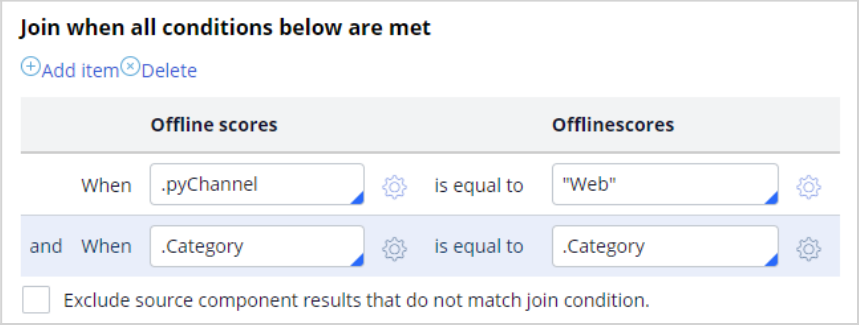
Data join properties

Source components | **Join** | Properties mapping

Join source components with

Type	Pages
Name*	Offlinescores
Class	CDH-SR

To output only relevant scores, you specify the channel and the product category to match the Prediction.



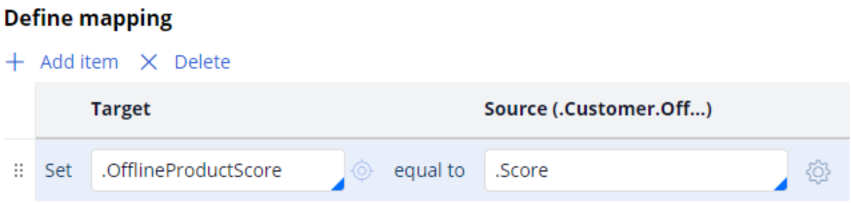
Join when all conditions below are met

⊕Add item ⊗Delete

		Offline scores		Offlinescores
	When	.pyChannel	is equal to	"Web"
and	When	.Category	is equal to	.Category

□ Exclude source component results that do not match join condition.

You then create a Decimal property and map it to accommodate the model scores.



Define mapping

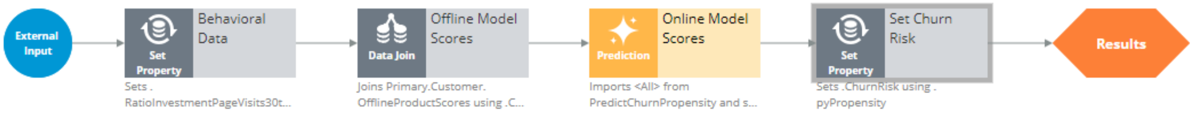
+ Add item ✕ Delete

	Target		Source (.Customer.Off...)
Set	.OfflineProductScore	equal to	.Score

The third example of a parameterized predictor is the score of a churn Prediction that runs in Customer Decision Hub. The Data Scientist team develops Predictive Models based on the latest insights and data that drive the churn Prediction.

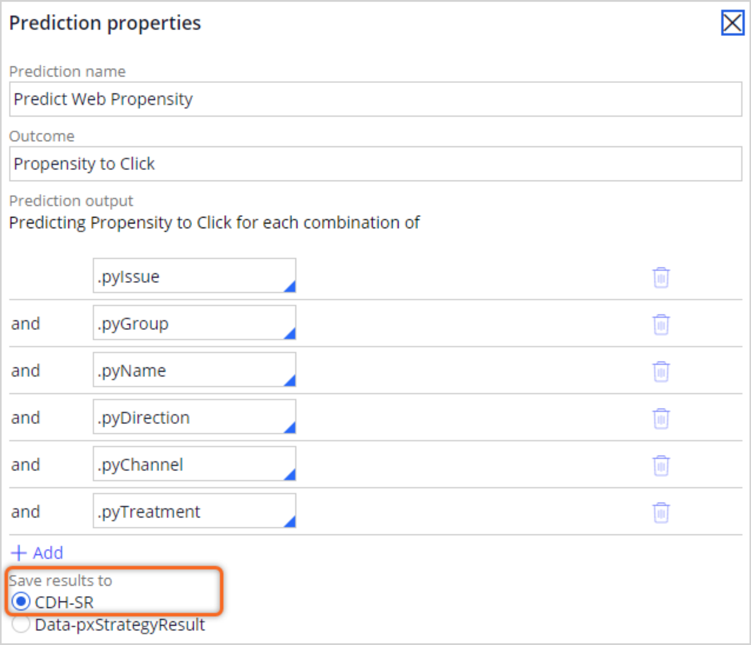
To make the churn score available to the Adaptive Models as a parameterized predictor, add a Prediction component to the extension strategy. Reference the churn Prediction in the *Customer* class.

Add a Set Property component to the extension strategy to create a new property and map it to the propensity calculated by the churn Prediction.


External Input
Set Property
Behavioral Data
Sets . RatioInvestmentPageVisits30t...
Data Join
Offline Model Scores
Joins Primary.Customer. OfflineProductScores using .C...
Prediction
Online Model Scores
Imports <All> from PredictChurnPropensity and s...
Set Property
Set Churn Risk
Sets .ChurnRisk using . pyPropensity
Results

Run the strategy by using a test customer profile to verify that the strategy outputs include the real-time churn risk score, the offline product scores, and the Expression.

Next, you use Prediction Studio to map three new parameterized predictors in the Prediction. The Predict Web Propensity Prediction calculates the likelihood that a customer clicks on a web banner. Like the new properties, the system saves the Prediction results to the *CDH-SR* class.



Prediction properties

Prediction name
Predict Web Propensity

Outcome
Propensity to Click

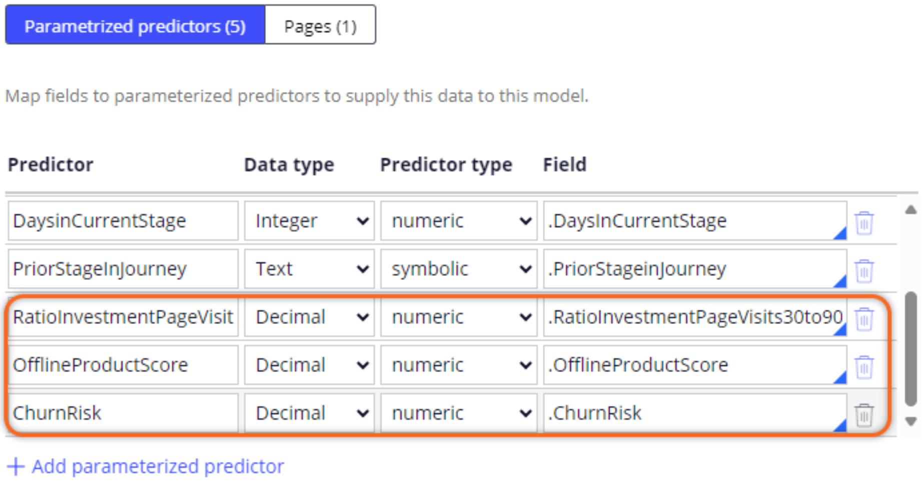
Prediction output
Predicting Propensity to Click for each combination of

	.pyIssue
and	.pyGroup
and	.pyName
and	.pyDirection
and	.pyChannel
and	.pyTreatment

+ Add

Save results to
● CDH-SR
○ Data-pxStrategyResult

You then create three new parameterized predictors and map them to the new properties that represent customer behavior, offline scores, and online scores.



Map fields to parameterized predictors to supply this data to this model.

Predictor	Data type	Predictor type	Field
DaysinCurrentStage	Integer	numeric	.DaysInCurrentStage
PriorStageInJourney	Text	symbolic	.PriorStageinJourney
RatioInvestmentPageVisit	Decimal	numeric	.RatioInvestmentPageVisits30to90
OfflineProductScore	Decimal	numeric	.OfflineProductScore
ChurnRisk	Decimal	numeric	.ChurnRisk

+ Add parameterized predictor

When you submit your work for deployment, the application saves all changes made in Predictions to a *Change prediction* change request.

After a review by the Team Lead, a Revision Manager deploys the changes as part of a revision.

You have reached the end of this video. You have learned:

- How to configure the pre-processing extension strategy for parameterized predictors.
- How to add properties as parameterized predictors to the Prediction.
- How to submit the changes for deployment.

Using predictions in engagement strategies

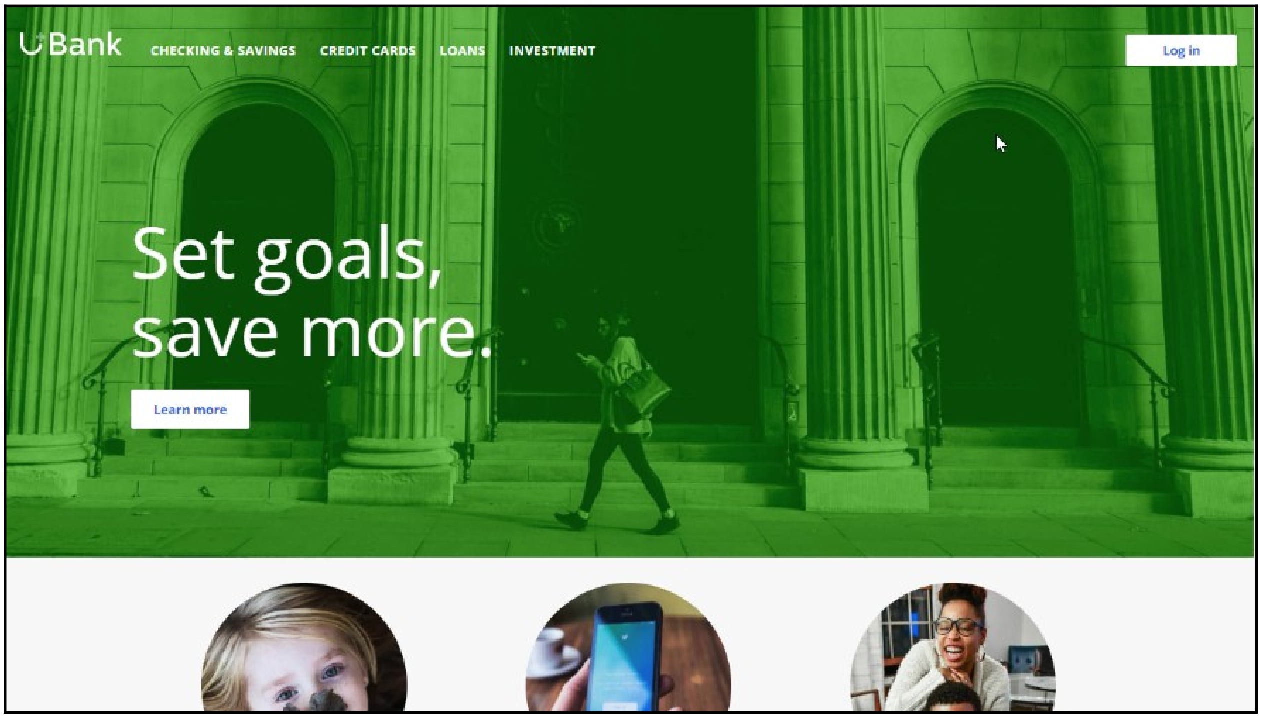
Introduction

A prediction is used to predict customer behavior such as offer acceptance and churn based on characteristics such as credit risk, income, and product subscriptions. Learn how to arbitrate between different groups of actions to display more relevant offers to customers. Gain experience using a prediction in a decision strategy and learn how applicability rules can be defined to reflect the bank's requirements in a decision strategy.

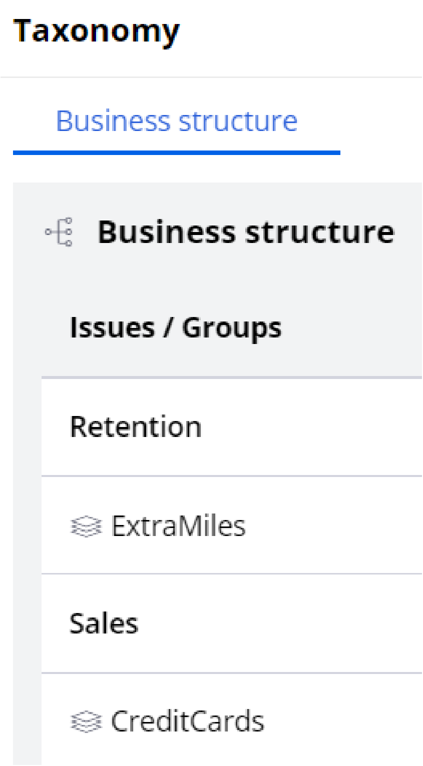
Transcript

This demo shows you how to use a prediction in an engagement strategy to determine customer applicability for a retention offer.

Currently, U+ Bank is cross-selling on the web by showing various credit cards to eligible customers who log in to its website. The bank now wants to show a retention offer, instead of a credit card offer, to customers who are likely to churn in the near future. The credit card offers are shown only to loyal customers.



To meet this business requirement, a decisioning administrator has already set up the taxonomy by defining a new business issue called Retention, and an offer group.



Taxonomy

Business structure

Business structure

Issues / Groups

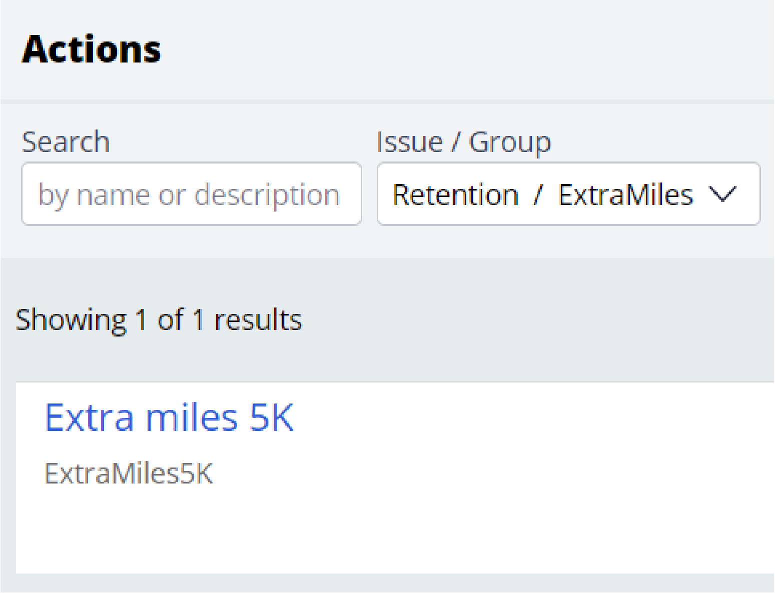
Retention

ExtraMiles

Sales

CreditCards

This ExtraMiles group contains a retention offer, Extra Miles 5K.



Actions

Search

by name or description

Issue / Group

Retention / ExtraMiles

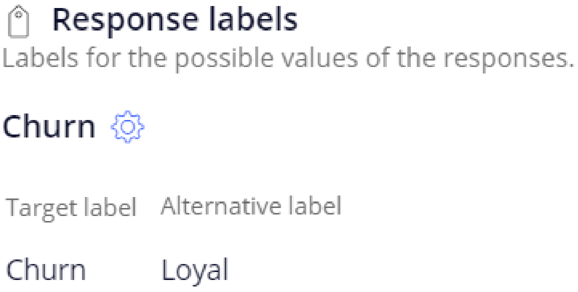
Showing 1 of 1 results

Extra miles 5K

ExtraMiles5K

The next step is to create an applicability condition that makes a customer qualify for a retention offer when there is a high likelihood that the customer might churn. A data scientist has created a prediction that identifies these high-risk customers. When you open the prediction in Prediction Studio, notice that the possible response labels are **Churn** and

Loyal to predict customer behavior. The result of the prediction is stored in the pxSegment property.



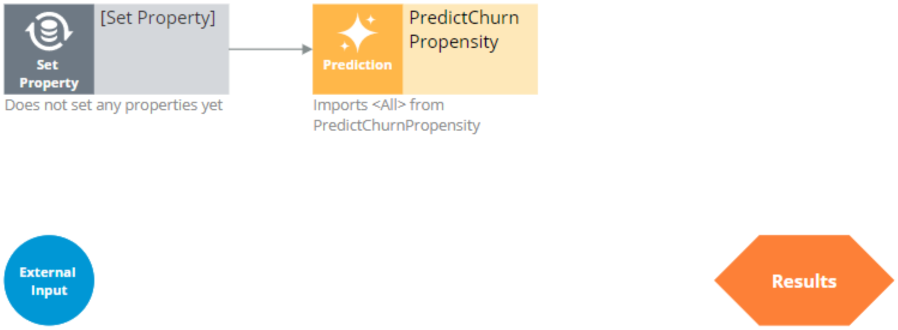
Response labels

Labels for the possible values of the responses.

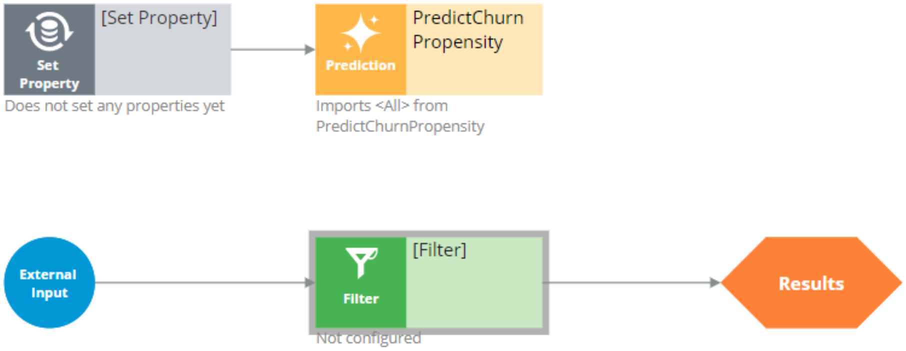
Churn

Target label	Alternative label
Churn	Loyal

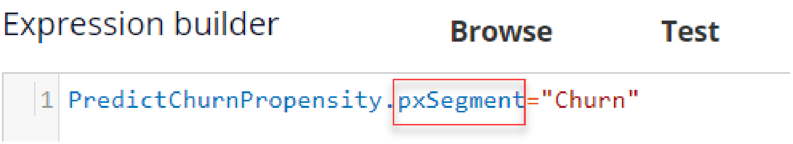
To define the applicability condition, you create a decision strategy to output a retention offer only if the response label of the prediction is **Churn**. Add a **Prediction** component to the canvas and configure it to reference the churn prediction. Add a **Set Property** component and connect it to the Prediction component. You can configure the Set Property component at a later point to accommodate parameterized fields.



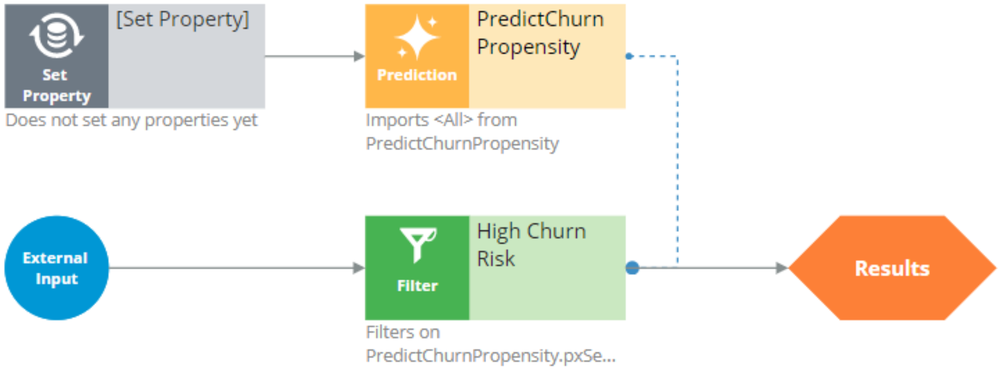
Next, add a filter component to filter out the loyal customers and pass retention offers to high churn risk customers only.



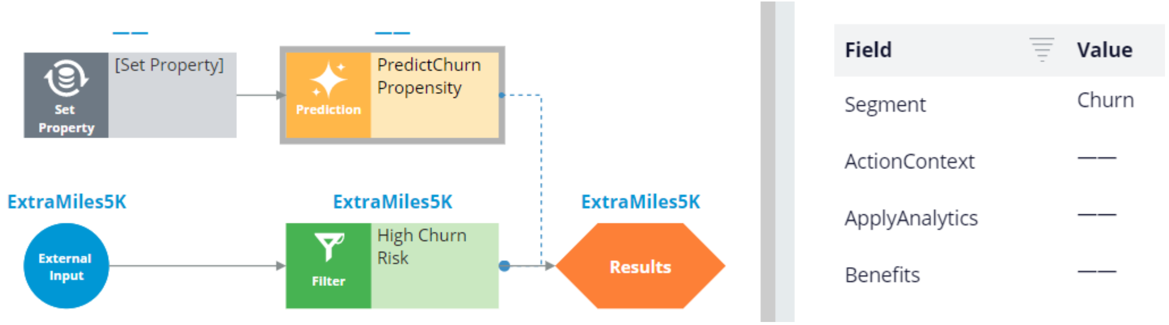
The filter condition is defined to output a retention offer when the pxSegment property of the prediction is equal to **Churn**.



```
PredictChurnPropensity.pxSegment="Churn"
```

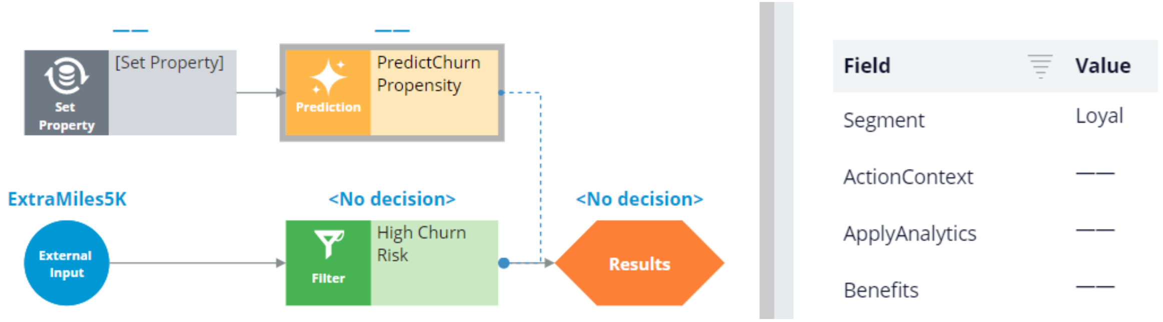

Set Property
[Set Property]
Does not set any properties yet
Prediction
PredictChurn Propensity
Imports <All> from PredictChurnPropensity
External Input
Filter
High Churn Risk
Filters on PredictChurnPropensity.pxSe...
Results

Next, test the strategy using two customer profiles, **Troy** and **Barbara**. For external inputs, consider all available retention offers. The strategy outputs a result for **Troy** because the result of the prediction is **Churn**.



Field	Value
Segment	Churn
ActionContext	——
ApplyAnalytics	——
Benefits	——

The strategy does not have a result for **Barbara**, because the Segment value is **Loyal**.



Field	Value
Segment	Loyal
ActionContext	——
ApplyAnalytics	——
Benefits	——

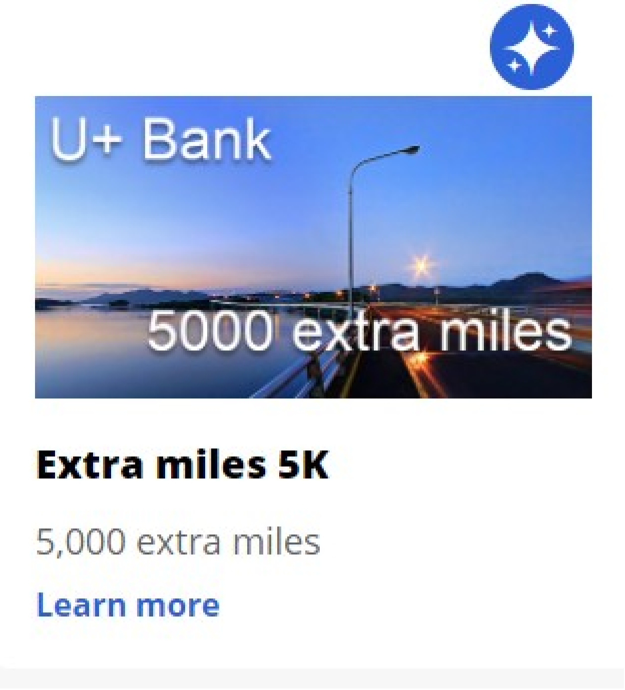
By checking in the strategy, you commit your changes so that they go into effect. You can now use this strategy in the Next-Best-Action Designer engagement policy as an applicability condition.

The first business rule you need to implement is that the **ExtraMiles** group is applicable only to high churn risk customers. To implement this rule, in the **Applicability** section, define a condition for the customer field. Select the **RetentionStrategy**. The condition is: the RetentionStrategy has results for the High Churn Risk component.

The second business rule you need to implement is: U+ Bank wants to show credit card offers to low-risk customers only; meaning the **CreditCards** group is not applicable for high-risk customers. To implement this rule, modify the **Applicability** section of the **CreditCards** group. The condition is: the RetentionStrategy doesn't have results for the High Churn Risk component.

Once the applicability conditions are defined, you need to amend the **Channels** configuration. Because U+ Bank introduced a new group, **ExtraMiles**, which belongs to a new business issue, **Retention**, you need to select the results from the appropriate business structure level. In this case, the bank wants to arbitrate between two different business issues: Sales and Retention. Therefore, select All Issues/All Groups from the business structure level. Saving the configuration implements the business requirement.

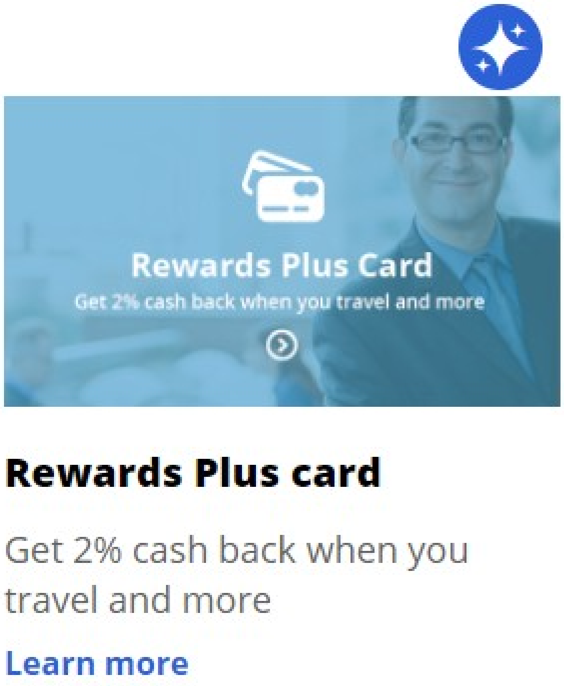
On the U+ Bank website, when you log in as **Troy**, notice that the retention offer is displayed because **Troy** is predicted to churn in the near future.



Extra miles 5K

5,000 extra miles

Now, when you log in as **Barbara**, notice that the credit card offer is displayed because she is predicted to remain loyal for now.



Rewards Plus card

Get 2% cash back when you travel and more

You have reached the end of this demo. What did it show you?

- How to use a prediction in a decision strategy
- How to arbitrate between different groups of actions to display more relevant offers to customers
- How to define applicability rules using a decision strategy in Next-Best-Action Designer

Model governance

Description

AI has the potential to deliver significant benefits, but improper controls can result in regulatory issues, public relations problems, and liability. The Pega T-Switch™ settings in Prediction Studio, which define the transparency thresholds for business issues, help companies to enable users to deploy AI algorithms responsibly and safely.

U+ Bank uses Pega Customer Decision Hub™ to personalize the offers that customers see when they log in to the U+ Bank website.

Customer Decision Hub provides tools to give local and global explanations for the behavior of the adaptive model, such as Customer Profile Viewer for local explanations and the ADM predictor importance report for global explanations. An ethical bias simulation enables you to test your engagement policies for unwanted bias.

Learning objectives

- Explain how Prediction Studio reflects the company's model transparency policy
- Provide local explanations in the Customer Profile Viewer for a single propensity calculation of an adaptive model
- Inspect the ADM predictor importance report to provide global explanations of the factors that drive predictions
- Explain the function of an ethical bias simulation
- Detect unwanted bias in engagement policies

Model transparency

AI has the potential to deliver significant benefits, but without proper controls, it can lead to regulatory issues, public relations problems, and liability.

In Prediction Studio, a senior Data Scientist can set the transparency thresholds within their business to enable users to deploy AI algorithms responsibly and safely. The Pega T-Switch™ settings help companies mitigate potential risks, maintain regulatory compliance, and responsibly provide differentiated experiences to their customers.

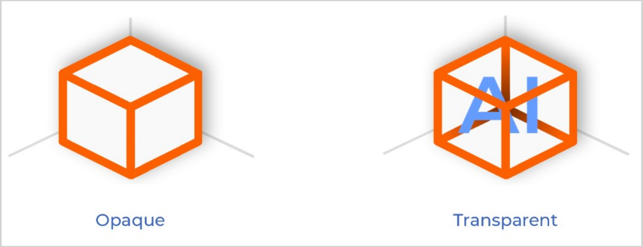
Transcript

Companies in various industries use AI to make predictions and decisions based on those predictions.

To use AI models responsibly, it's important to make sure that the decisions they make can be easily understood by the people who are impacted by them.

Responsible use is especially critical in industries such as finance and healthcare, where AI-driven decisions can have a significant impact on customers. Consumers and regulators demand a high level of trust and transparency in the models that drives these impactful decisions.

Predictive models can be either opaque, making it difficult or impossible to understand how a model reached a decision, or more transparent.



Various factors, such as the complexity and interpretability of the model determine the transparency of a model type.

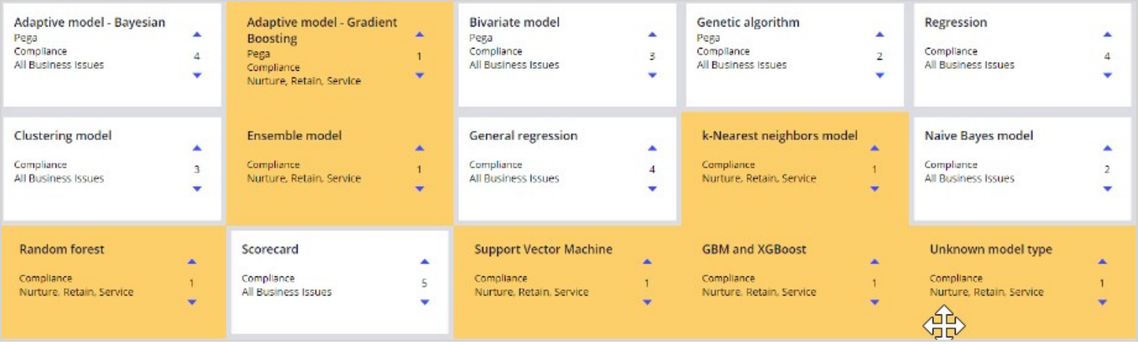
Pega Customer Decision Hub™ includes the tools with which companies can employ AI models with confidence in their implementations and avoid ethical and regulatory issues. Prediction Studio has a dedicated page for model transparency.

Each model type that Pega Platform™ includes receives a recommended transparency score on a scale of 1 to 5, with 1 being the least transparent and 5 being the most transparent. For example, linear regression models are considered more transparent than neural network models because they are simpler and easier to interpret. The adaptive Bayesian models that drive the out-of-the-box Customer Decision Hub predictions are highly transparent as they include model monitoring, reporting, and propensity explanations.

The required transparency may differ between business issues. Marketing might call for the use of more complex models. Still, when dealing with a business issue such as collections or a risk assessment, decisions must be highly explainable to both regulators and customers.

By default, all models are allowed for all business issues. In the development phase of a Pega Customer Decision Hub project, the project lead coordinates with stakeholders to collect the requirements for the transparency policy.

In Prediction Studio, a senior Data Scientist then implements the transparency policy by adjusting the business issue thresholds. Depending on the company policy, model techniques are marked as compliant or non-compliant for a specific business issue.



Adaptive model - Bayesian Pega Compliance All Business Issues ▲ 4 ▼	Adaptive model - Gradient Boosting Pega Compliance Nurture, Retain, Service ▲ 1 ▼	Bivariate model Pega Compliance All Business Issues ▲ 3 ▼	Genetic algorithm Pega Compliance All Business Issues ▲ 2 ▼	Regression Compliance All Business Issues ▲ 4 ▼
Clustering model Compliance All Business Issues ▲ 3 ▼	Ensemble model Compliance Nurture, Retain, Service ▲ 1 ▼	General regression Compliance All Business Issues ▲ 4 ▼	k-Nearest neighbors model Compliance Nurture, Retain, Service ▲ 1 ▼	Naive Bayes model Compliance All Business Issues ▲ 2 ▼
Random forest Compliance Nurture, Retain, Service ▲ 1 ▼	Scorecard Compliance All Business Issues ▲ 5 ▼	Support Vector Machine Compliance Nurture, Retain, Service ▲ 1 ▼	GBM and XGBoost Compliance Nurture, Retain, Service ▲ 1 ▼	Unknown model type Compliance Nurture, Retain, Service ▲ 1 ▼

You have reached the end of this video. You have learned:

- How each model type that comes with Pega Platform has a recommended transparency score.
- How Prediction Studio reflects the model transparency policy of the company.

Explainable AI

U+ Bank uses Pega Customer Decision Hub™ to personalize the offers that customers see when they log in to the U+ Bank website. The adaptive models that calculate the propensity that a customer clicks on the offer are highly transparent. Customer Decision Hub includes tools that give local and global explanations for the adaptive model behavior.

Local explanations enhance the ability to understand the reasoning behind a single propensity calculation made by the model. Customer Profile Viewer offers local explanations of the decisions made by Customer Decision Hub, including the propensities calculated by the adaptive models that drive these decisions.

Global explanations refer to predictor importance in an adaptive model. Customer Decision Hub has built-in support to calculate predictor importance by using decision trees and random forests.

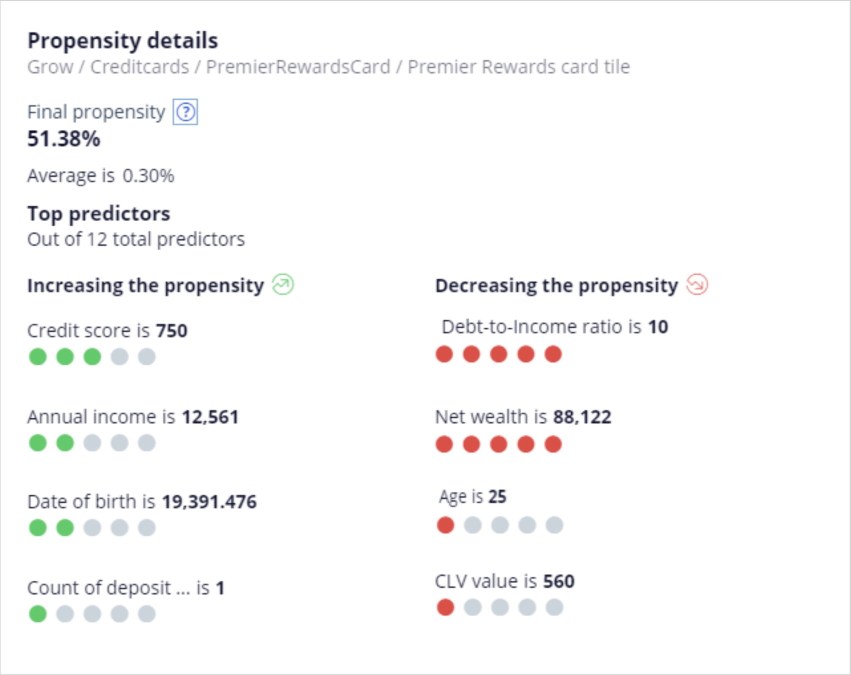
Transcript

Local and global explanations support transparency in how predictive models calculate propensity. Local explanations enable understanding the rationale behind a single propensity calculation made by a predictive model, and they focus on understanding how the input predictors influence that single calculation. Global explanations focus on the overall behavior of a model and provide an understanding of which predictors are most important across all customers. Both local and global explanations are important to ensure that the models behave as expected and to build trust with customers and regulators in the use of AI.

U+ Bank uses Pega Customer Decision Hub to optimize customer interactions on the bank's website, and the Customer Profile Viewer report provides local explanations of the propensities calculated by the out-of-the-box adaptive models. You can review basic customer information for Joanna, including key indicators and demographic data. You can review recent interactions with Joanna in the timeline and how she responded to the decisions made by Customer Decision Hub. You can filter the interactions by the outcome and time frame.

Customer Profile Viewer also shows the current prioritization of next-best-action results. For the U+ Bank website channel, the direction is inbound, and TopOffers is the real-time container that connects with the website. Joanna is eligible for two credit card offers as determined by engagement policies and constraints. The adaptive models calculate the propensities for these offers. The final propensity used in the arbitration formula may differ from the model propensity, for example, when the customer is part of a model control group or when the adaptive model is still in the initial learning phase.

For example, U+ Bank can explain the decision to offer customer Joanna the offer with the highest propensity, the RewardsPlus credit card. The credit score and annual income predictors increase the propensity for Joanna to click on the personalized credit card offer. In contrast, the debt-to-income ratio and net wealth predictors decrease it. This explanation is valid for the RewardsPlus credit card offer to Joanna at this point in time.



Propensity details
Grow / Creditcards / PremierRewardsCard / Premier Rewards card tile

Final propensity
51.38%

Average is 0.30%

Top predictors
Out of 12 total predictors

Increasing the propensity	**Decreasing the propensity**
Credit score is **750** ●●●○○	Debt-to-Income ratio is **10** ●●●●●
Annual income is **12,561** ●●○○○	Net wealth is **88,122** ●●●●●
Date of birth is **19,391.476** ●●○○○	Age is **25** ●○○○○
Count of deposit ... is **1** ●○○○○	CLV value is **560** ●○○○○

In machine learning, predictor importance refers to the relative importance of each predictor in predicting the target variable. Customer Decision Hub includes built-in support for calculating predictor importance using decision trees and random forests. These techniques determine the relative contribution of each predictor to the predictions made by the adaptive model.

In Prediction Studio, the ADM predictor importance report lists the predictor importance score for each predictor, which indicates their relative importance in the model. By analyzing predictor importance, you can offer global explanations of the factors that drive the predictions made by the adaptive models. For example, when you select the Age predictor, you see that the importance of this predictor is relatively high in the adaptive model for the Premier Rewards credit card offer. You can export the report for further analysis.

You have reached the end of this video. You have learned:

- How Customer Profile Viewer provides data on past interactions with a customer.

- How Customer Profile Viewer provides local explanations for the decisions that Customer Decision Hub makes for a customer.

- How Prediction Studio supports global explanations of the factors that drive predictions.

Ethical bias

An ethical bias simulation enables you to test your engagement policies for unwanted bias. That is, you can check if your conditions discriminate based on age, gender, ethnicity, or any other attributes specific to your business scenario. This is particularly useful during the testing stage, while developing and changing actions.

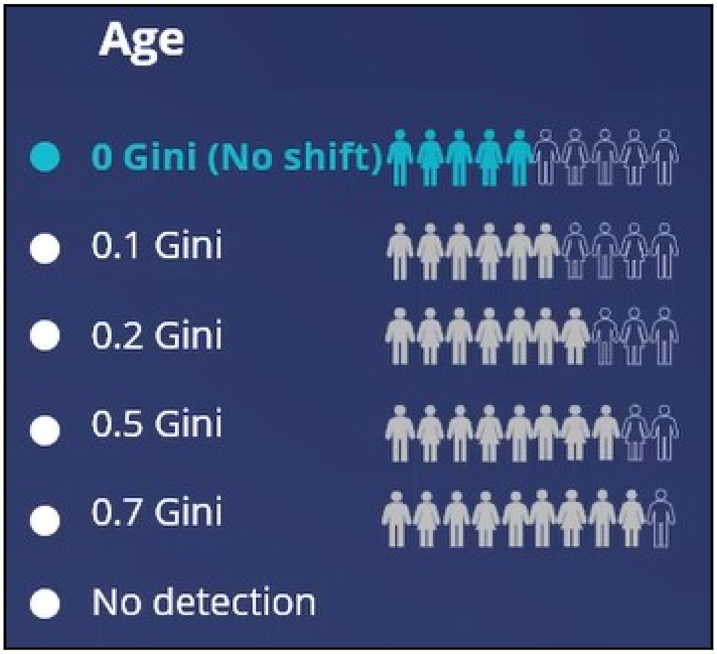
Transcript

This video explains what ethical bias is and how you can avoid unwanted bias in your engagement policies.

Ethical bias testing helps check your engagement policies for unwanted bias. That is, you can test if your conditions discriminate based on age, gender, ethnicity or any other attributes specific to your business scenario.

An ethical bias policy forms the base of an ethical bias simulation. This policy will include the bias fields and the thresholds for each field. You can include any property from your customer class. For instance, age and gender properties are typical properties you might want to include in bias testing.

As age is a numerical field, a Gini coefficient is used to calculate the bias. This is a method of measuring the statistical inequality of a value distribution. A Gini coefficient of 0 represents perfect distribution equality. You can select a warning threshold between 0 (warn if any bias is detected) and 0.7 (warn only if very high bias is detected). You can also choose not to check for bias within a particular business issue.

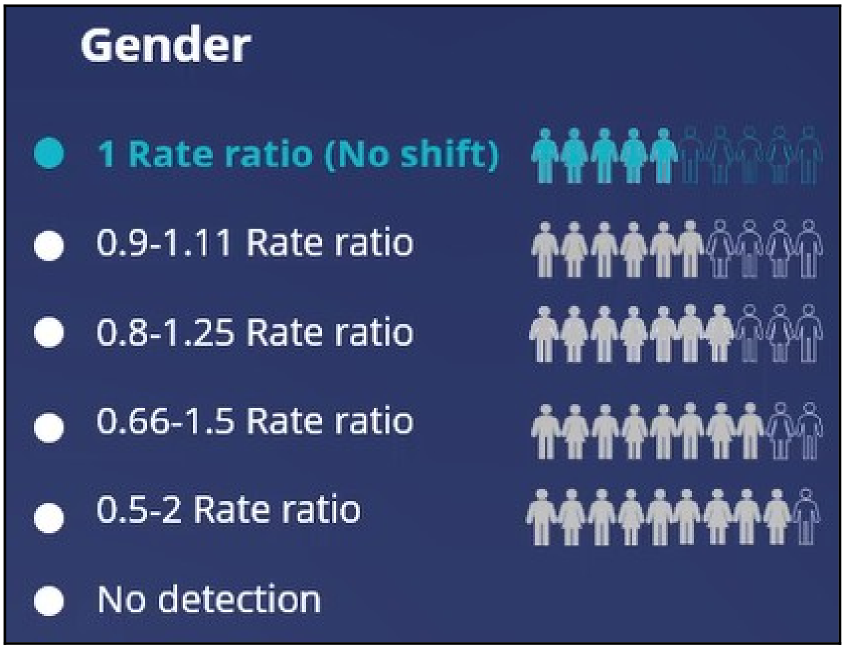


Age

- ● 0 Gini (No shift)
- ○ 0.1 Gini
- ○ 0.2 Gini
- ○ 0.5 Gini
- ○ 0.7 Gini
- ○ No detection

As Gender is a nominal value, a rate ratio is used to determine bias. A rate ratio is used to determine bias for categorical fields by comparing the number of customers who were selected for an action to those not selected for an action, and correlating that to the selected bias field.

A rate ratio of 1 represents perfect distribution equality. You can select a warning threshold between 1, warn if any bias is detected that significantly deviates from a rate ratio of 1, and 0.50 - 2.00, warn only if very high bias is detected. You can also choose to ignore this bias field for a particular issue in your business structure.



Gender

- ● 1 Rate ratio (No shift)
- ○ 0.9-1.11 Rate ratio
- ○ 0.8-1.25 Rate ratio
- ○ 0.66-1.5 Rate ratio
- ○ 0.5-2 Rate ratio
- ○ No detection

Even though no bias is ideal, it is not practical to set the threshold at that level as that would block any actions from reaching customers. Hence, in real business scenarios, a little bias is often allowed to prevent customers from receiving no actions.

Detecting unwanted bias in engagement policies

Ethical bias testing checks your engagement policies for unwanted bias. That is, you can test if your conditions discriminate based on age, gender, ethnicity, or any other attributes specific to your business scenario.

Transcript

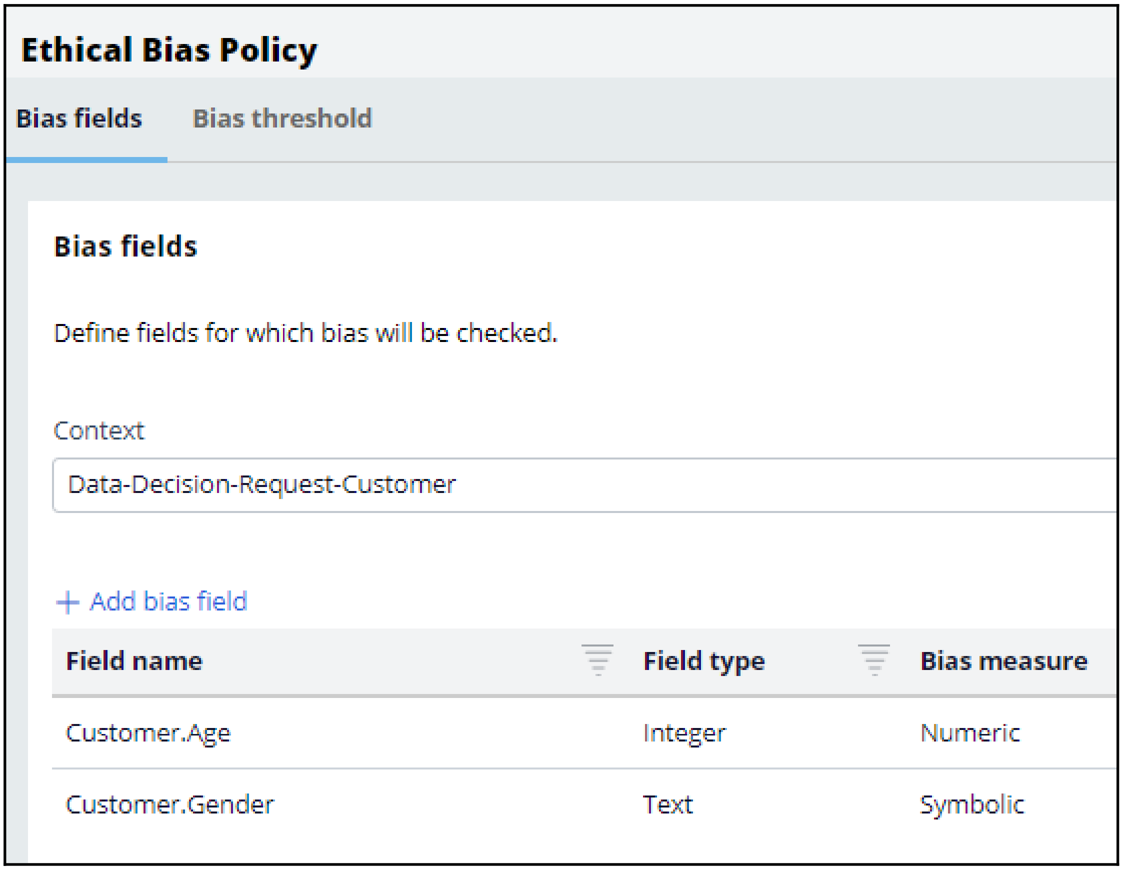
This demo will show you how to run an ethical bias simulation and identify any unwanted bias in the engagement policy conditions.

U+, a retail bank, recently updated their engagement policy conditions to present credit card offers to qualified customers. Now they would like to run an ethical bias simulation to ensure there is no unwanted bias based on age or gender before they push the changes to the live production environment.

This is the Pega Customer Decision Hub™ portal.

To create a simulation, first configure the ethical bias policy. This policy will include the bias fields and threshold. You can select any property from your customer class. In this case, you will use the age and gender properties for the test, which are the most commonly used properties for bias testing. The age property value is a number. In the **Add bias** field window, specify it as an ordinal number. Age is an example of an ordinal number.

Next, add gender as a bias field.



Ethical Bias Policy

Bias fields

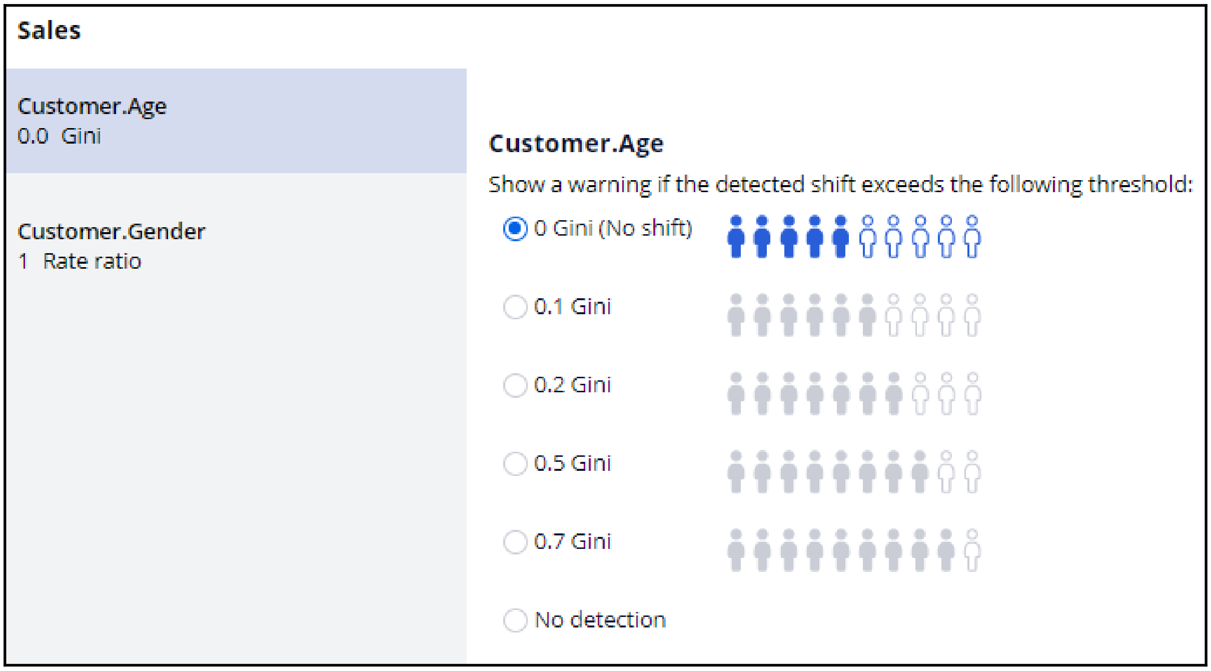
Define fields for which bias will be checked.

Context: Data-Decision-Request-Customer

Field name	Field type	Bias measure
Customer.Age	Integer	Numeric
Customer.Gender	Text	Symbolic

Now, on the **Bias threshold** tab, review and configure the bias threshold settings for each property you selected. This configuration is done at the business issue level. This means, for every business issue, you can decide how much bias to allow. For instance, for the risk issue, you might want to disallow any bias, whereas for sales, you might allow some bias, if you designed a special offer for a specific age category or a specific gender. The bias threshold measurement depends on the type of field that you select. By default, the threshold is set for no detection for these fields.

In this case, the bank does not want to discriminate on age, so select the appropriate threshold value.



Sales

Customer.Age
0.0 Gini

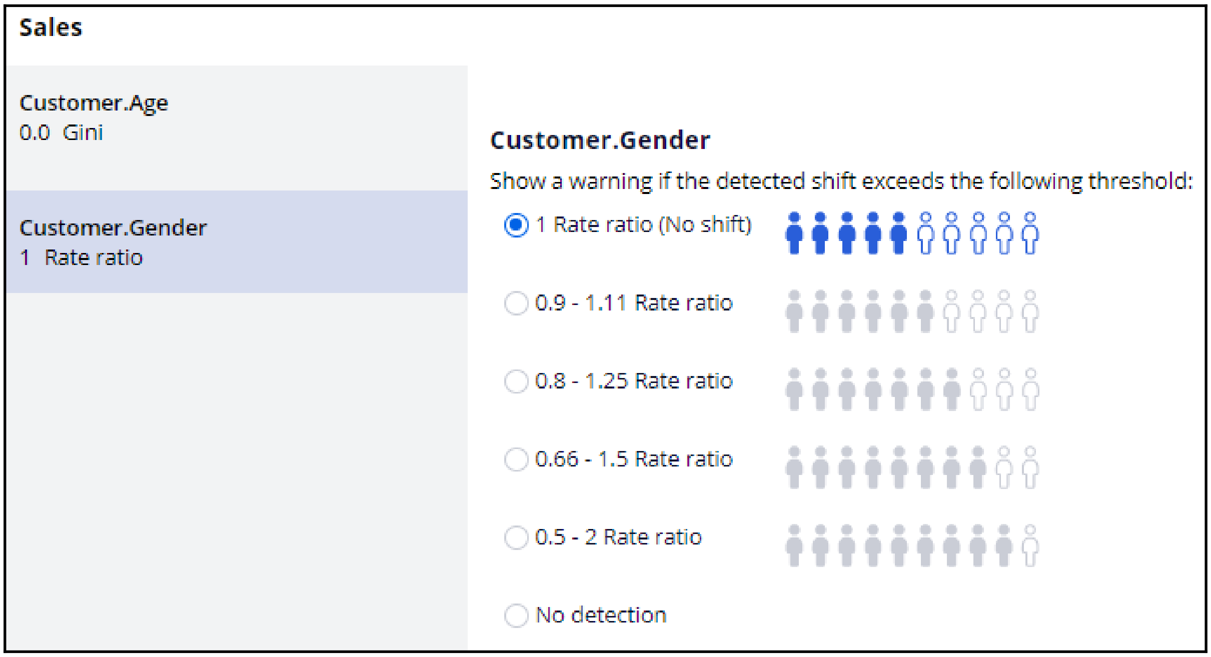
Customer.Gender
1 Rate ratio

Customer.Age

Show a warning if the detected shift exceeds the following threshold:

- ● 0 Gini (No shift)
- ○ 0.1 Gini
- ○ 0.2 Gini
- ○ 0.5 Gini
- ○ 0.7 Gini
- ○ No detection

The bank also does not want to discriminate on gender, so select the appropriate threshold value. Save the changes.



Sales

Customer.Age
0.0 Gini

Customer.Gender
1 Rate ratio

Customer.Gender

Show a warning if the detected shift exceeds the following threshold:

- ● 1 Rate ratio (No shift)
- ○ 0.9 - 1.11 Rate ratio
- ○ 0.8 - 1.25 Rate ratio
- ○ 0.66 - 1.5 Rate ratio
- ○ 0.5 - 2 Rate ratio
- ○ No detection

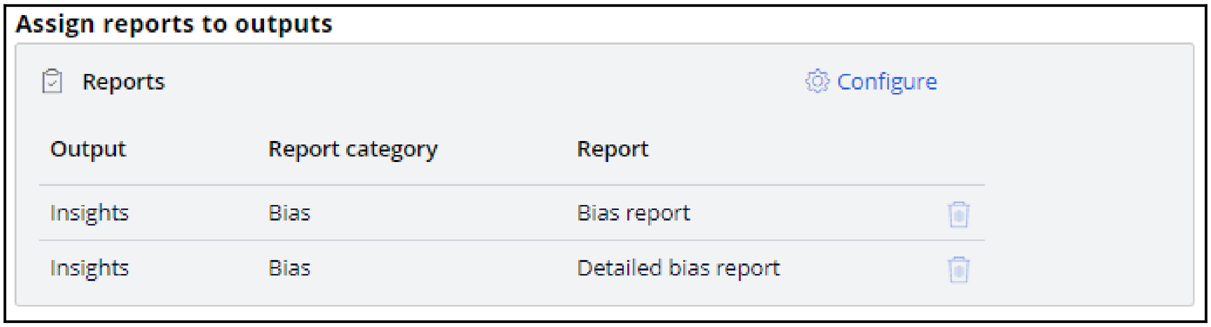
Navigate to the **Simulations** landing page to create and run an **Ethical bias** simulation.

On the Simulation Creation page, select the top-level strategy on which you would like to run the simulation. Then, select the input population on which you want to execute the simulation.

Now, rename the simulation. This will help you easily identify specific simulation runs.

Note that the simulation results are output to the **Insights** data set. This is an internal dataset where the results will be available for an hour by default.

The ethical bias simulation has two bias reports automatically available as output. The results can be examined post simulation run.

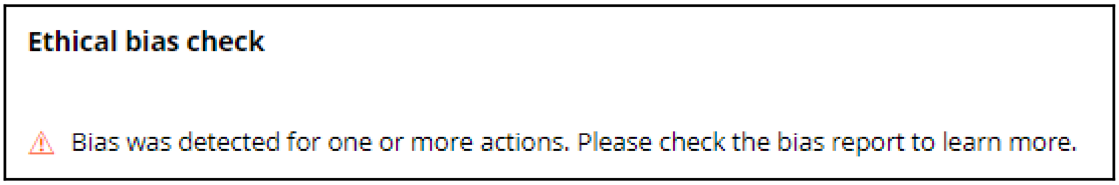


Assign reports to outputs

Reports

Output	Report category	Report
Insights	Bias	Bias report
Insights	Bias	Detailed bias report

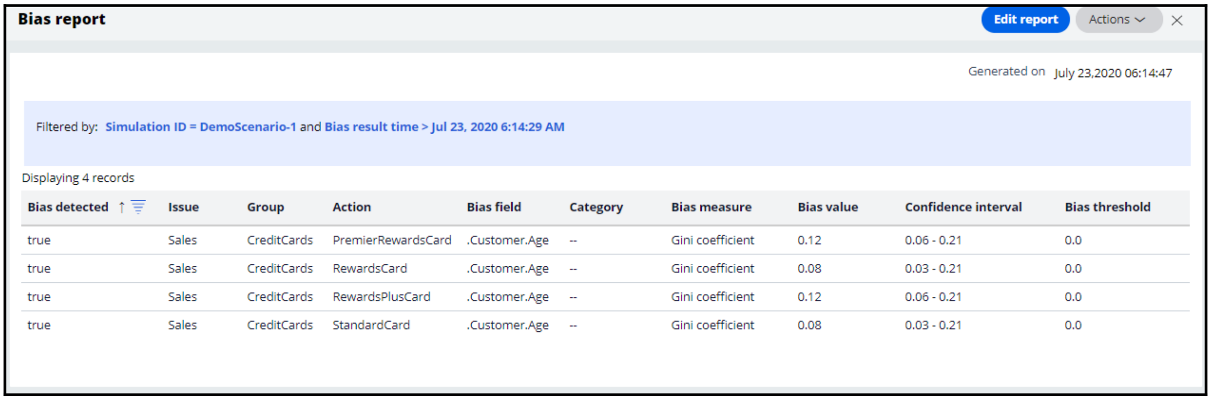
Run the simulation. Once the run is complete, there will be an indication if any unwanted bias is identified. In this scenario, although business did not want any bias, it seems some bias has been detected.



Ethical bias check

⚠ Bias was detected for one or more actions. Please check the bias report to learn more.

Open the generated reports and view the information in detail to understand where the bias was detected. You can sort the report in the bias detected column. Notice that there is bias detected on age. You can check the **Bias value** against the **Confidence interval** to check if the bias value is within the bias threshold range.



Bias report

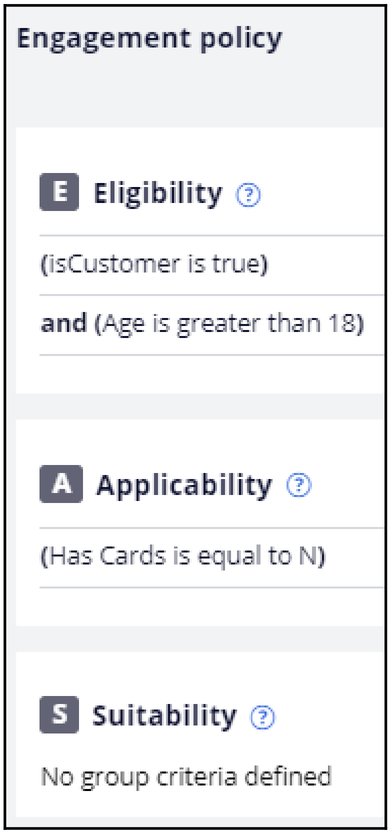
Generated on July 23,2020 06:14:47

Filtered by: Simulation ID = DemoScenario-1 and Bias result time > Jul 23, 2020 6:14:29 AM

Displaying 4 records

Bias detected	Issue	Group	Action	Bias field	Category	Bias measure	Bias value	Confidence interval	Bias threshold
true	Sales	CreditCards	PremierRewardsCard	.Customer.Age	--	Gini coefficient	0.12	0.06 - 0.21	0.0
true	Sales	CreditCards	RewardsCard	.Customer.Age	--	Gini coefficient	0.08	0.03 - 0.21	0.0
true	Sales	CreditCards	RewardsPlusCard	.Customer.Age	--	Gini coefficient	0.12	0.06 - 0.21	0.0
true	Sales	CreditCards	StandardCard	.Customer.Age	--	Gini coefficient	0.08	0.03 - 0.21	0.0

Navigate to Next-Best-Action Designer to view the engagement policy conditions. Note that there is an eligibility condition that uses age. Credit cards are valid only for customers with an age greater than 18. This is a hard eligibility rule that cannot be ignored. It is likely that bias has been detected due to this condition.



Engagement policy

E Eligibility

(isCustomer is true)

and (Age is greater than 18)

A Applicability

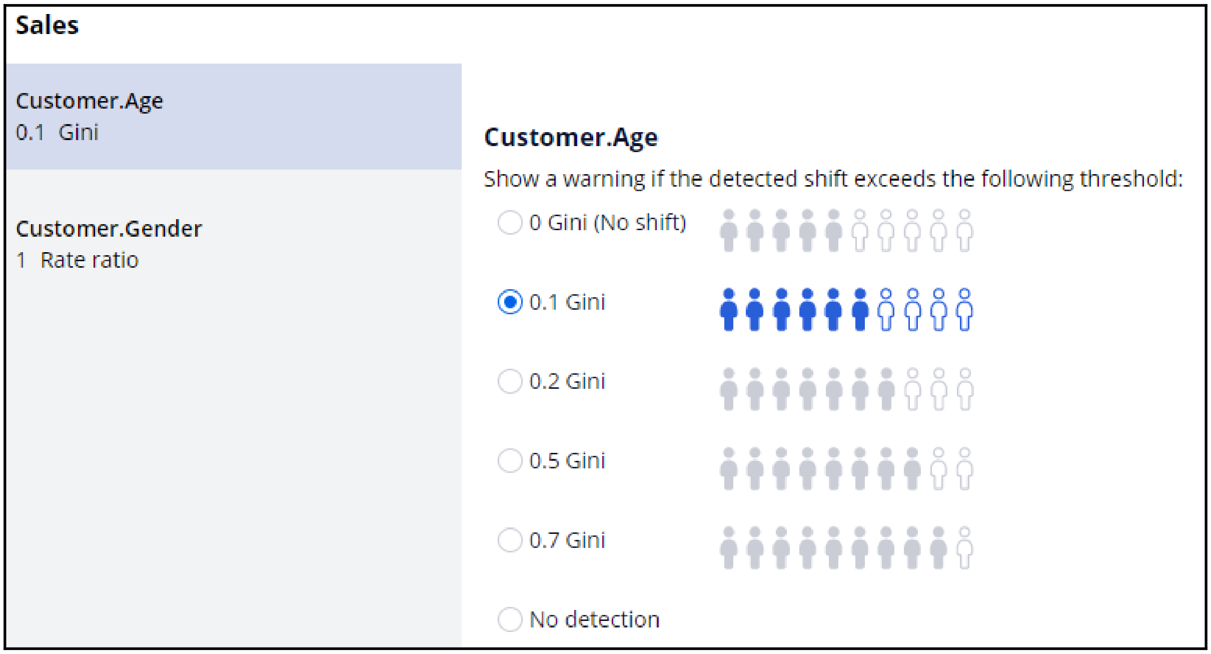
(Has Cards is equal to N)

S Suitability

No group criteria defined

Based on the bank’s policies and regulations, you must review the bias threshold and decide how much bias the business will allow.

To modify the policy, open the Ethical Bias policy. Due to the eligibility condition, which cannot be ignored, business decided to allow a maximum bias of 0.1 Gini. So, increase the bias threshold, and re-run the same ethical bias simulation.



Sales

Customer.Age
0.1 Gini

Customer.Gender
1 Rate ratio

Customer.Age

Show a warning if the detected shift exceeds the following threshold:

○ 0 Gini (No shift)

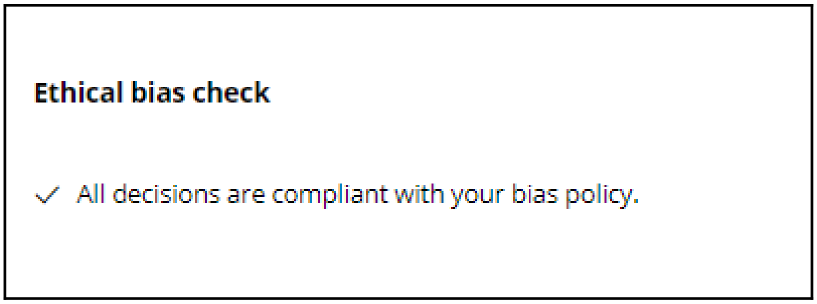
● 0.1 Gini

○ 0.2 Gini

○ 0.5 Gini

○ 0.7 Gini

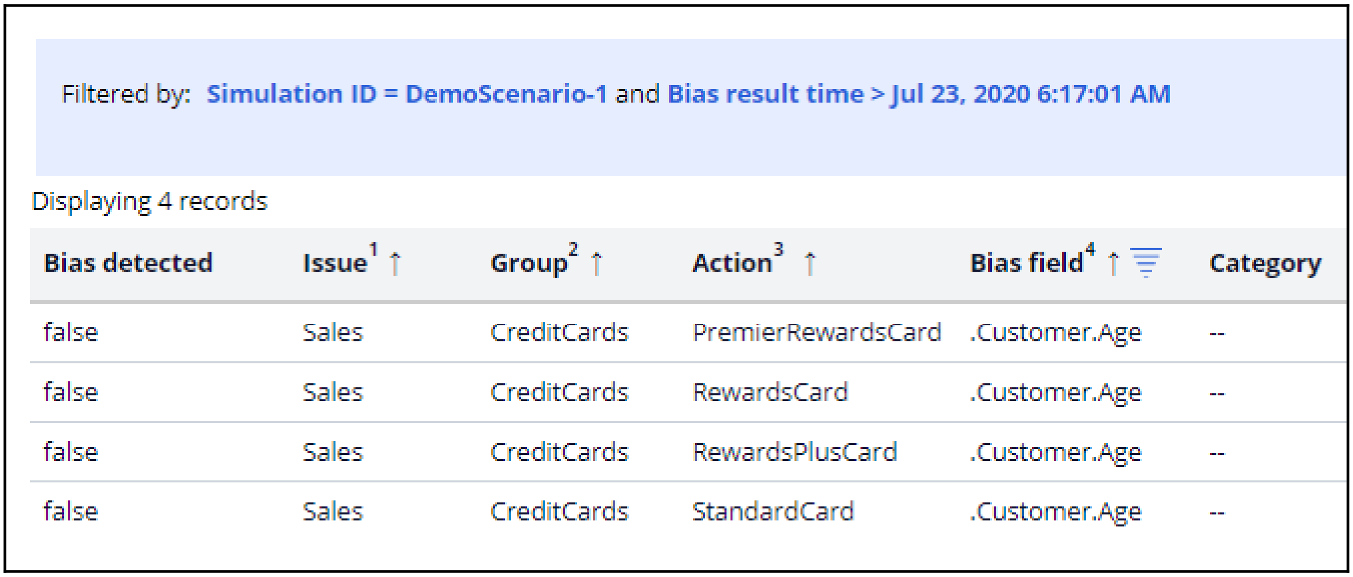
○ No detection



Ethical bias check

✓ All decisions are compliant with your bias policy.

Since the age bias threshold was increased, no bias should be detected.

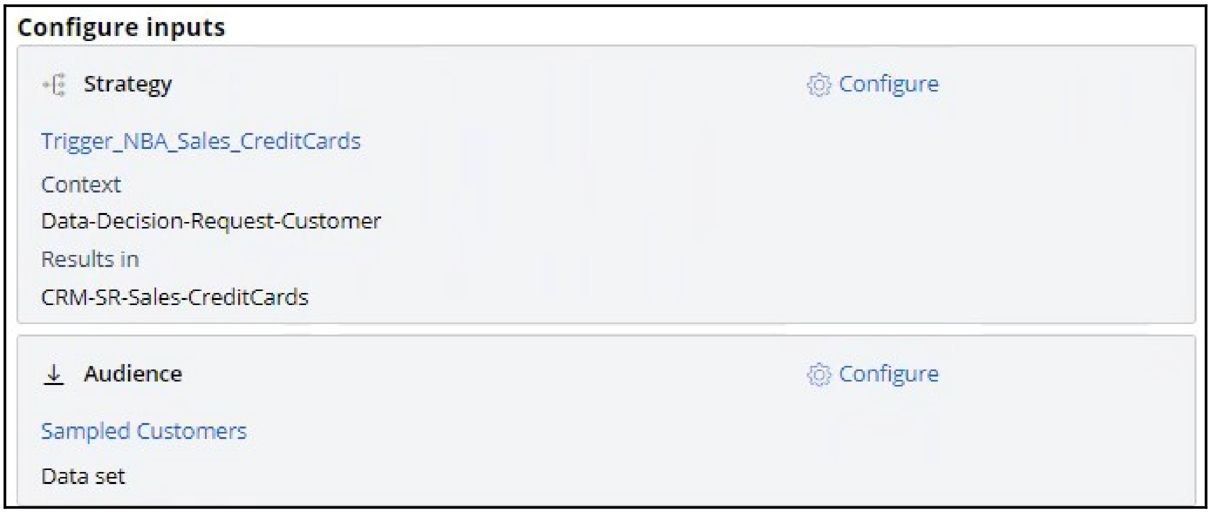


Filtered by: **Simulation ID = DemoScenario-1** and **Bias result time > Jul 23, 2020 6:17:01 AM**

Displaying 4 records

Bias detected	Issue[1] ↑	Group[2] ↑	Action[3] ↑	Bias field[4] ↑	Category
false	Sales	CreditCards	PremierRewardsCard	.Customer.Age	--
false	Sales	CreditCards	RewardsCard	.Customer.Age	--
false	Sales	CreditCards	RewardsPlusCard	.Customer.Age	--
false	Sales	CreditCards	StandardCard	.Customer.Age	--

Now, let's run an ethical bias simulation at the **Trigger_NBA_Sales_CreditCards** strategy level. This strategy includes engagement policy conditions, arbitration, adaptive analytics, constraints, and treatments and channels processing.



Configure inputs

Strategy — Configure

Trigger_NBA_Sales_CreditCards

Context

Data-Decision-Request-Customer

Results in

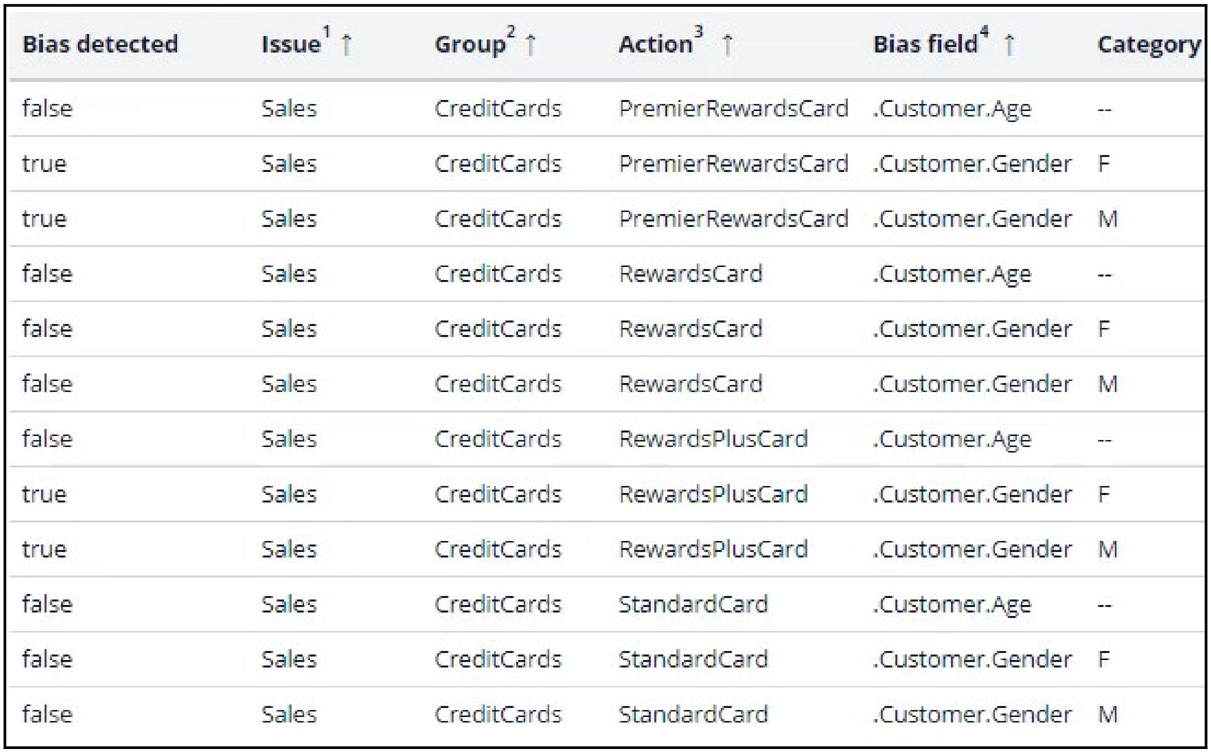
CRM-SR-Sales-CreditCards

Audience — Configure

Sampled Customers

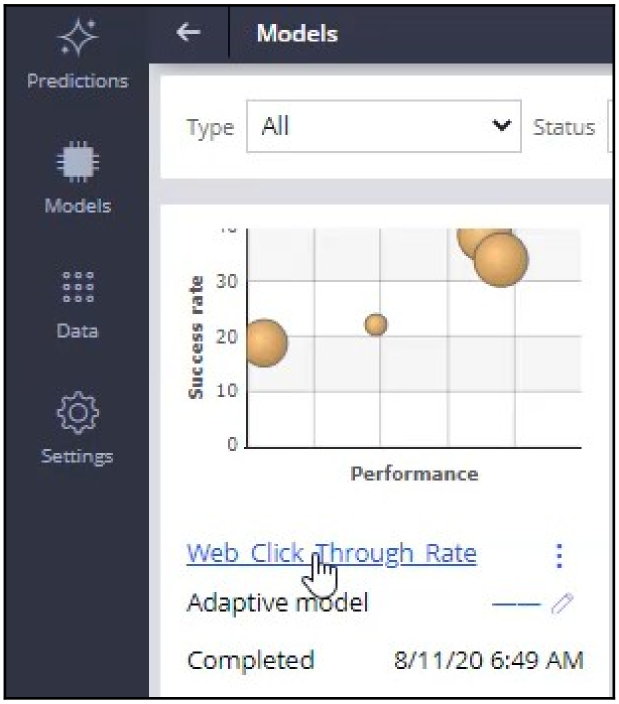
Data set

Note that bias is detected on gender. However, it is at the CreditCards group level, with only engagement policies in scope, thus, there was no bias.



Bias detected	Issue[1] ↑	Group[2] ↑	Action[3] ↑	Bias field[4] ↑	Category
false	Sales	CreditCards	PremierRewardsCard	.Customer.Age	--
true	Sales	CreditCards	PremierRewardsCard	.Customer.Gender	F
true	Sales	CreditCards	PremierRewardsCard	.Customer.Gender	M
false	Sales	CreditCards	RewardsCard	.Customer.Age	--
false	Sales	CreditCards	RewardsCard	.Customer.Gender	F
false	Sales	CreditCards	RewardsCard	.Customer.Gender	M
false	Sales	CreditCards	RewardsPlusCard	.Customer.Age	--
true	Sales	CreditCards	RewardsPlusCard	.Customer.Gender	F
true	Sales	CreditCards	RewardsPlusCard	.Customer.Gender	M
false	Sales	CreditCards	StandardCard	.Customer.Age	--
false	Sales	CreditCards	StandardCard	.Customer.Gender	F
false	Sales	CreditCards	StandardCard	.Customer.Gender	M

Navigate to **Prediction studio** to view the **Web Click Through Rate** adaptive model.



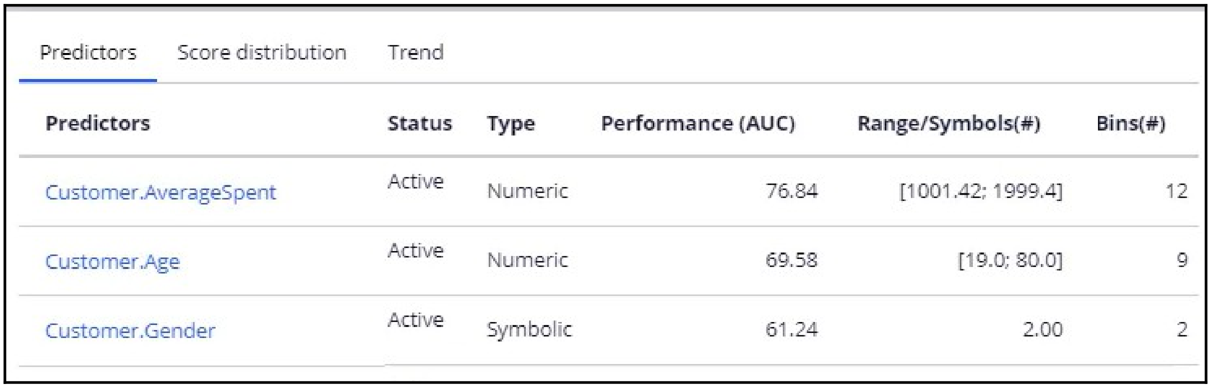
Type: All
Status

Web Click Through Rate

Adaptive model

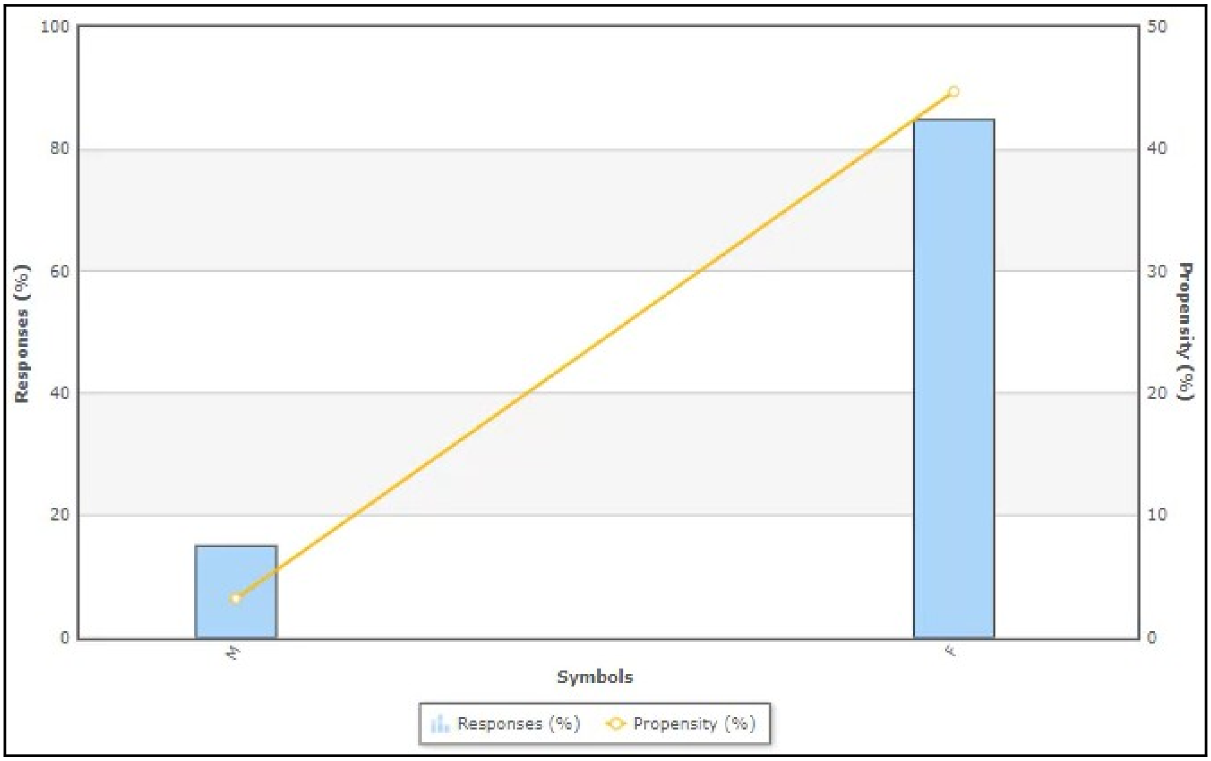
Completed 8/11/20 6:49 AM

Now, open the Premier rewards card model report to view the predictors.



Predictors	Status	Type	Performance (AUC)	Range/Symbols(#)	Bins(#)
Customer.AverageSpent	Active	Numeric	76.84	[1001.42; 1999.4]	12
Customer.Age	Active	Numeric	69.58	[19.0; 80.0]	9
Customer.Gender	Active	Symbolic	61.24	2.00	2

Notice that gender is one of the top predictors. As gender is a predictor, and the model is learning constantly, bias is detected even though there is no engagement policy condition on gender. Open the predictor to view the customer responses. Here you can see that there is a clear difference in the number of times the offer was presented to male customers versus female customers.



If you do not want the adaptive model to learn based on gender, ensure gender is not included as a predictor.

This demo has concluded. What did it show you?

- How to configure an ethical bias policy.
- How to run an ethical bias simulation.
- How to view the auto-generated ethical bias reports to understand the bias threshold deviation.

Pega Process AI overview

Description

Gain a greater understanding of the key features, capabilities, and benefits of Prediction Studio in Pega Process AI™ context. Prediction Studio is the dedicated workspace for data scientists to control the life cycles of predictions and the predictive models that drive them. Configure the predictions that are deployed in Pega Process AI to increase efficiency and effectiveness in case management.

Learning objectives

- Describe the use of Pega Process AI in case management
- Explain the types of predictions that are available in Prediction Studio

Pega Process AI overview

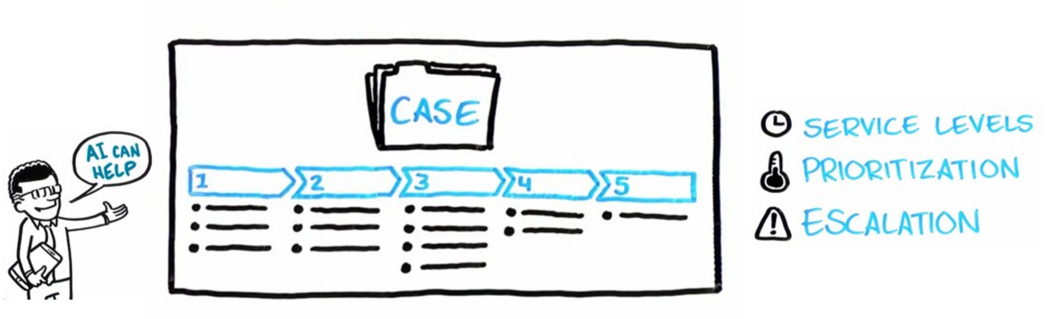
Introduction

In recent years, artificial intelligence has moved out of the labs and helped enterprises generate proven business value. At the same time, operationalizing AI can be a bottleneck. Pega Process AI™ tackles this problem by using AI to self-optimize processes and applying your own AI in Pega case management.

Transcript

This video provides an overview of the Pega Process AI capabilities in intelligent automation.

Process management aims to optimize business processes by increasing efficiency, consistency, and transparency, which decreases costs and improves quality.



For example, consider an online order process. The customer submits an order, and the company processes and then delivers the order.

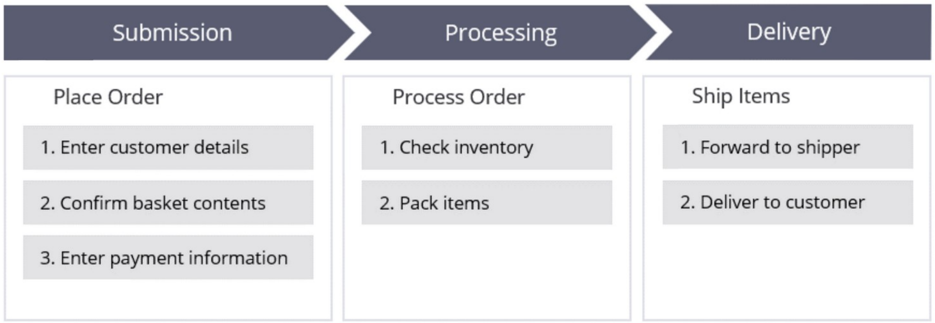
Submit order
Process order
Deliver order

An Pega Platform application that models the online order process follows the same sequence as a series of stages. A **case type** is the abstract model of that process.

Case types model repeatable business transactions that might refer to a customer, or another entity, such as a machine in a maintenance case type.

The **case life cycle** for a case type helps to visualize the work to complete as part of a business transaction.

Each stage in the life cycle contains the steps required to complete it and move to the next stage.



Submission

Place Order

1. Enter customer details
2. Confirm basket contents
3. Enter payment information

Processing

Process Order

1. Check inventory
2. Pack items

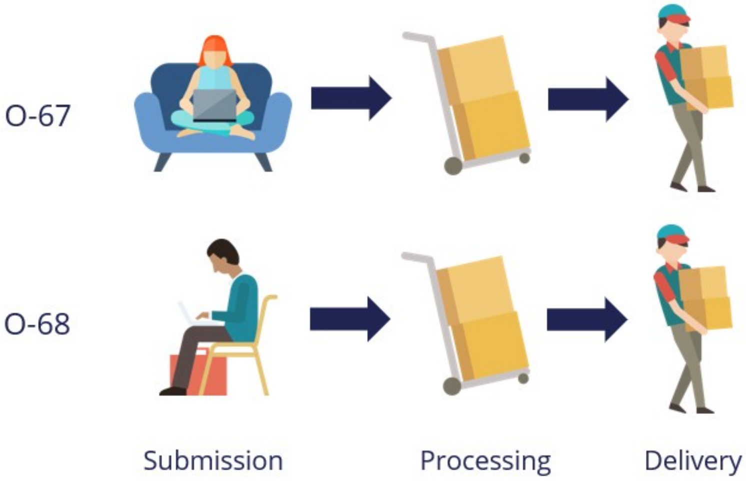
Delivery

Ship Items

1. Forward to shipper
2. Deliver to customer

A **case** is a specific transaction instance of the case type.

Each time a user submits an online order, Pega Platform creates an order case and assigns the case a unique identifier.



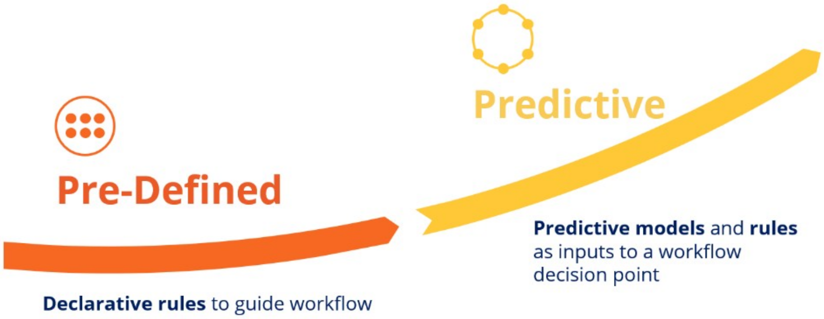
A case type can use declarative rules to manage the workflow, for example, to confirm that the order contains a valid shipping address or the order amount threshold to qualify for free shipping.

Pre-Defined
Declarative rules to guide workflow

Pega Process AI can improve the quality of the decisions in the workflow by weighting in predictions, driven by predictive models.

The first approach is to operationalize existing predictive models that have proven their efficiency, to support the decisions that benefit from predictions, such as credit risk in a sales case or fraud risk in a claims case.

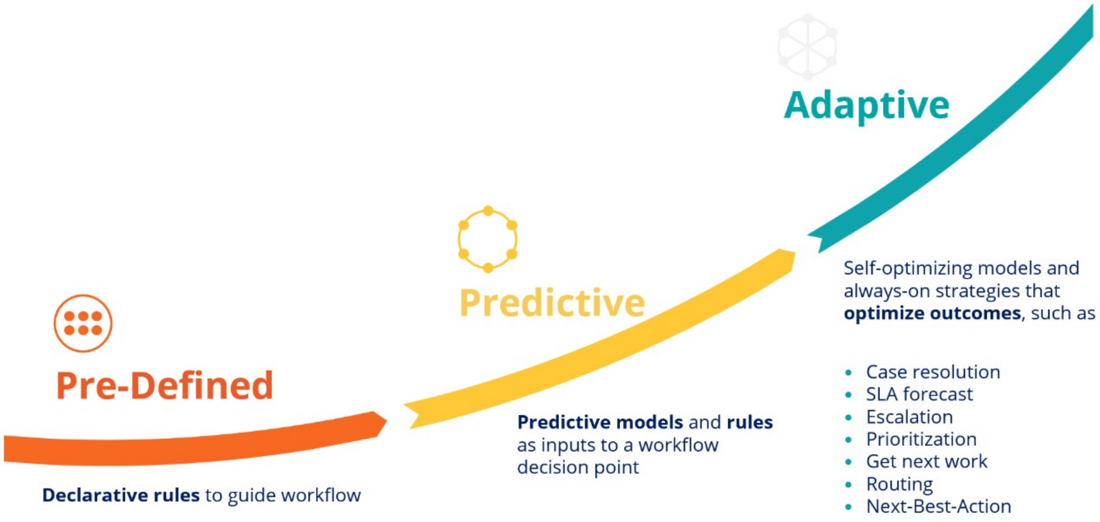
For example, the decision not to process an order can be based on a high credit risk score, and then the application can route the dubious claim for closer inspection.


Predictive
Pre-Defined
Predictive models and rules as inputs to a workflow decision point
Declarative rules to guide workflow

The inputs for such a predictive model can be attributes of the case itself, such as the claimed amount in a claims case type, but they can also include data such as the number of claims submitted recently by the same customer.

You can build predictive models in Prediction Studio, import the models in the PMML and H2O formats, or run externally on the Amazon SageMaker and Google ML platforms to drive a prediction.

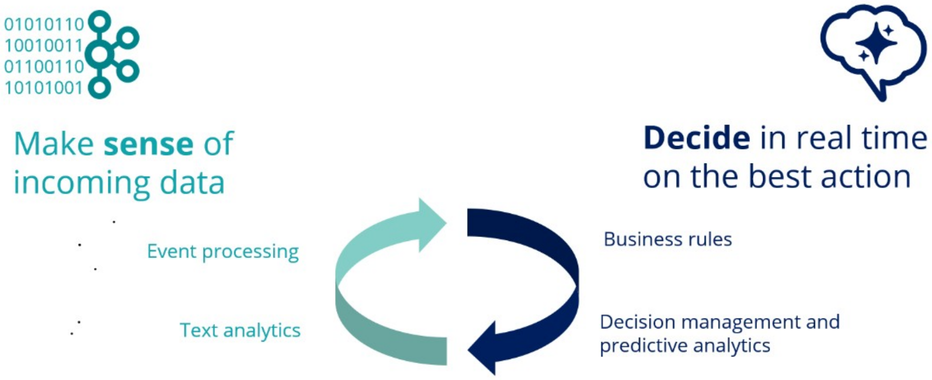
To optimize case outcomes, use adaptive models that can predict outcomes, such as case resolution, or intelligently prioritize and route cases to optimize business value and customer experience.


Pre-Defined
Declarative rules to guide workflow
Predictive
Predictive models and rules as inputs to a workflow decision point
Adaptive
Self-optimizing models and always-on strategies that optimize outcomes, such as
Case resolution
SLA forecast
Escalation
Prioritization
Get next work
Routing
Next-Best-Action

Adaptive models self-optimize by learning from the previous case outcomes that they capture.

The objective of Pega Process AI is to make sense of the incoming data and then decide on the best action to take in a specific stage of the case.

You can enhance the incoming data analysis by event processing to detect patterns of interest in real-time data streams and by natural language processing of incoming text.

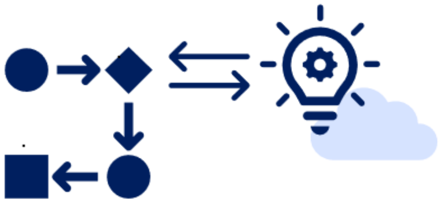

01010110
10010011
01100110
10101001
Make sense of incoming data
Event processing
Text analytics
Decide in real time on the best action
Business rules
Decision management and predictive analytics

The decision is based on the business rules and supported by predictive analytics. This process is repeated every time that the case requests a decision.

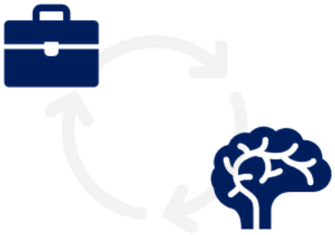
As the number of processed cases increases and model evidence accumulates, the predictive power of the models increases over time.

To summarize, Pega Process AI uses artificial intelligence in case management to produce better business outcomes.

You can use real-time, adaptive case outcome predictions and your own AI models in custom predictions.



Custom AI predictions



Real-time, adaptive **case outcome predictions**

Process AI predictions

Introduction

With the decision management capability of Pega Platform™, you can enhance applications to help optimize business processes, predict customer behavior, analyze natural language, and make informed decisions to better meet the needs of customers and achieve positive business outcomes.

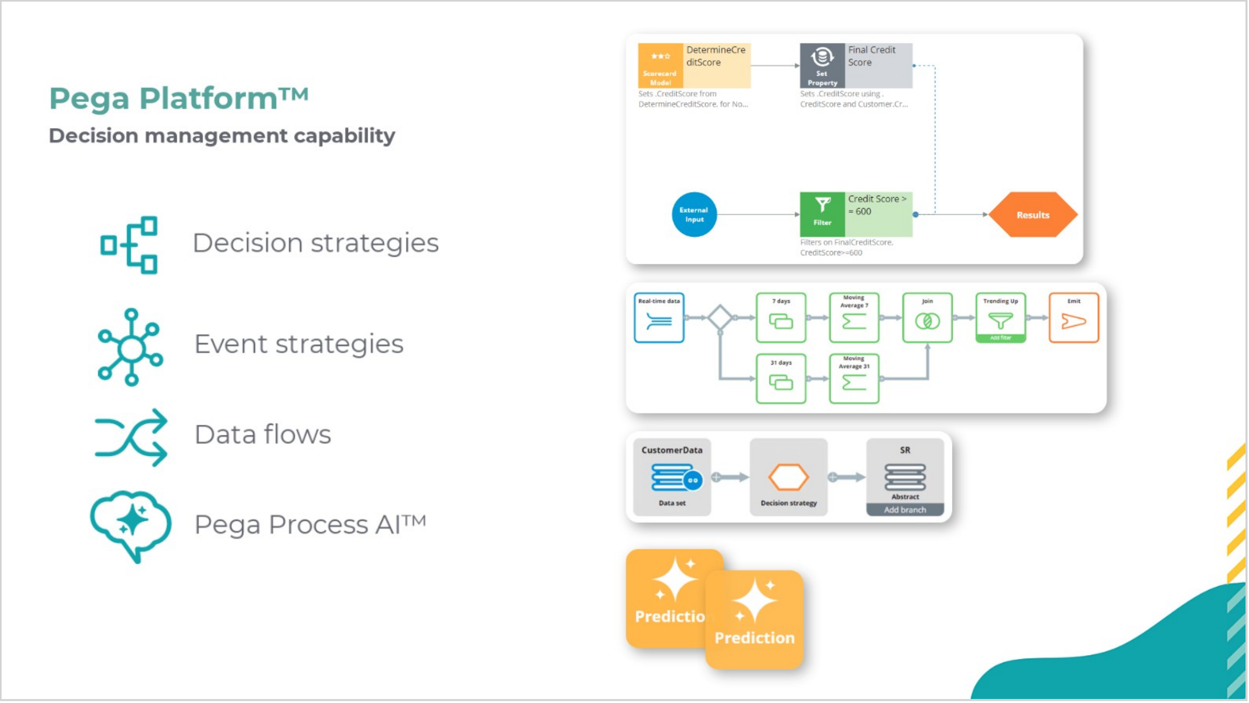
Transcript

This video introduces you to Pega AI, a feature of the decision management capability of Pega Platform.

Other decisioning features of Pega Platform include:

- Decision strategies that feature a business- and user-friendly canvas with which you can create decision logic that uses behavioral and operational data to improve intelligent processes.
- Event strategies to detect patterns in data streams and react to them.
- Data flows as scalable and resilient data pipelines to ingest, process, and move data from one or more sources to one or more destinations.

Decision management uses Pega AI to make predictions about the possibility of fraud, successful case completion, and other subjects to make decisions more relevant.



Pega Platform™

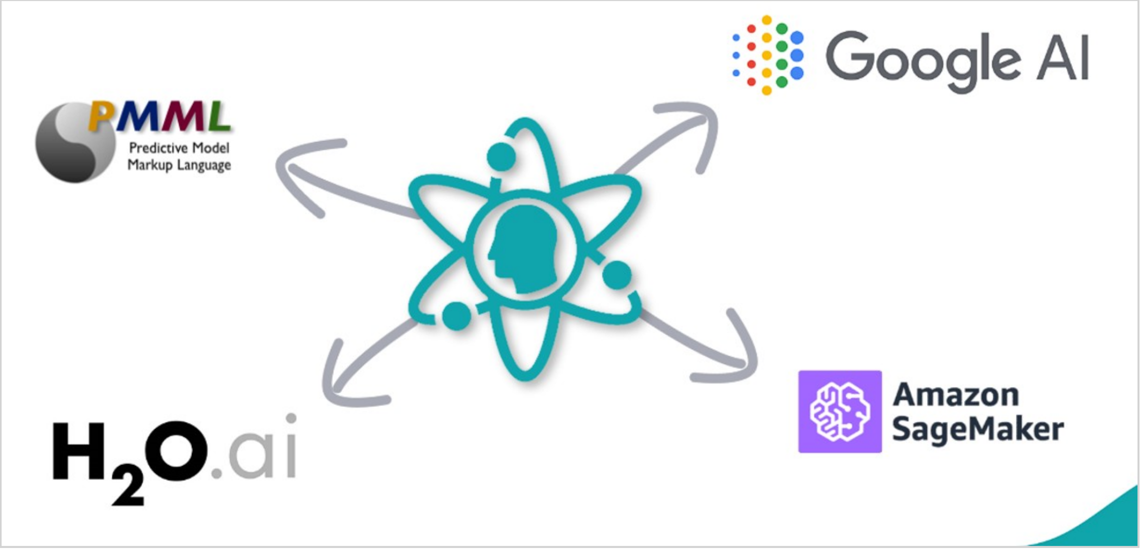
Decision management capability

- Decision strategies
- Event strategies
- Data flows
- Pega Process AI™

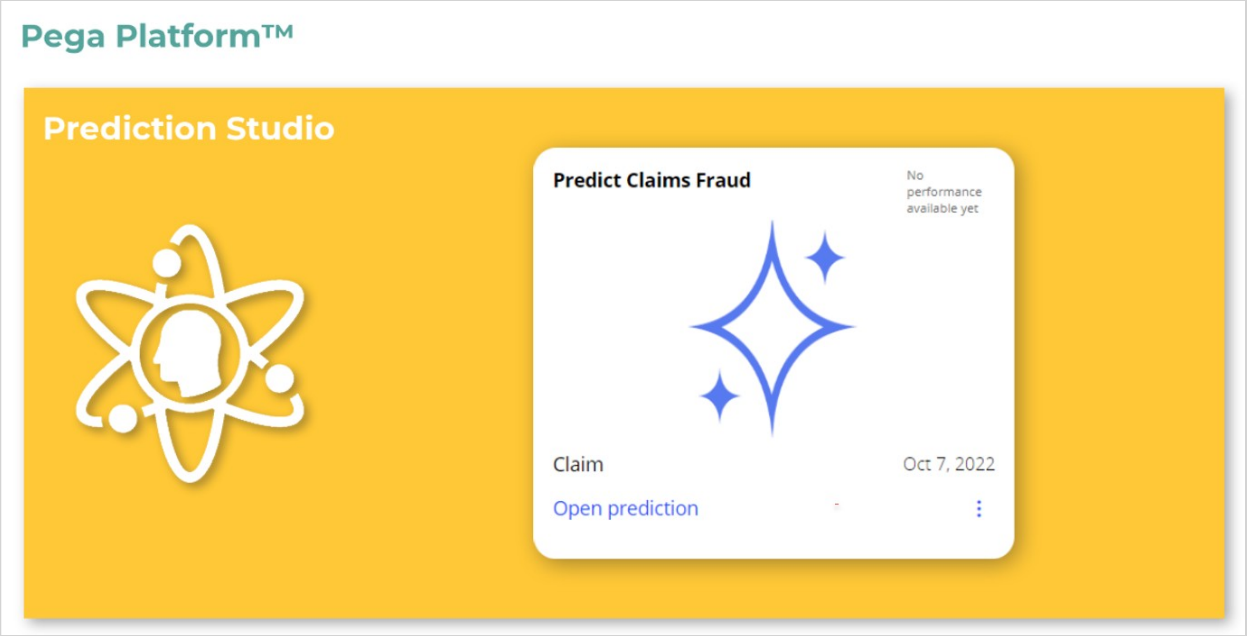
Decision management is a Pega Platform capability. You can apply decision management to any application that is built on Pega Platform.

Various and versatile predictions are available, but one or more predictive models drive them all.

For example, a data scientist can create a predictive model in Pega Platform or an external environment that can export the model as a PMML or H2O file back to Pega Platform. Another option is to connect to a machine learning service such as Google ML or AWS SageMaker.

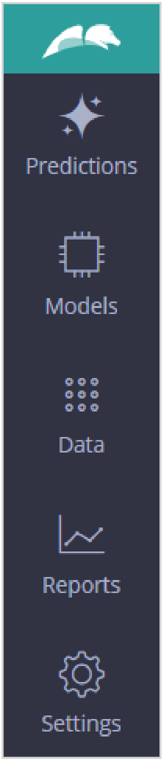


If an insurance company wants to use Pega Process AI™ to route incoming claims that might be fraudulent to an expert, based on the outcome of a predictive model, the data scientist creates a fraud model to drive a new case management prediction in Prediction Studio.

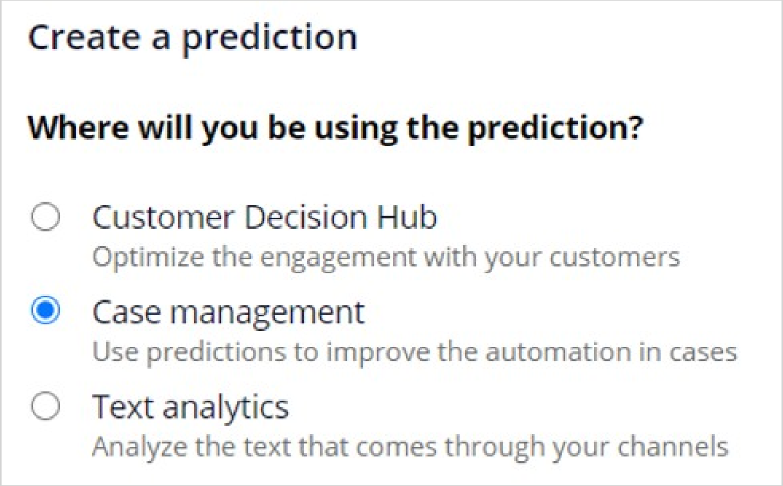


Prediction Studio is the dedicated workspace where you manage the life cycle of predictive models and the predictions that they drive.

The workspace provides data scientists with everything they need to author, deploy, govern, monitor, and change predictions. Prediction Studio has five work areas: **Predictions**, **Models**, **Data**, **Reports**, and **Settings**.



On the Predictions landing page, you create and manage predictions. There are three types of predictions, but Process AI focuses on the **Case management** prediction.



Create a prediction

Where will you be using the prediction?

○ Customer Decision Hub
Optimize the engagement with your customers

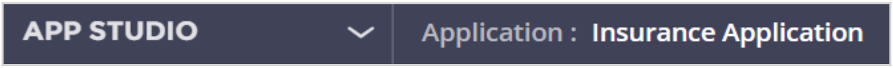
● Case management
Use predictions to improve the automation in cases

○ Text analytics
Analyze the text that comes through your channels

Case management predictions are used in case types to support decisions in business processes. For example, predictive models can help to predict whether an insurance claim is fraudulent or distinguish regular from complex claims.

This dependence routes cases more accurately and strengthens the separation of concerns.

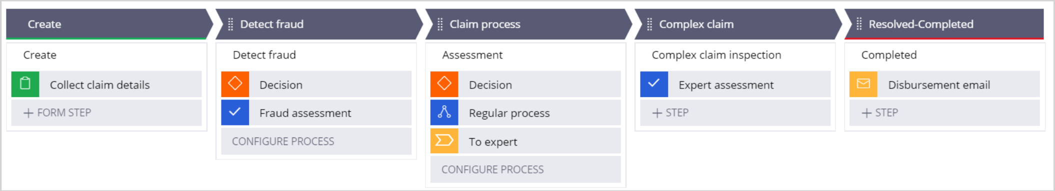
The decision step in a case type uses **case management** predictions.



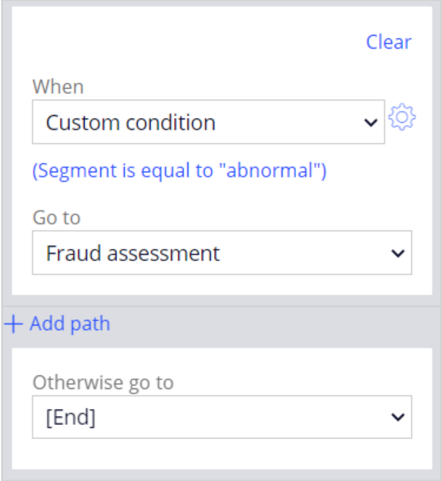
APP STUDIO

Application : Insurance Application

Consider the following case type, which handles incoming car insurance claims:



An application developer can use the outcome of the prediction in the condition of a decision step instead of a business rule. Based on the condition, the system routes a case to a fraud expert when the prediction flags the claim as abnormal.



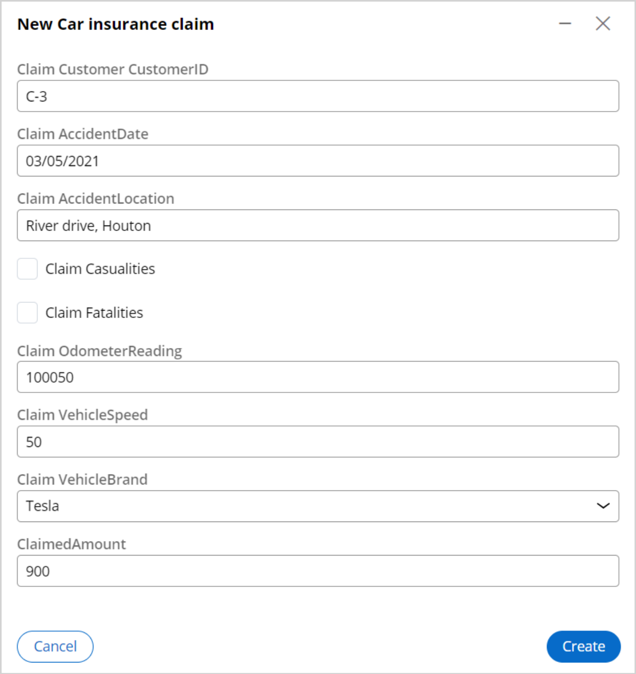
When: Custom condition

(Segment is equal to "abnormal")

Go to: Fraud assessment

Otherwise go to: [End]

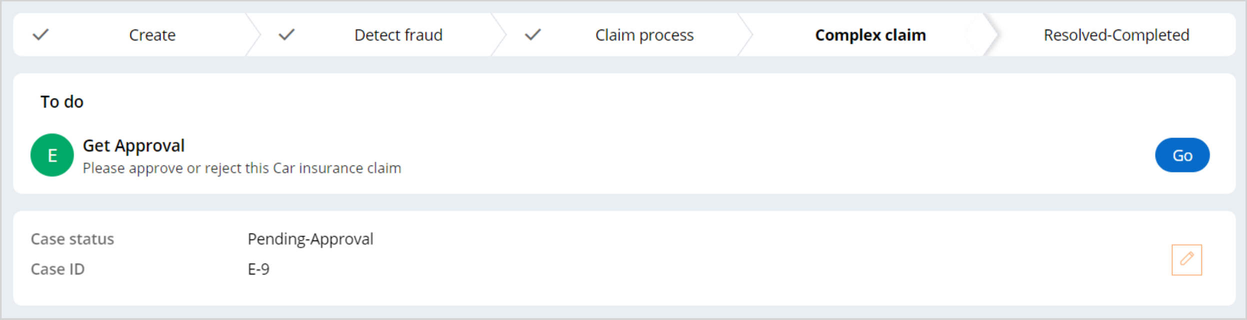
Pega Process AI can also help to distinguish regular from complex claims. It helps speed up the process by identifying such cases early and routing them to the right person.



New Car insurance claim

Claim Customer CustomerID	C-3
Claim AccidentDate	03/05/2021
Claim AccidentLocation	River drive, Houton
□ Claim Casualities	
□ Claim Fatalities	
Claim OdometerReading	100050
Claim VehicleSpeed	50
Claim VehicleBrand	Tesla
ClaimedAmount	900

In the following case, the data scientists create a prediction that aims to identify cases that are likely to miss their deadlines. Then, an application developer uses the prediction outcome when configuring the case type so that the system can automatically route a complex case to a senior employee for evaluation.

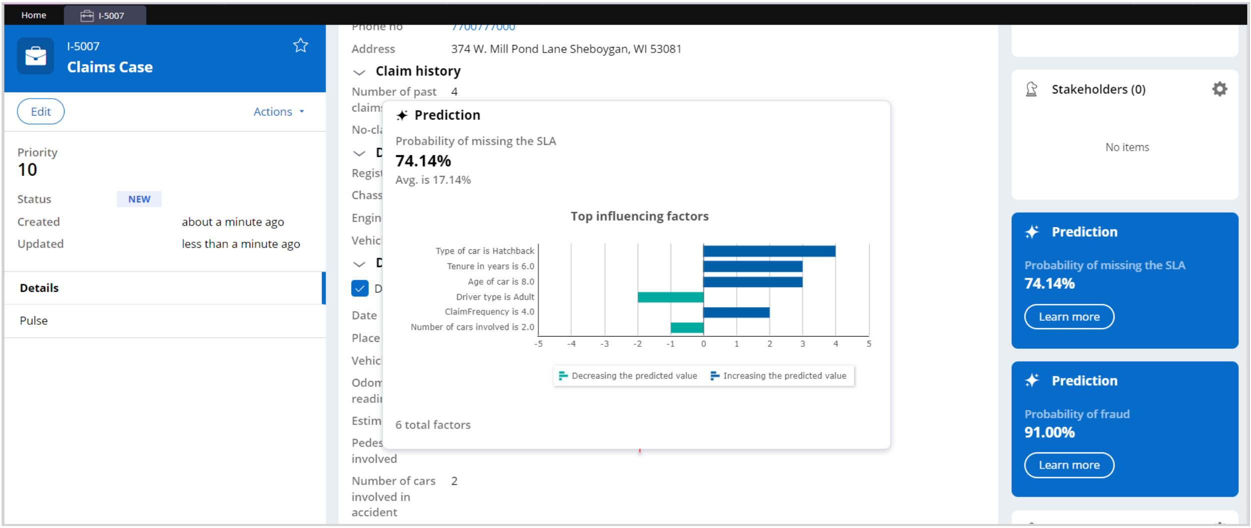


Get Approval

Please approve or reject this Car insurance claim

Case status	Pending-Approval
Case ID	E-9

Some additional helpful information widgets are available in the case view. Notice the prediction widget in the claims case. The widget conveniently displays the output of the trained prediction models, in this case, fraud or missing service-level agreement (SLA) probability. By clicking **Learn more**, you can see the details of the model and whether it received training with adequate data (in this example, the probability of case completion).



I-5007
Claims Case

Priority
10

Status NEW

Created about a minute ago

Updated less than a minute ago

Details

Pulse

Address 374 W. Mill Pond Lane Sheboygan, WI 53081

Claim history

Number of past claims 4

Number of cars involved in accident 2

Prediction

Probability of missing the SLA

74.14%

Avg. is 17.14%

6 total factors

Stakeholders (0)

No items

Prediction

Probability of missing the SLA

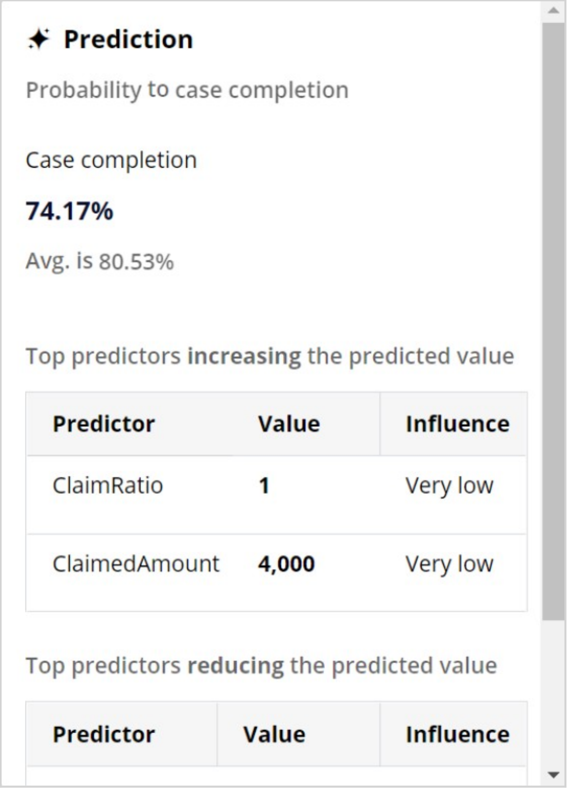
74.14%

Prediction

Probability of fraud

91.00%

The widget also displays which predictors contribute positively (increasing the predicted value), and which contribute negatively, (decreasing the predicted value).



Prediction

Probability to case completion

Case completion

74.17%

Avg. is 80.53%

Top predictors **increasing** the predicted value

Predictor	Value	Influence
ClaimRatio	1	Very low
ClaimedAmount	4,000	Very low

Top predictors **reducing** the predicted value

Predictor	Value	Influence

You have reached the end of this video. You have learned:

- How Pega AI allows you to improve business processes by using predictions.
- How predictive models drive predictions.
- How to create and manage predictions in Prediction Studio.

- How to use predictions in a case type to improve business processes.

Applying NLP for Case classification

Description

Pega Process AI utilizes natural language processing to automatically categorize and route claims based on customer descriptions, eliminating delays and errors associated with manual processing. Learn how to enable AI Case classification, define topics, train models, and validate the accuracy of automatic categorization to streamline claims processing and enhance operational efficiency.

Learning objectives

- Enable AI Case classification in Pega Process AI by creating a new accident category field and defining topics.
- Train and test models in Prediction Studio
- Validate the automatic AI accident category prediction and Case routing by creating a new Claims Case for a test Persona.
- Verify the effectiveness of intelligent Case routing, which automatically routes Cases to the correct Work Queue based on the predicted accident category.

Applying NLP for Case classification

In the complex and dynamic world of insurance, processing claims is an important task. It requires a high degree of precision, efficiency, and speed. However, the traditional approach, which involves manual categorization and routing of Cases, often leads to delays and errors. These issues can negatively impact customer satisfaction and operational efficiency.

Pega Process AI, equipped with advanced features, addresses these challenges. By using the power of natural language processing (NLP), Pega Process AI can automatically identify the accident category based on the Case description provided by the customer in their insurance claim. As shown in the following demo, Pega Process AI not only eliminates the need for manual intervention but also ensures swift and accurate routing of the Case to the correct Work Queue.

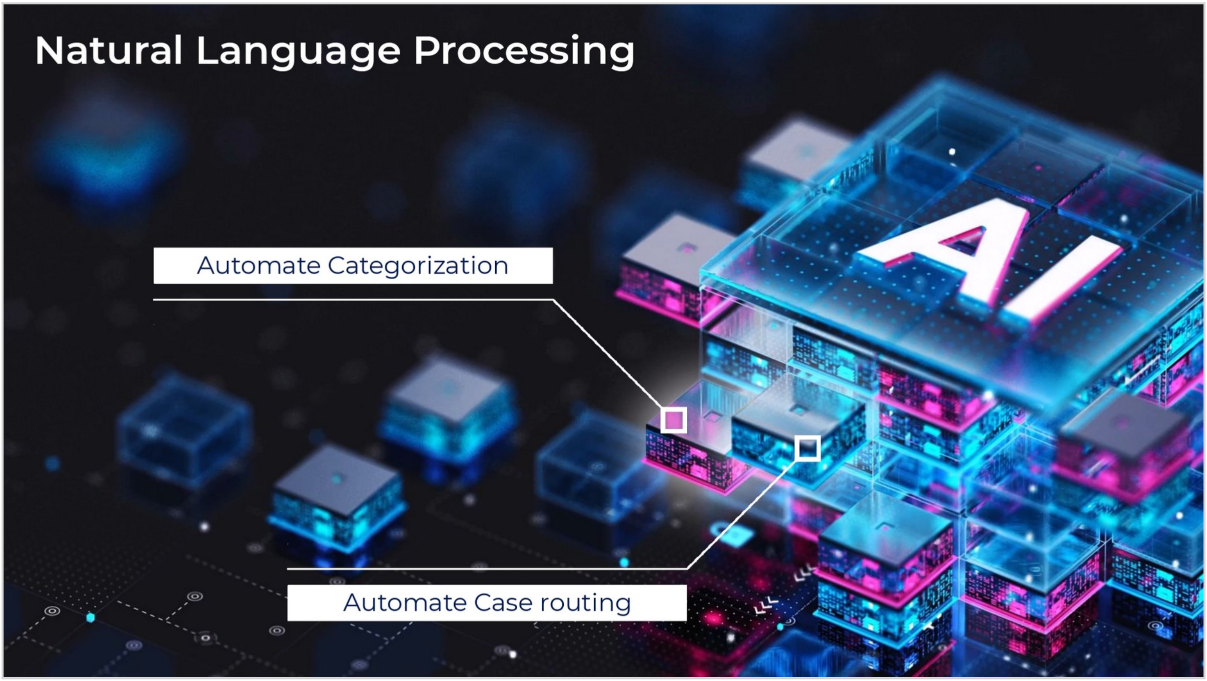
Transcript

This demo shows you how to enable Case classification with AI and test it in a real-world scenario.

Consider a scenario that involves an insurance company, U+ Insurance. This company is currently handling a high volume of car insurance claims. The process they follow is traditional and manual; experts review all Cases and manually categorize and route them based on the description provided by customers. This process often results in delays and errors due to the sheer volume of Cases and the human element involved.

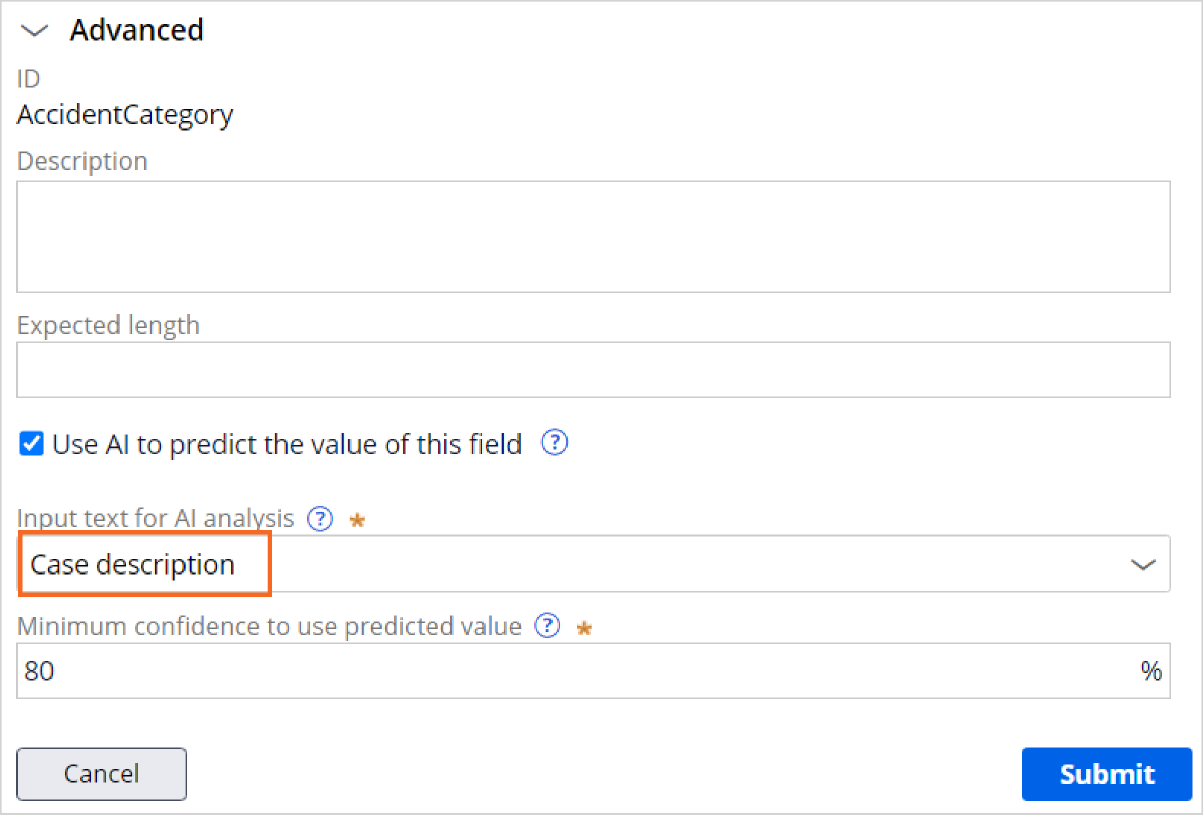
Recognizing the need for a more efficient and accurate process, the company decides to utilize natural language processing in Pega Process AI.

The goal is to automate the categorization and Case routing, which reduces the time it takes to process claims and minimizes the possibility of errors as a result.



An Application Developer must enable the AI-powered Case classification prediction. If the desired category field is nonexistent in the application, the developer begins by creating it in the Case and defining the topics. The system uses the choices set by the Application

Developer as topics for the new Accident Category Prediction. This is a crucial step as it sets the foundation for the AI to understand and categorize the Cases. Now, the Application Developer must enable AI and specify the Case description provided by the customer as the input text for AI analysis.



Advanced

ID
AccidentCategory

Description

Expected length

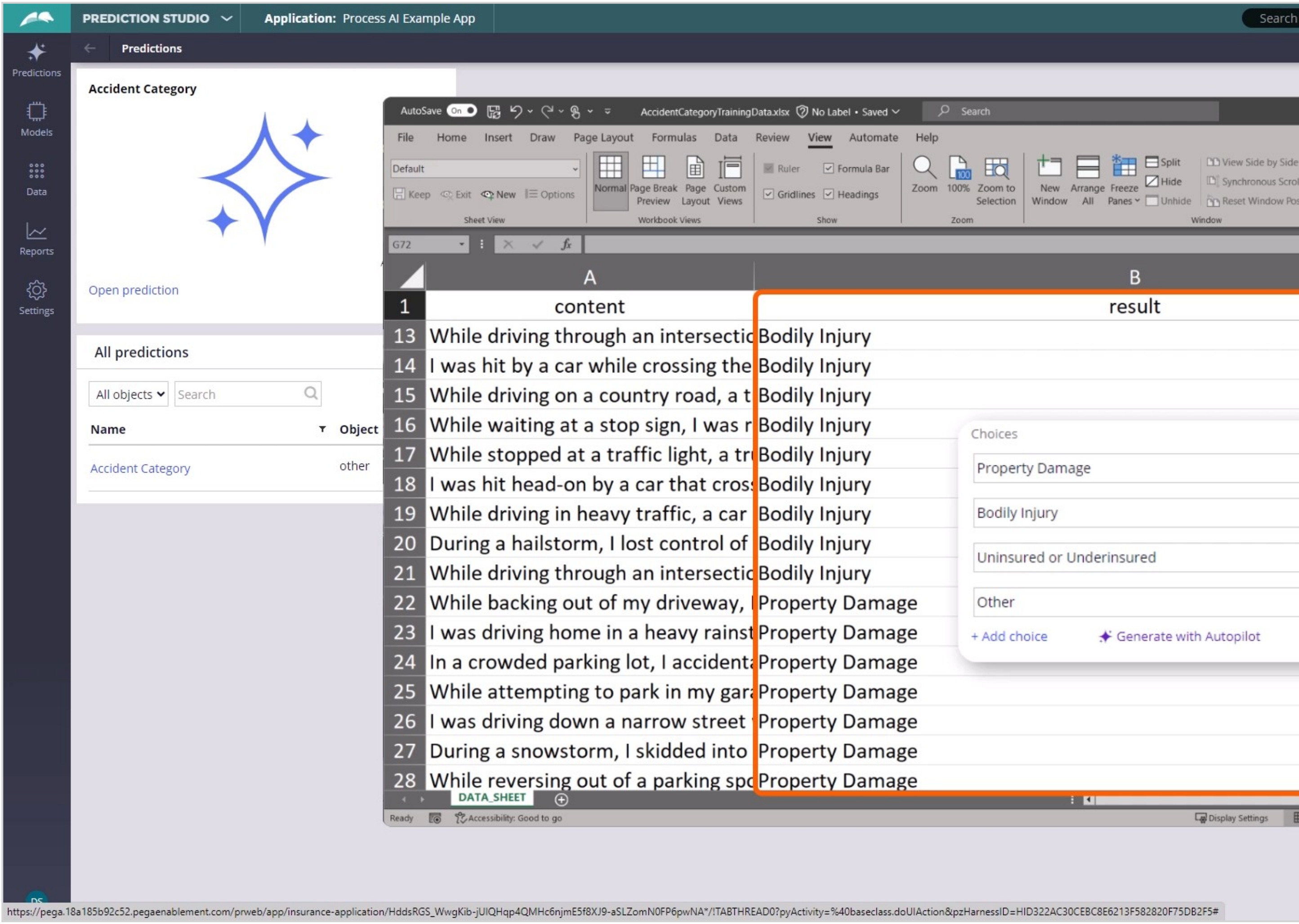
☑ Use AI to predict the value of this field

Input text for AI analysis *
Case description

Minimum confidence to use predicted value *
80 %

After the Application Developer enables AI, the system automatically creates a new Prediction in Prediction Studio. When insurance claims arrive, the Prediction uses AI models to recognize the topic of each claim and categorize it based on the Case description provided by the customer. The topics correspond with the choices set in the new Accident Category field.

At first, the models are not trained. Data Scientist trains and builds the models in Prediction Studio. Data Scientist uses a provided training data file, which contains examples of Case descriptions and their corresponding topics. The topics match accident categories that the prediction needs to detect.



	A	B
1	content	result
13	While driving through an intersectic	Bodily Injury
14	I was hit by a car while crossing the	Bodily Injury
15	While driving on a country road, a t	Bodily Injury
16	While waiting at a stop sign, I was r	Bodily Injury
17	While stopped at a traffic light, a tr	Bodily Injury
18	I was hit head-on by a car that cros	Bodily Injury
19	While driving in heavy traffic, a car	Bodily Injury
20	During a hailstorm, I lost control of	Bodily Injury
21	While driving through an intersectic	Bodily Injury
22	While backing out of my driveway, I	Property Damage
23	I was driving home in a heavy rainst	Property Damage
24	In a crowded parking lot, I accidenta	Property Damage
25	While attempting to park in my gar	Property Damage
26	I was driving down a narrow street	Property Damage
27	During a snowstorm, I skidded into	Property Damage
28	While reversing out of a parking spc	Property Damage

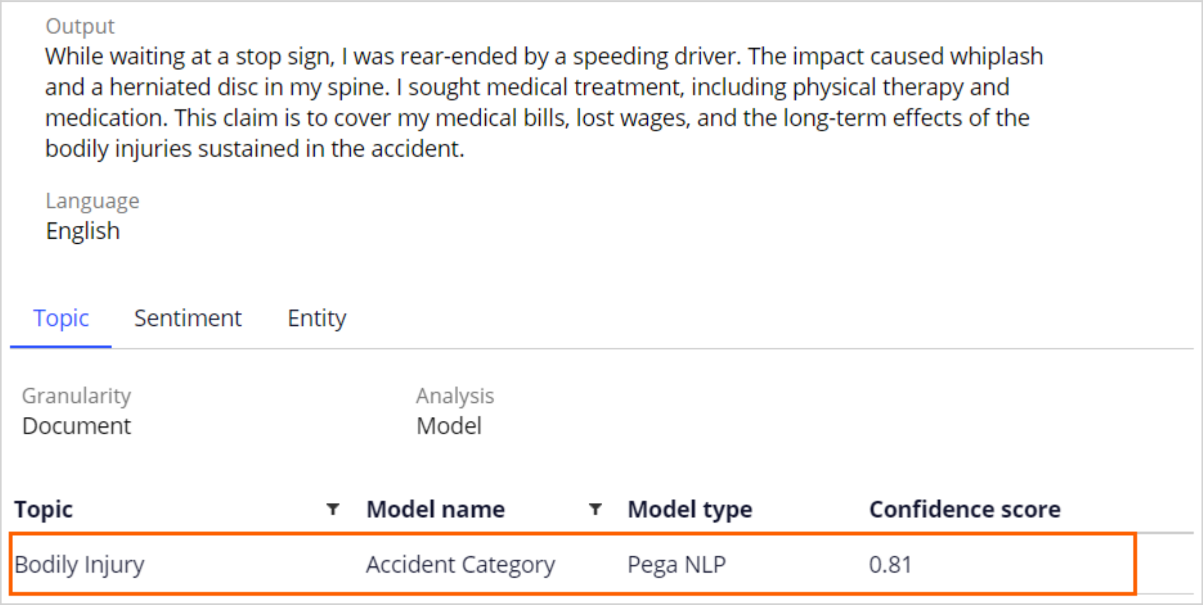
Choices

- Property Damage
- Bodily Injury
- Uninsured or Underinsured
- Other

+ Add choice
Generate with Autopilot

https://pega.18a185b92c52.pegaenablement.com/prweb/app/insurance-application/HddsRGS_WwgKib-jUIQHqp4QMHc6njmE5f8XJ9-aSLZomN0FP6pwNA*/!TABTHREAD0?pyActivity=%40baseclass.doUIAction&pzHarnessID=HID322AC30CEBC8E6213F582820F75DB2F5#

After the models train on the data, the text Prediction undergoes testing in Prediction Studio. The Data Scientist confirms that the AI correctly detects the topic of each message by providing the AI with sample Case descriptions and checking if the AI correctly identifies the accident category.



Output
While waiting at a stop sign, I was rear-ended by a speeding driver. The impact caused whiplash and a herniated disc in my spine. I sought medical treatment, including physical therapy and medication. This claim is to cover my medical bills, lost wages, and the long-term effects of the bodily injuries sustained in the accident.

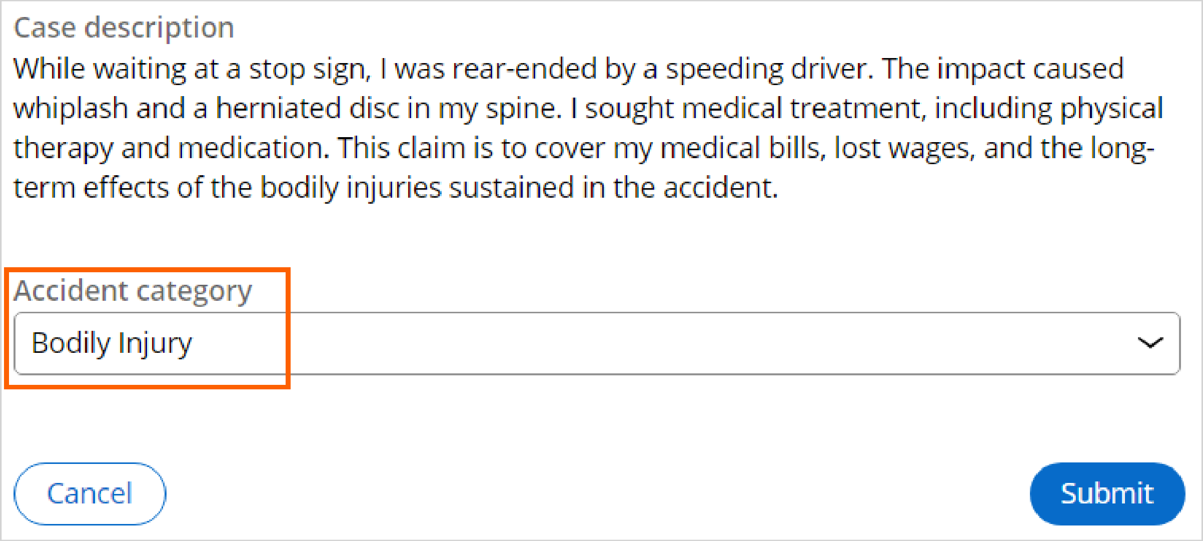
Language
English

Granularity
Document

Analysis
Model

Topic	Model name	Model type	Confidence score
Bodily Injury	Accident Category	Pega NLP	0.81

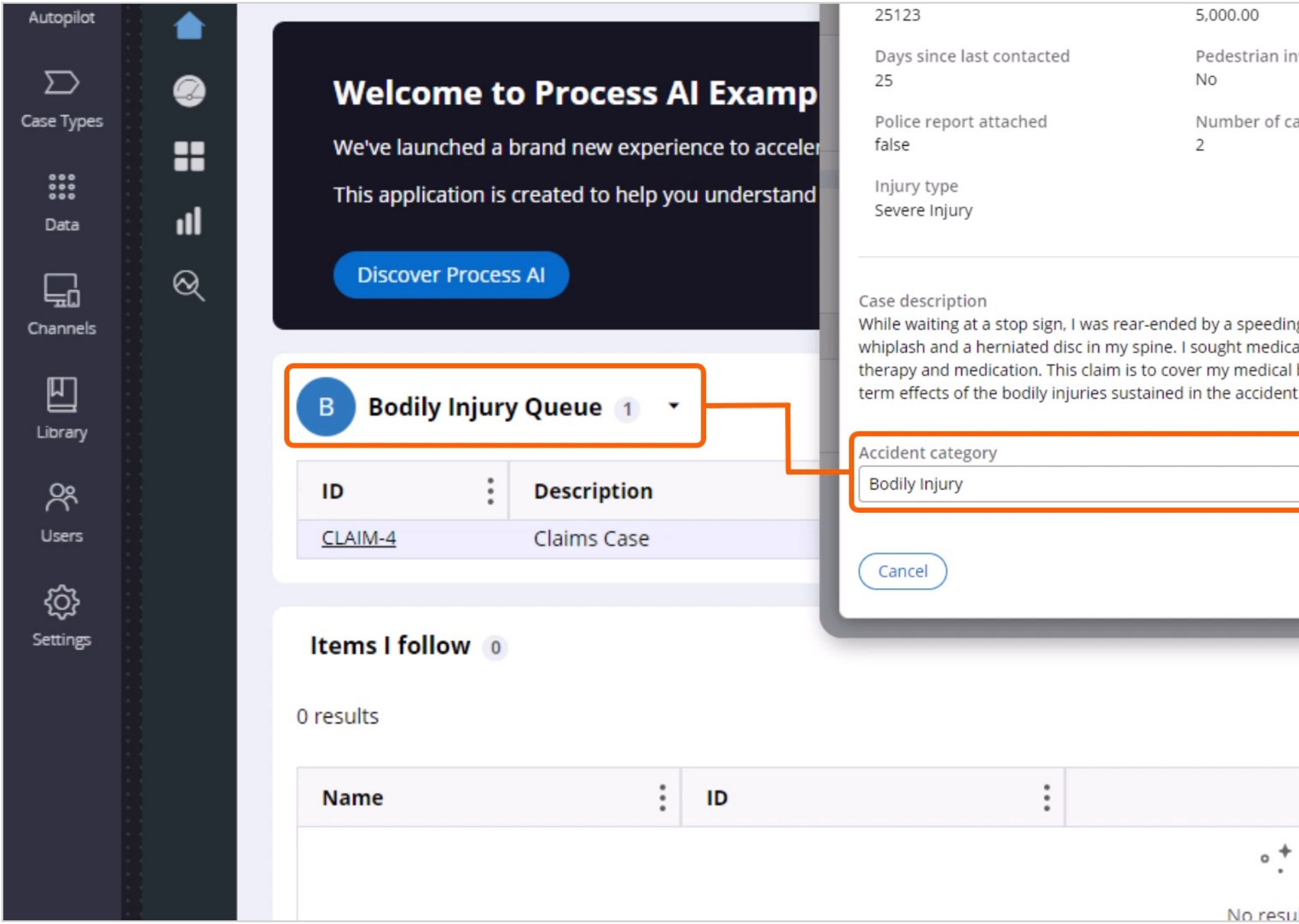
After the models complete training and testing, the Application Developer steps back in to validate the functionality of the automatic AI accident category prediction and Case routing in the Process AI Example Application. The developer creates a new Claims Case for each Persona and validates the detected category. This step helps ensure that the AI is accurately categorizing and routing the Cases in a real-world scenario.



Case description
While waiting at a stop sign, I was rear-ended by a speeding driver. The impact caused whiplash and a herniated disc in my spine. I sought medical treatment, including physical therapy and medication. This claim is to cover my medical bills, lost wages, and the long-term effects of the bodily injuries sustained in the accident.

Accident category
Bodily Injury

Finally, the Application Developer verifies the pre-configured intelligent Case routing. The system automatically routes the Cases to the correct Work Queue based on the predicted accident category. This action eliminates the need for manual classification and ensures that the appropriate teams handle Cases, which improves the efficiency of the claims processing workflow.



Welcome to Process AI Examp

We've launched a brand new experience to acceler

This application is created to help you understand

B **Bodily Injury Queue** 1

ID	Description
CLAIM-4	Claims Case

Items I follow 0

0 results

Name	ID

No resu

25123 | 5,000.00

Days since last contacted: 25

Pedestrian in: No

Police report attached: false

Number of ca: 2

Injury type: Severe Injury

Case description
While waiting at a stop sign, I was rear-ended by a speeding
whiplash and a herniated disc in my spine. I sought medica
therapy and medication. This claim is to cover my medical
term effects of the bodily injuries sustained in the accident

Accident category: Bodily Injury

This demo has concluded. What did it show you?

- How to enable AI Case classification in Pega Process AI by creating a new accident category field and defining topics.
- How to train and test models in Prediction Studio.
- How to validate the automatic AI accident category prediction and Case routing by creating a new Claims Case for a test Persona.
- How to verify the effectiveness of intelligent Case routing, which automatically routes Cases to the correct Work Queue based on the predicted accident category.

Predicting fraud

Description

Occasionally, an insurance claim might be erroneous or even fraudulent. To detect fraud and optimize the way in which the application routes work and meets business goals, learn how to use your own predictive models in case management.

Learning objectives

- Create a prediction to detect fraud
- Use the new prediction in a case type

Predicting fraud

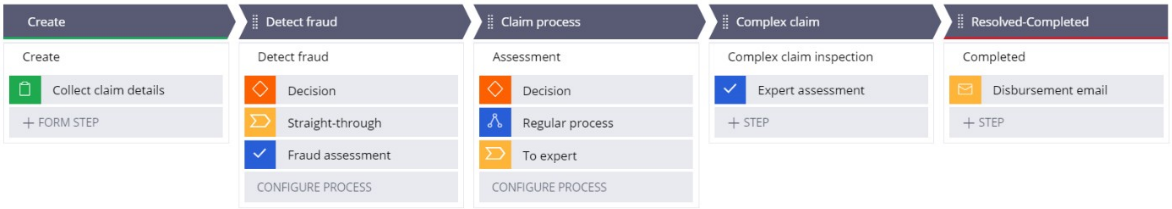
Introduction

Pega Process AI™ lets you bring your own predictive models to Pega. Use predictions in case types to optimize the way in which your application processes work and to meet your business goals. Learn how to use a predictive fraud model to effectively route suspicious claims for closer inspection.

Transcript

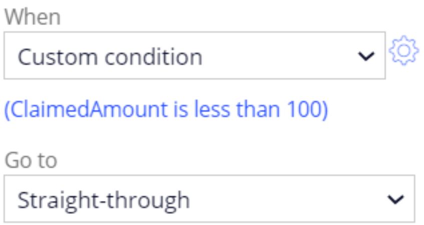
This demo will show you how to use a predictive fraud model in a case type to route suspicious claims to an expert.

U+ Insurance uses Pega Platform™ for case management. The life cycle of the case type that processes incoming car insurance claims contains a fraud detection stage, a regular process stage, and a complex claim process stage.



When the case is resolved, the claimant receives an email that communicates the decision.

The decision step in the **Detect fraud** stage routes cases with a low claimed amount for straight-through processing.



When

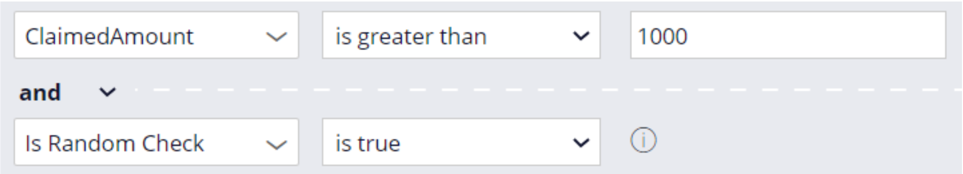
Custom condition

(ClaimedAmount is less than 100)

Go to

Straight-through

A set percentage of claims with a high claimed amount is routed to an expert for fraud assessment.

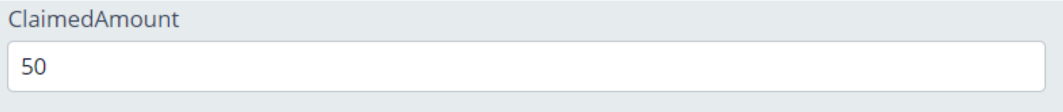


ClaimedAmount | is greater than | 1000

and

Is Random Check | is true

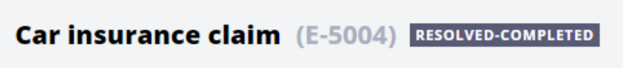
Consider this car insurance claim. The claimed amount is 50.



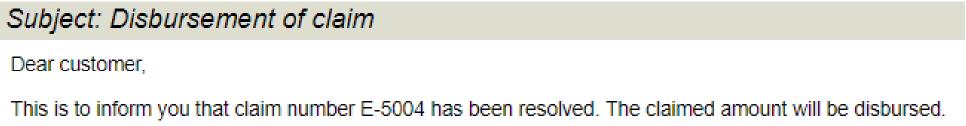
ClaimedAmount

50

The claim qualifies for straight-through processing as the claimed amount is below the threshold. The case is automatically resolved, and the claimant receives an email that states that the claimed amount will be disbursed.



Car insurance claim (E-5004) RESOLVED-COMPLETED

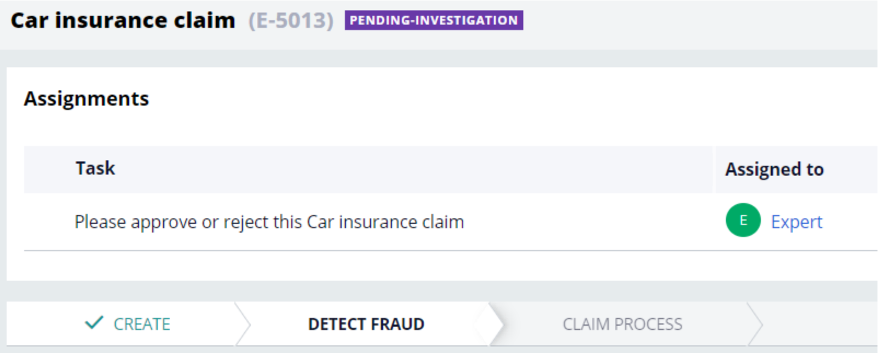


Subject: Disbursement of claim

Dear customer,

This is to inform you that claim number E-5004 has been resolved. The claimed amount will be disbursed.

A fraud expert inspects a set percentage of cases with a high claimed amount.

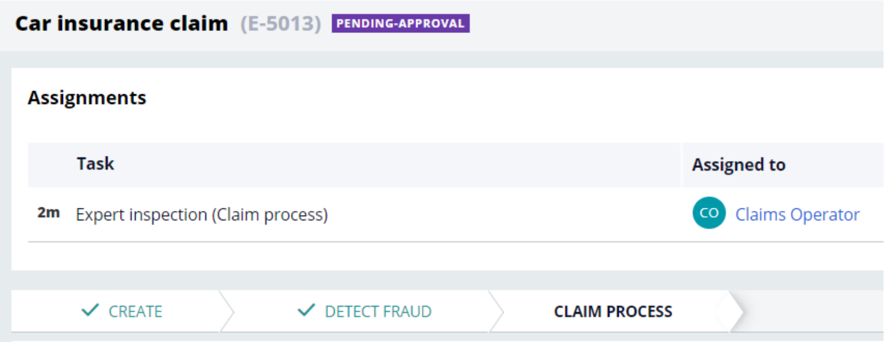


Car insurance claim (E-5013) PENDING-INVESTIGATION

Assignments

Task	Assigned to
Please approve or reject this Car insurance claim	E Expert

After approval, the system routes the case to the regular claim process.



Car insurance claim (E-5013) PENDING-APPROVAL

Assignments

	Task	Assigned to
2m	Expert inspection (Claim process)	CO Claims Operator

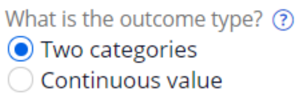
U+ Insurance wants to improve the effectiveness of fraud detection by using a predictive model that calculates the fraud risk of each claim.

The business requirements are that claims only qualify for straight-through processing if the fraud risk score is very low, while all claims with a high fraud risk score are inspected by the fraud expert. The routing of randomly selected cases to the fraud expert must remain in place to create a control group.

The data scientist team of U+ Insurance has developed a fraud model on the H2O.ai platform and has validated the model against historical data that the company captured.

The system qualifies a claim as abnormal if the probability of fraud exceeds the threshold; otherwise, the system classifies the case as normal.

To implement the fraud model, you create a new case management prediction. You can create a custom prediction that can forecast binary or numerical outcomes.

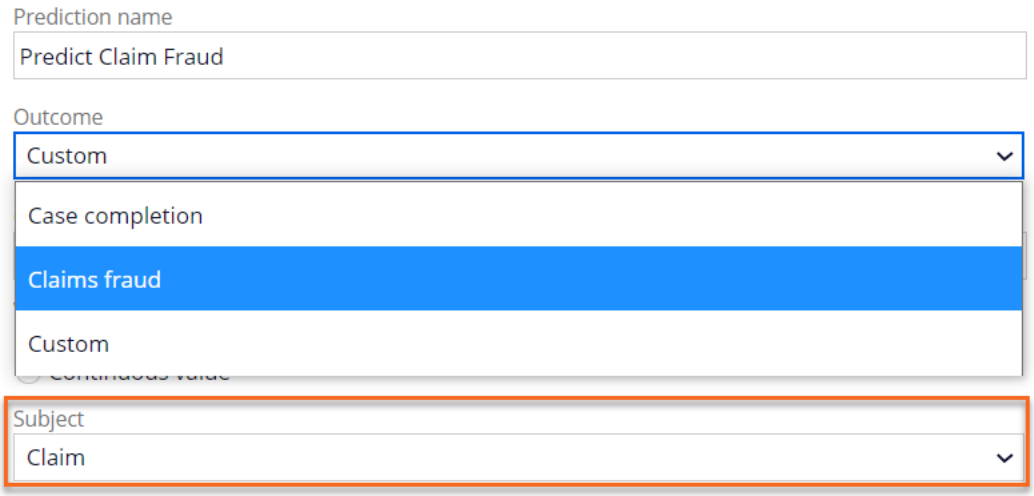


What is the outcome type?

● Two categories

○ Continuous value

For fraud detection, Process AI provides an out-of-the-box template. The claim is the subject of the prediction.



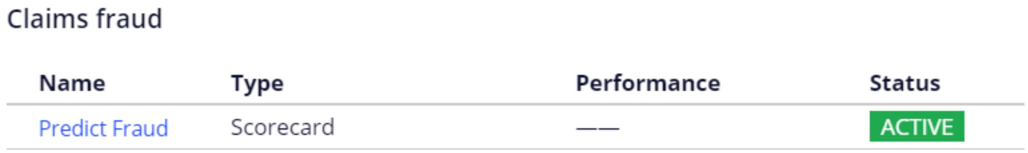
Prediction name: Predict Claim Fraud

Outcome: Custom

- Case completion
- Claims fraud
- Custom

Subject: Claim

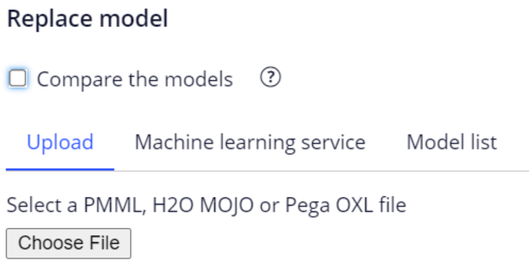
A placeholder scorecard initially drives the prediction.



Claims fraud

Name	Type	Performance	Status
Predict Fraud	Scorecard	——	ACTIVE

When the predictive fraud model replaces the scorecard, the prediction is ready for implementation in the Car insurance claim case type. You replace the placeholder with a machine learning model, a scorecard, or a field that contains a precalculated score. You can upload a machine learning model as a PMML or H2O file. Alternatively, you can connect to online machine learning services.

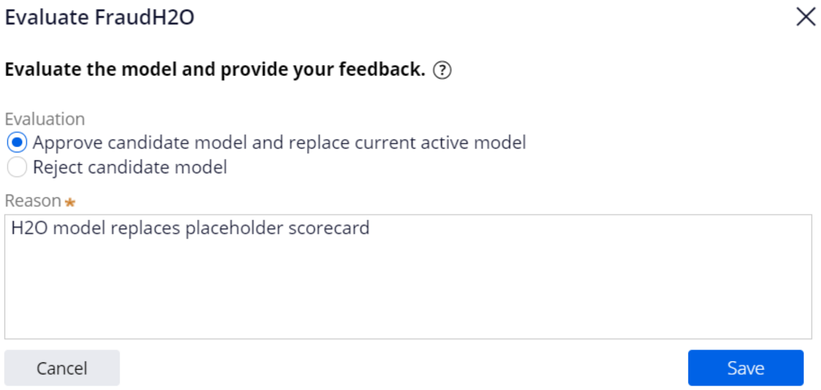


Replace model

□ Compare the models ⓘ

Select a PMML, H2O MOJO or Pega OXL file

You can select predictive models that are available in the application in the model list. When the model is ready for review, approve the model to replace the scorecard.



Evaluate FraudH2O

Evaluate the model and provide your feedback. ⓘ

Evaluation

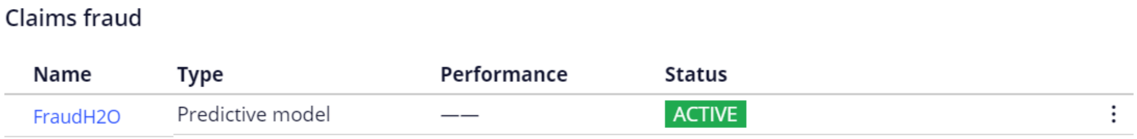
● Approve candidate model and replace current active model

○ Reject candidate model

Reason *

H2O model replaces placeholder scorecard

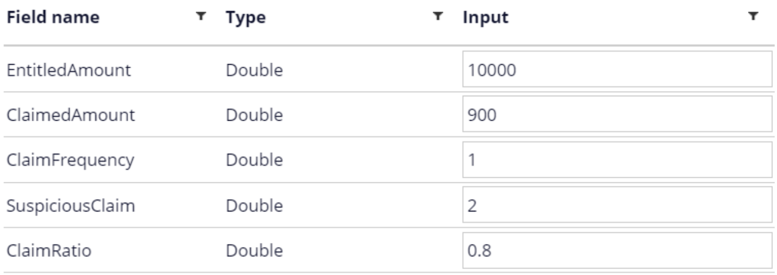
The fraud model now drives the prediction.



Claims fraud

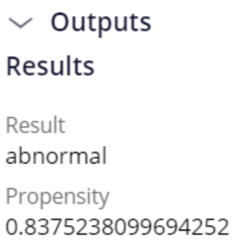
Name	Type	Performance	Status
FraudH2O	Predictive model	——	ACTIVE

When you run the model with these input values, the model qualifies the claim as abnormal.



Field name	Type	Input
EntitledAmount	Double	10000
ClaimedAmount	Double	900
ClaimFrequency	Double	1
SuspiciousClaim	Double	2
ClaimRatio	Double	0.8

The model predicts the claim to be abnormal because the propensity value is above the threshold.



Outputs

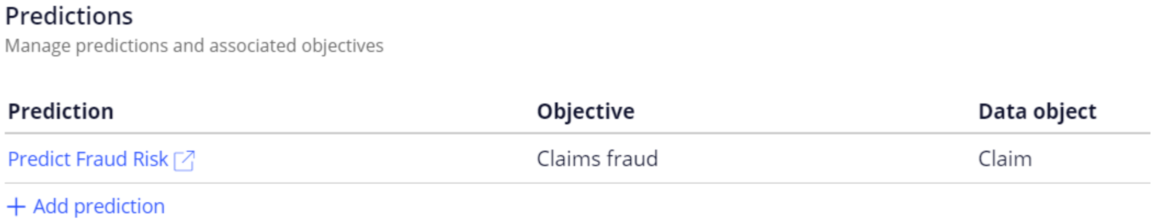
Results

Result
abnormal

Propensity
0.8375238099694252

Predictors of the model include the claim data, such as location and claimed amount, but can also cover customer behavior data, such as the number of recent claims.

As an application developer, you can implement the fraud prediction to route claims based on the fraud risk calculated by the model.

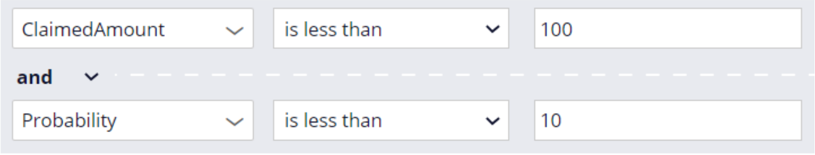


Predictions
Manage predictions and associated objectives

Prediction	Objective	Data object
Predict Fraud Risk	Claims fraud	Claim

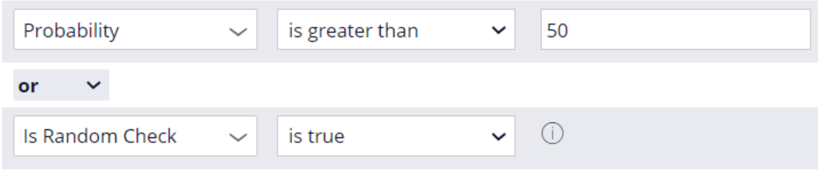
+ Add prediction

In the **Decision** step in the **Detect fraud** stage of the life cycle, implement the prediction. Add the condition that only claims with a very low predicted fraud risk qualify for straight-through processing.



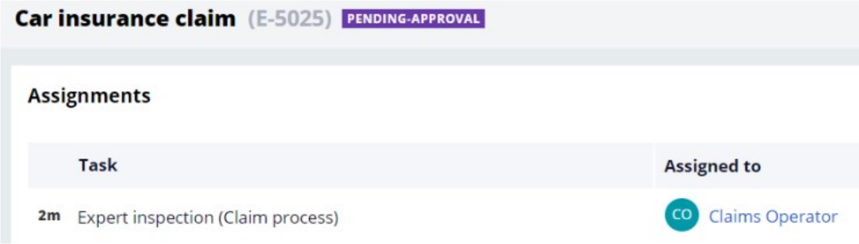
ClaimedAmount | is less than | 100
and
Probability | is less than | 10

Replace the condition that routes a claim to a fraud expert based on the claimed amount with a condition that is based on the outcome of the fraud model and change the logical operator to generate the control group.



Probability | is greater than | 50
or
Is Random Check | is true

When you run the same claim that previously qualified for straight-through the claim now disqualifies because the condition that fraud risk is very low is not met and the system consequently routes the case for regular processing.

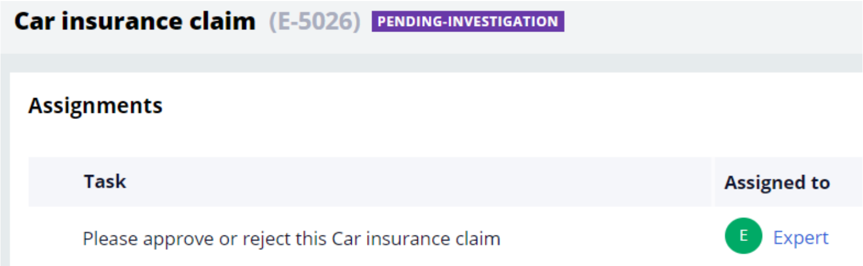


Car insurance claim (E-5025) PENDING-APPROVAL

Assignments

	Task	Assigned to
2m	Expert inspection (Claim process)	CO Claims Operator

When a claim with the same predictor values as previously tested in Prediction Studio is run, the system routes the case to the fraud expert.



Car insurance claim (E-5026) PENDING-INVESTIGATION

Assignments

Task	Assigned to
Please approve or reject this Car insurance claim	E Expert

This demo has concluded. What did it show you?

- How to create a case management prediction driven by a predictive model.
- How to use a prediction in a case type.

Predicting missing the Service-Level Agreement

Description

Pega Process AI™ can help to distinguish regular from complex claims. Complex claims often escalate into a lengthy process, which is not only costly, but also leads to poor customer experiences.

Learn how to use Process AI to create an adaptive model to route complex cases to an experienced handler and leave many of the claims for straight-through processing. As the adaptive model learns from the outcome of each case, it becomes more accurate at predicting which claims to escalate, and in that way to self-optimize the process.

Learning objectives

- Create a prediction that predicts case outcomes
- Use the new prediction to route complex cases to an expert

Predicting missing the Service-Level Agreement

Introduction

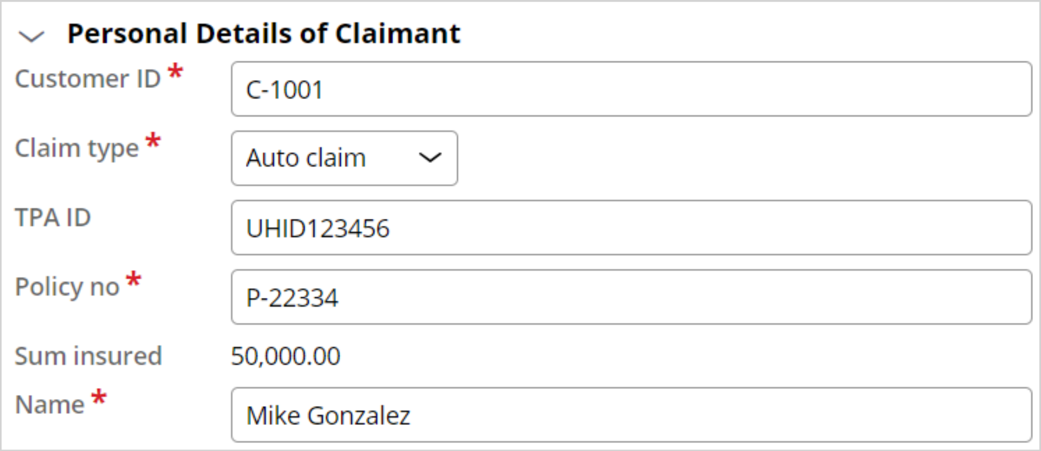
Pega Process AI™ can help to distinguish regular from complex claims. Complex claims often escalate into a lengthy process, which is costly and leads to a bad customer experience. The distinction lets you detect these claims early and address them at once.

Learn how to create a prediction that aims to identify cases that are likely to miss their deadlines and route them to a senior employee to handle them more efficiently and improve the customer experience.

Transcript

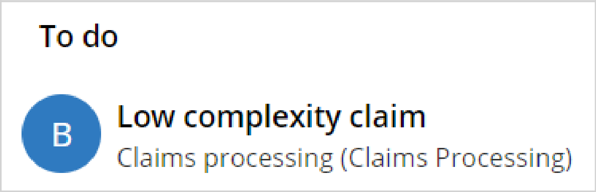
This demo shows you how to use adaptive models to predict missing the Service-Level Agreement (SLA).

U+ Insurance uses Pega Platform™ for case management. An incoming car insurance claim is straight through processed, or routed to a claims operator, who approves or rejects the claim to resolve the case.



Personal Details of Claimant

Customer ID *	C-1001
Claim type *	Auto claim
TPA ID	UHID123456
Policy no *	P-22334
Sum insured	50,000.00
Name *	Mike Gonzalez

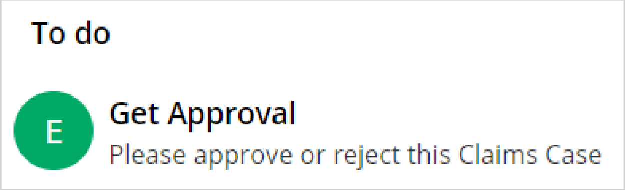


To do

B Low complexity claim

Claims processing (Claims Processing)

A case is escalated to an expert when the claim is not completed in the allotted time for regular processing.

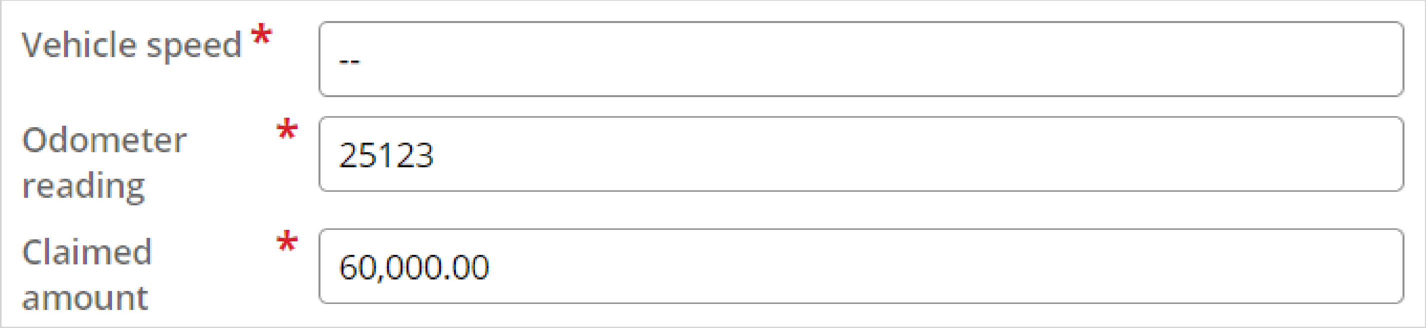


To do

E Get Approval

Please approve or reject this Claims Case

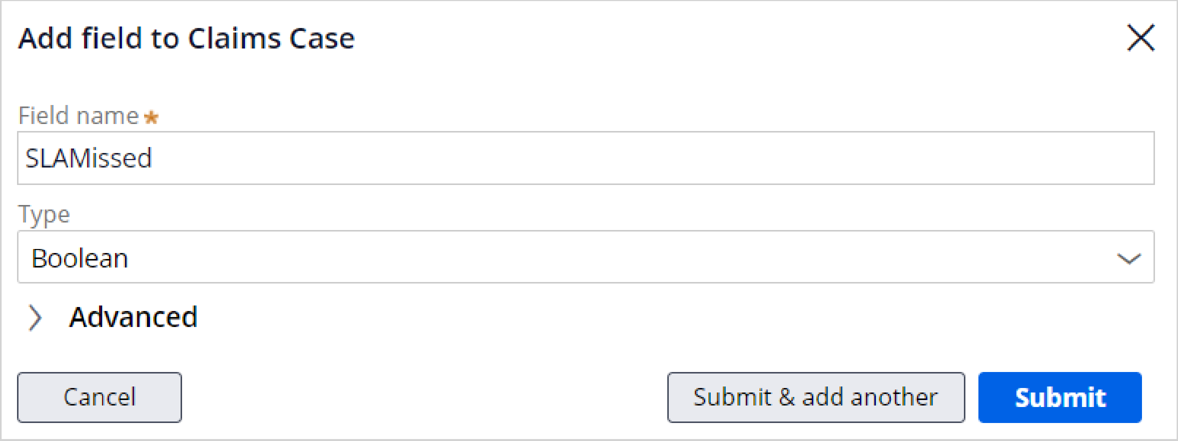
In the current configuration, claims that exceed 45000 are considered highly complex and are always investigated by an expert as a precaution.



Vehicle speed *	--
Odometer reading *	25123
Claimed amount *	60,000.00

However, decisioning using hard business rules, like in this case based on a simple cutoff value is not efficient, because even cases that exceed 45000 can be often resolved on time in the regular claims process. As a result, the experts consequently spend valuable time on relatively simple claims.

Process AI can help optimize the process by predicting the likelihood that a case is resolved before the deadline in the regular workflow and otherwise, route it to an expert irrespective of the cause of the complexity of the claim. First, it is an Application Developer's task to create a Boolean outcome field. It serves the adaptive model as the outcome field and allows it to distinguish cases that missed the SLA. You add the outcome field in the case type data model settings to make it available in that case type.



Add field to Claims Case

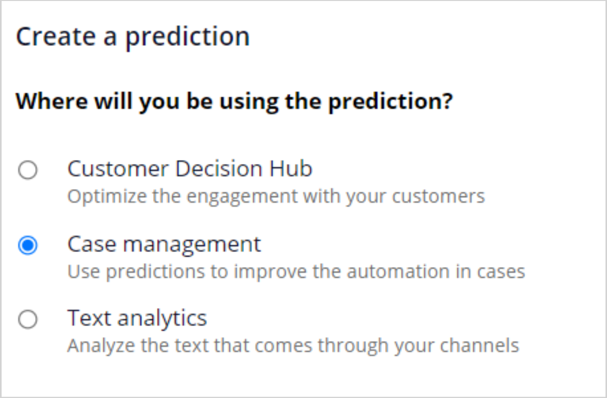
Field name *

SLAMissed

Type

Boolean

Next, to allow the model to learn from future outcomes, in Goal & deadline settings of the case type, the Application Developer configures a condition that automatically sets the outcome as missed when the deadline expires. Finally, a Data Scientist creates a case management prediction that calculates the propensity of whether the case is likely to miss the SLA.



Create a prediction

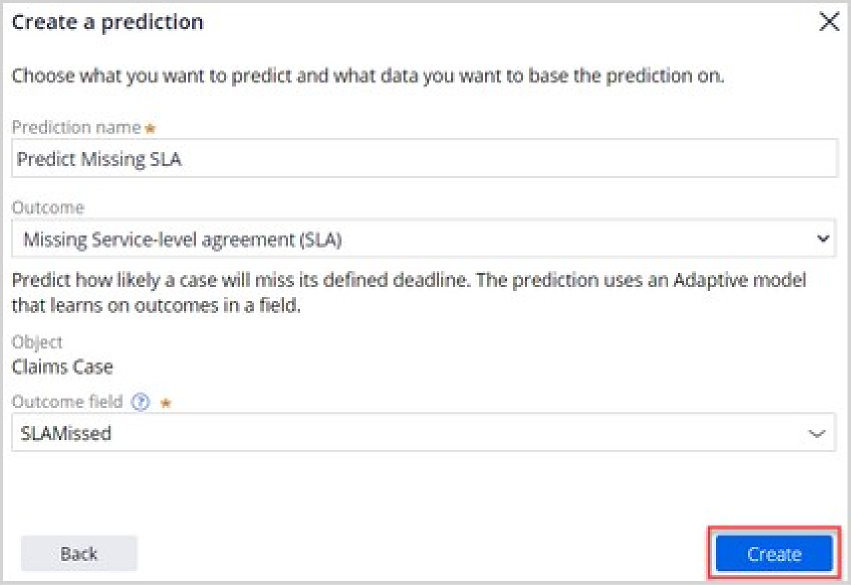
Where will you be using the prediction?

○ Customer Decision Hub
Optimize the engagement with your customers

● Case management
Use predictions to improve the automation in cases

○ Text analytics
Analyze the text that comes through your channels

Process AI offers a wizard to create Missing Service-Level Agreement (SLA) predictions.



Create a prediction

Choose what you want to predict and what data you want to base the prediction on.

Prediction name *
Predict Missing SLA

Outcome
Missing Service-level agreement (SLA)

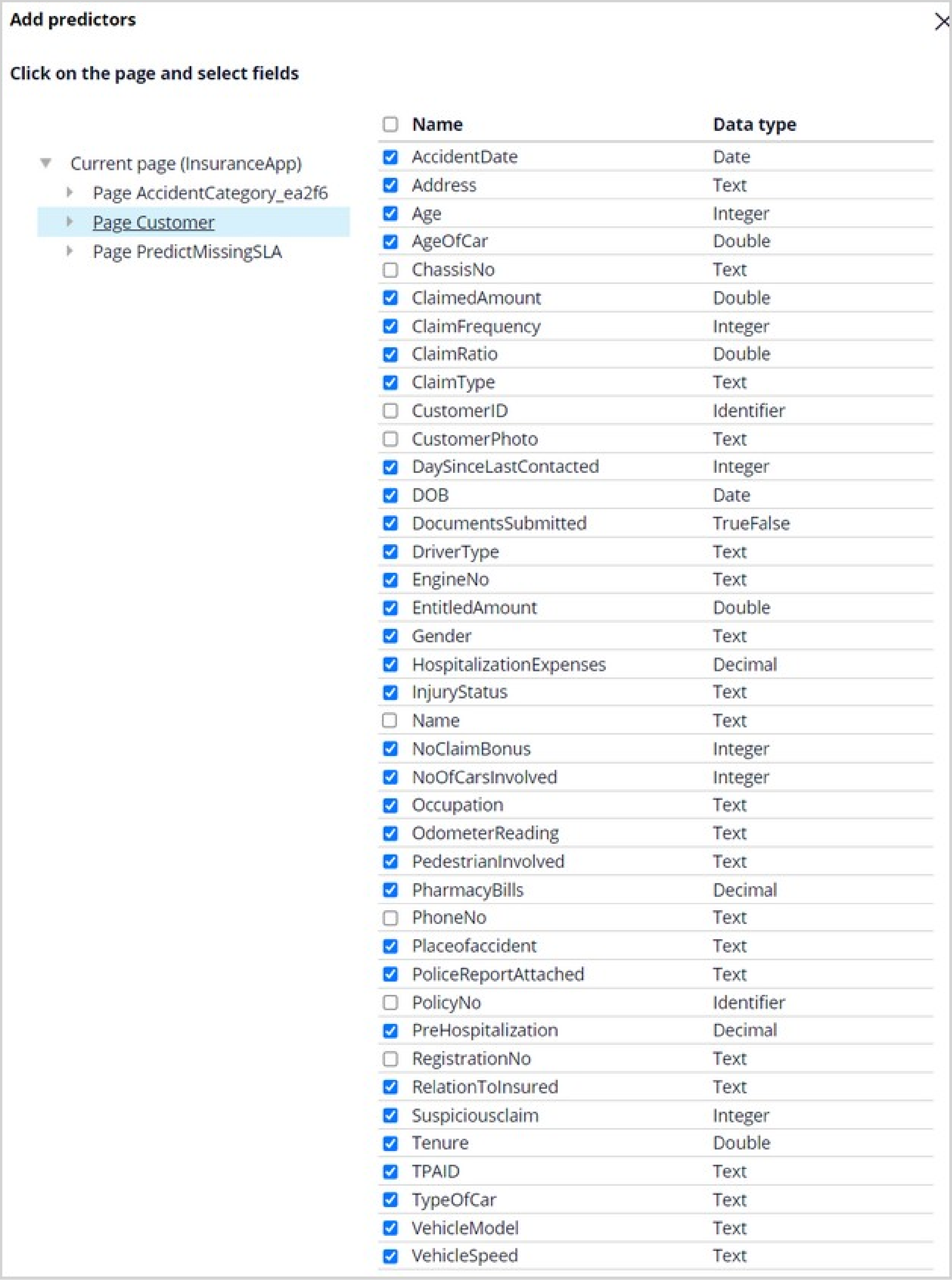
Predict how likely a case will miss its defined deadline. The prediction uses an Adaptive model that learns on outcomes in a field.

Object
Claims Case

Outcome field *
SLAMissed

The Outcome field reflects the Boolean field that the Application Developer creates. This associates the prediction with the case type.

Next step for the Data Scientist is to add potential predictors. Best practice is to include many unrelated fields, including the claim properties. It is also important to exclude predictors that are irrelevant and do not have any predicting power, like **ChassisNo**, **CustomerID**, **CustomerPhoto**, **Name**, **PhoneNo**, **PolicyNo**, and **RegistrationNo**.



Add predictors

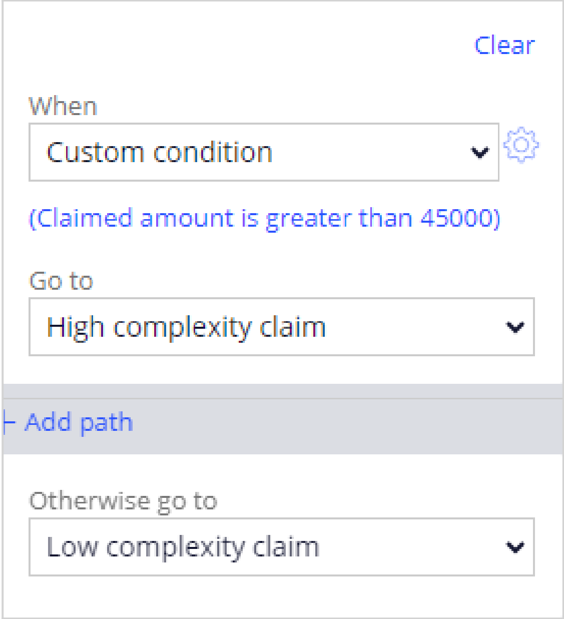
Click on the page and select fields

- Current page (InsuranceApp)
  - Page AccidentCategory_ea2f6
  - Page Customer
  - Page PredictMissingSLA

□	**Name**	**Data type**
☑	AccidentDate	Date
☑	Address	Text
☑	Age	Integer
☑	AgeOfCar	Double
□	ChassisNo	Text
☑	ClaimedAmount	Double
☑	ClaimFrequency	Integer
☑	ClaimRatio	Double
☑	ClaimType	Text
□	CustomerID	Identifier
□	CustomerPhoto	Text
☑	DaySinceLastContacted	Integer
☑	DOB	Date
☑	DocumentsSubmitted	TrueFalse
☑	DriverType	Text
☑	EngineNo	Text
☑	EntitledAmount	Double
☑	Gender	Text
☑	HospitalizationExpenses	Decimal
☑	InjuryStatus	Text
□	Name	Text
☑	NoClaimBonus	Integer
☑	NoOfCarsInvolved	Integer
☑	Occupation	Text
☑	OdometerReading	Text
☑	PedestrianInvolved	Text
☑	PharmacyBills	Decimal
□	PhoneNo	Text
☑	Placeofaccident	Text
☑	PoliceReportAttached	Text
□	PolicyNo	Identifier
☑	PreHospitalization	Decimal
□	RegistrationNo	Text
☑	RelationToInsured	Text
☑	Suspiciousclaim	Integer
☑	Tenure	Double
☑	TPAID	Text
☑	TypeOfCar	Text
☑	VehicleModel	Text
☑	VehicleSpeed	Text

The adaptive model learns from previous cases and automatically activates predictors that perform above a threshold and deactivates predictors when their performance drops over time. The prediction is ready to be implemented in the Claims case case type by an application developer.

In the current configuration, the Decide complexity decision step categorizes claims as low or high complexity depending only on the claimed amount. As a result, claims that exceed 45000 are categorized as complex.

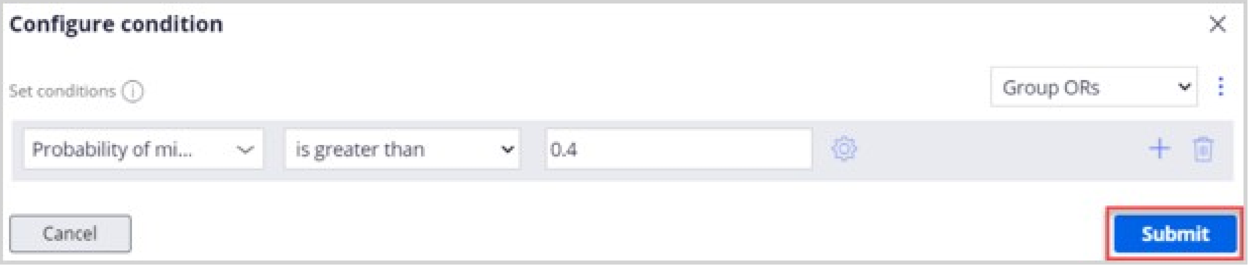


When
Custom condition
(Claimed amount is greater than 45000)

Go to
High complexity claim

Otherwise go to
Low complexity claim

This condition requires an edit to meet the new business requirement that the routing decision is based on the propensity calculated by the Missing SLA prediction. To categorize a claim as a high complexity claim in the **Decide complexity** decision step, the propensity to miss the SLA needs to exceed a threshold. In this case: 0.4.



Configure condition

Set conditions

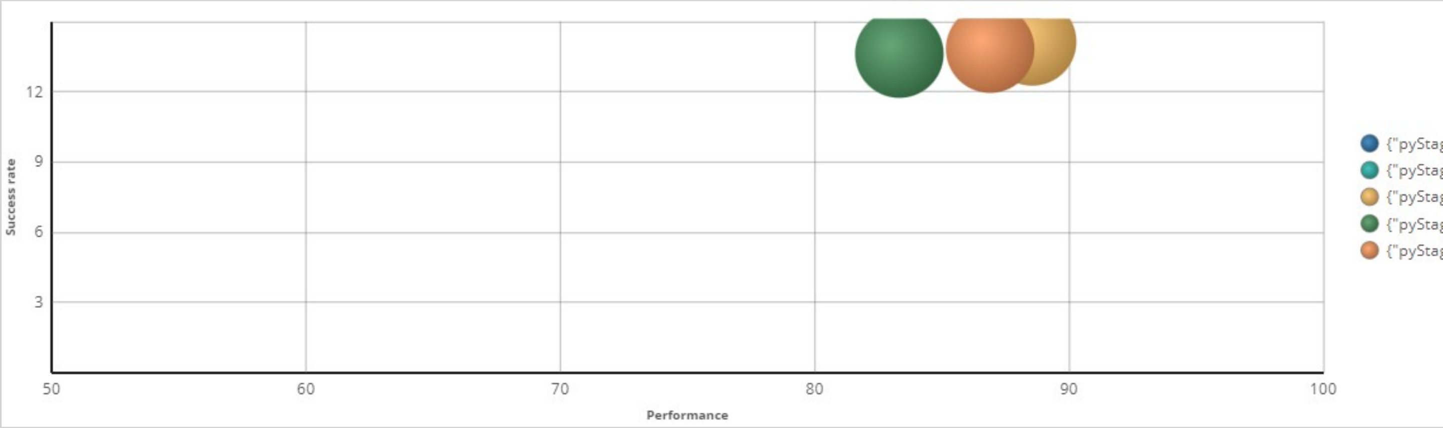
Group ORs

Probability of mi... | is greater than | 0.4

When a claims operator handles a claim, the case status changes to **Resolved-Completed** or **Resolved-Rejected**, and the outcome of the case maps to the alternative label (MissedSLA = False) for the prediction. When a complex claim misses the deadline, the outcome of the case maps to the target label (MissedSLA = True) for the prediction. The model learns using this information and as a result, depending on the outcome, the missing SLA propensity for a similar case in the future increases or decreases.

A claim with a high propensity to miss SLA is immediately routed to an expert. The claim is routed to the regular workflow when the expert assesses the claim and does not consider it a complex case. This reassignment allows the adaptive model to learn from cases that are incorrectly routed to the expert.

An adaptive model is created for each primary and alternative stage in the case type. A decision request in a stage uses the model that is specific to that stage to calculate the propensity.



At the very beginning, the models have no predictive power. The models learn and self-optimize with every captured case outcome.

This demo has concluded. What did it show you?

- How to create a missing SLA prediction.
- How to implement a missing SLA prediction to improve efficiency.

Pega NLP overview

Description

Gain a greater understanding of the key features, capabilities, and benefits of Pega natural language processing in decision management context. Businesses are exploring ways to utilize language processing. With Pega NLP you can analyze and extract meaningful information from a large amount of text by using text analytics to improve business performance and customer experience.

Learning objectives

- Explain text analytics in Pega NLP context.
- Describe practical use cases where text analytics can be used.

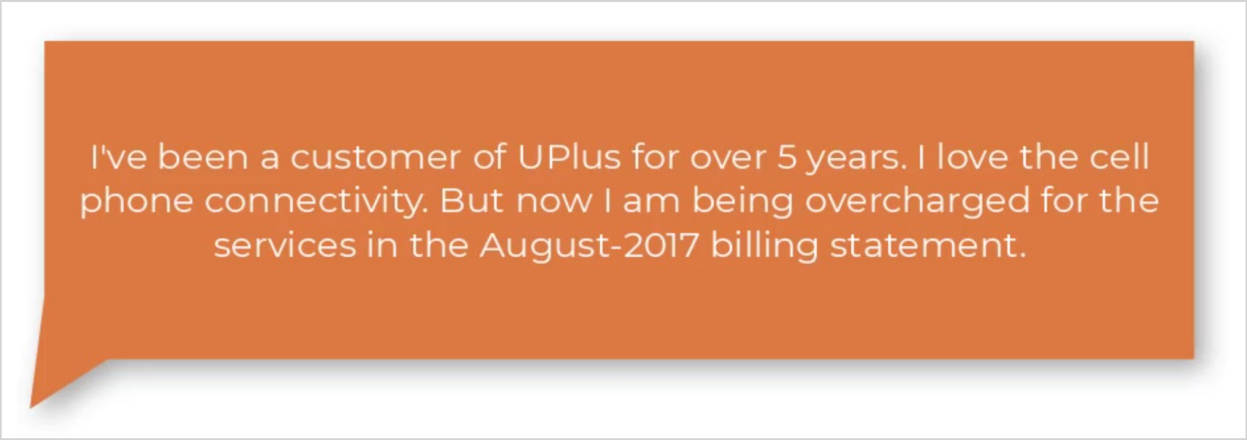
Text analytics

Today, digital sources such as news, blogs, social media, emails, and chatbots generate continuous text. Text analytics helps to transform large amounts of text into a structured format so that your applications can use the text to optimize processes such as email routing, automatic chatbot interactions, or entity extraction.

Transcript

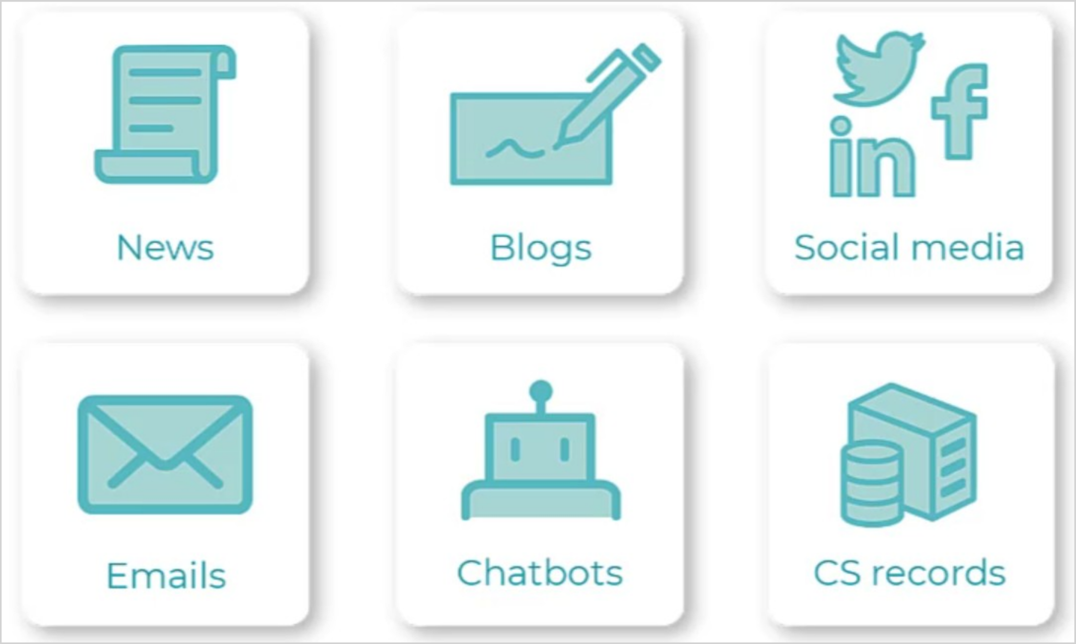
Learn about text analytics and how it is helps determine customer sentiments and needs.

Consider this complaint from a customer to a customer service department.



I've been a customer of UPlus for over 5 years. I love the cell phone connectivity. But now I am being overcharged for the services in the August-2017 billing statement.

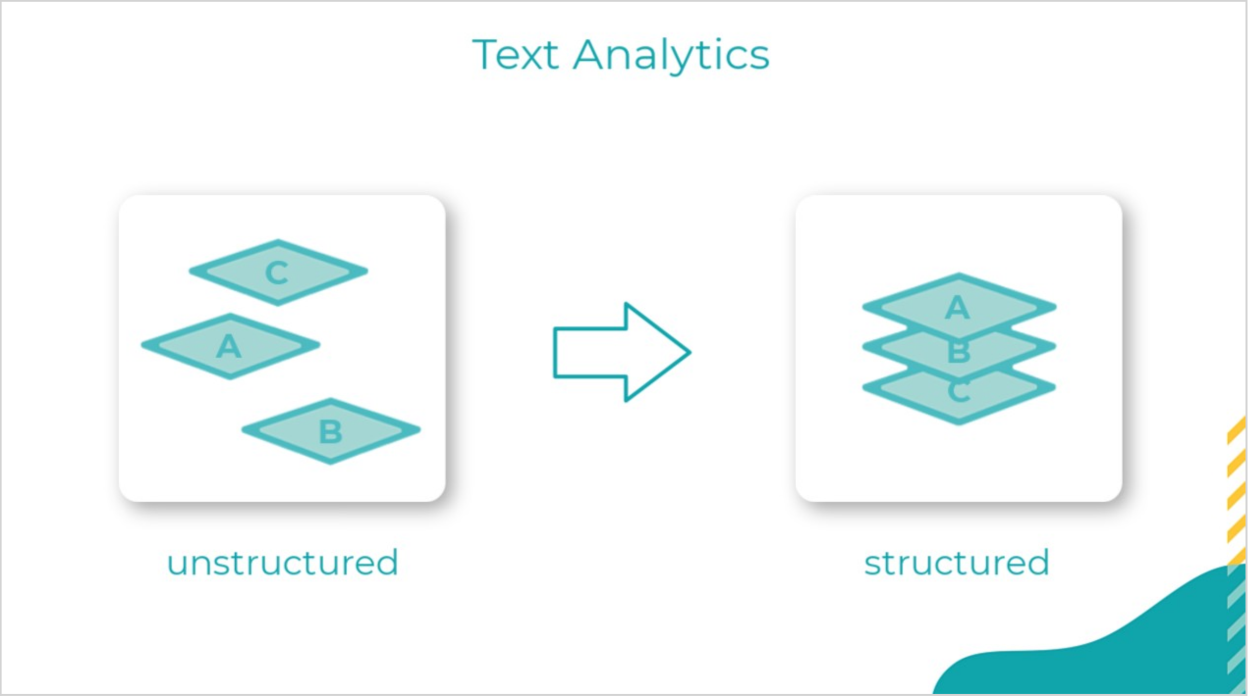
Humans can understand the implications of a piece of text by summarizing, elaborating, or even rephrasing it while preserving its original meaning. The challenge for humans comes when text exists on a large scale.



Consider a company with tens of thousands of customer service records in text form.

The company wants to route all complaints automatically to the customer service department. Then, it wants a report to understand whether customers are satisfied with various products and services. The company must find a way to process and extract the text, and text analytics is the most efficient and effective way to accomplish this task.

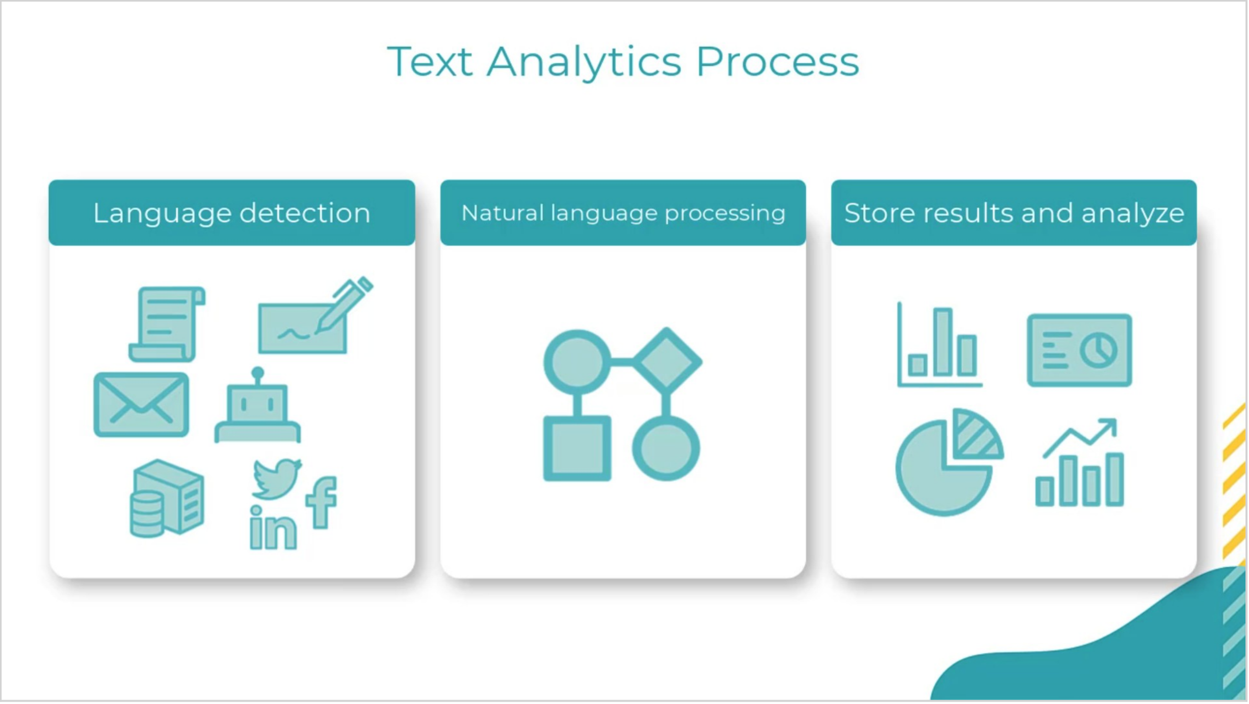
Text analytics is the process of deriving valuable information from text.



Text Analytics

It involves converting large amounts of text into structured data that can be analyzed using statistical methods. There are three main steps involved in text analytics:

1. Language detection
2. Natural language processing
3. Store results and analyze



Text Analytics Process

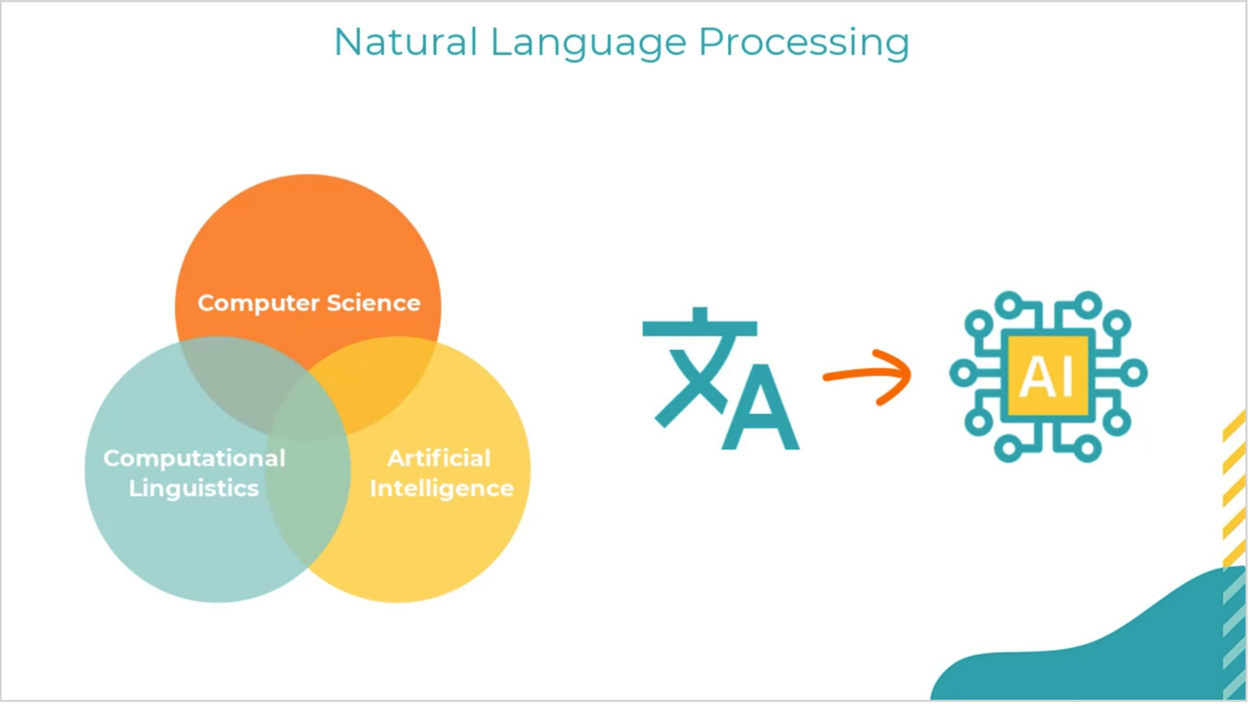
Language detection

Natural language processing

Store results and analyze

First, the system collects data from a source, such as social media, emails, chatbots, or customer service records. Second, natural language processing techniques are applied to extract specific attributes from the text and present them as structured data. And third, the extracted structured data is stored and analyzed.

Natural language processing is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human language.



Natural Language Processing

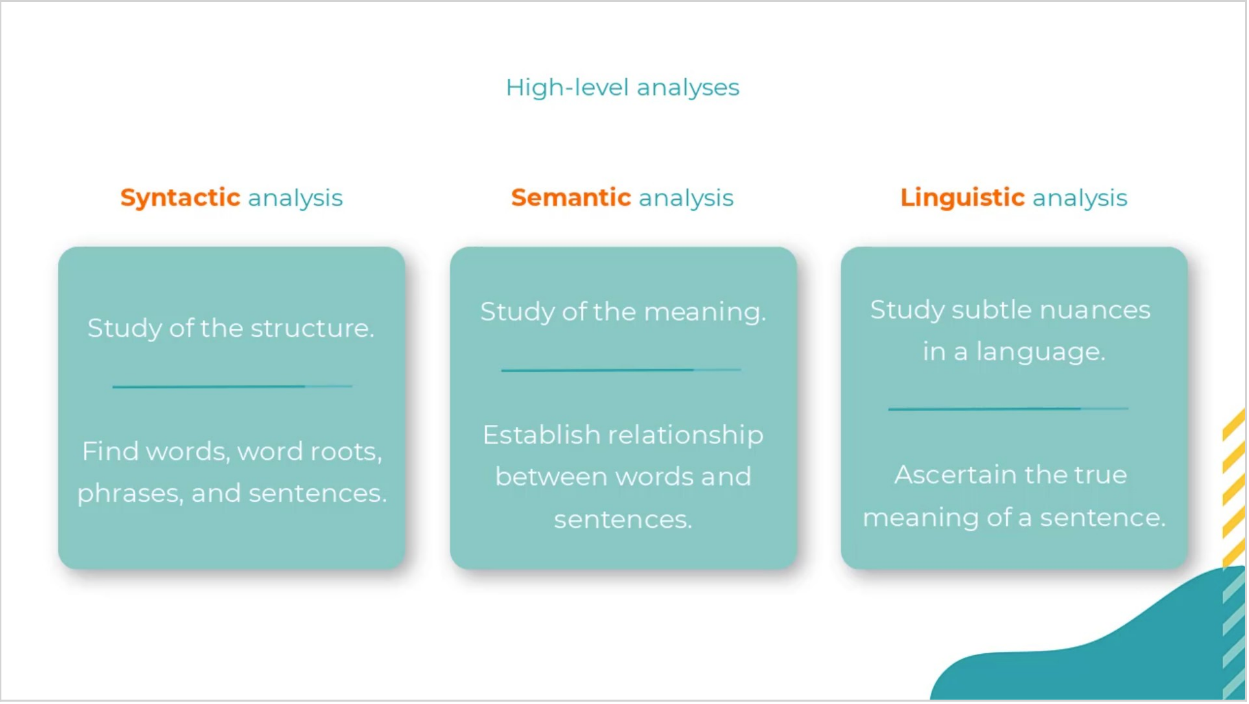
Computer Science

Computational Linguistics

Artificial Intelligence

In particular, it is concerned with programming computers to process large volumes of natural language text fruitfully. At a high level, the following analyses occur during natural language processing.

Syntactic analysis analyzes the structure of text and recognizes, for example, nouns, verbs, and adjectives. Semantic analysis establishes the meaning of a piece of text and the relationship between words and sentences. The linguistic analysis phase determines the exact meaning of the text.



High-level analyses

Syntactic analysis

Study of the structure.

Find words, word roots, phrases, and sentences.

Semantic analysis

Study of the meaning.

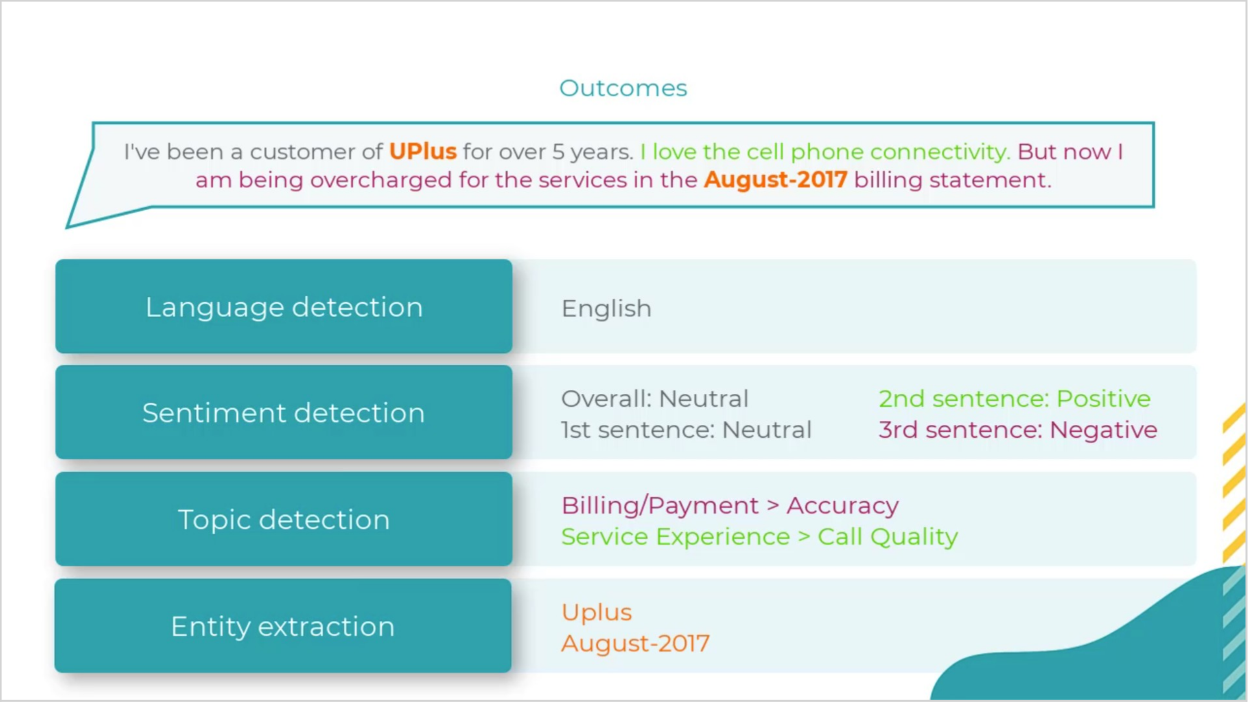
Establish relationship between words and sentences.

Linguistic analysis

Study subtle nuances in a language.

Ascertain the true meaning of a sentence.

The processing of the text from the customer who claims that they were overcharged produces a series of outcomes. First, the analysis detects the language because this action is a prerequisite for the next steps.



Outcomes

I've been a customer of **UPlus** for over 5 years. I love the cell phone connectivity. But now I am being overcharged for the services in the **August-2017** billing statement.

Language detection	English
Sentiment detection	Overall: Neutral 1st sentence: Neutral 2nd sentence: Positive 3rd sentence: Negative
Topic detection	Billing/Payment > Accuracy Service Experience > Call Quality
Entity extraction	Uplus August-2017

Then, the analysis detects the sentiment of the text as positive, negative, or neutral. The sentiment is the general attitude expressed towards a subject by the author of the message. The analysis reports the overall sentiment and sentiment for each sentence.

The analysis classifies the text into one or more predefined topics, such as the business functions of a company. It also reports the sentiment of the corresponding sentence for each category that it detects. Entities refer to the proper nouns found in the text (such as names of people, places, dates and times, and organizations), which helps establish the subject of the text.

You have reached the end of this video. You have learned:

- What text analytics is.
- How text analytics process works.
- What natural language processing is and how it works in text analytics.

Pega NLP overview

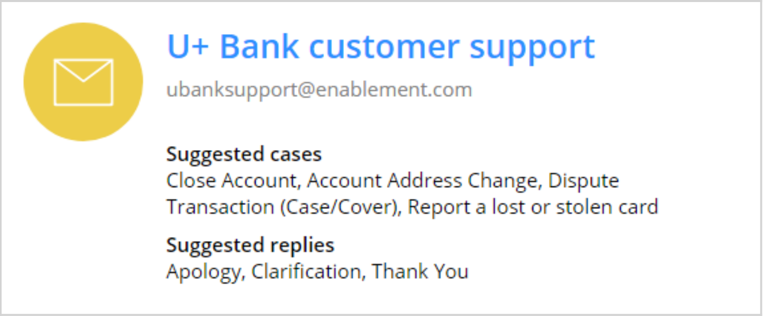
Introduction

Pega natural language processing (NLP) focuses on text categorization and text extraction. Discover how the system uses Pega NLP to route incoming emails based on the topic of the email, and how NLP drives a chatbot to automatically detect the topic of the message, extract necessary entities, create a case, and then route it to the correct customer service workbasket.

Transcript

This video introduces you to Pega NLP, a feature of the decision management capability of Pega Platform. You can apply decision management to any application that uses Pega Platform, which makes it highly versatile.

Pega NLP uses text analytics predictions to detect the topic and entities that are included in the message. For example, a company has an email channel where its customers can send any email.



U+ Bank customer support

ubanksupport@enablement.com

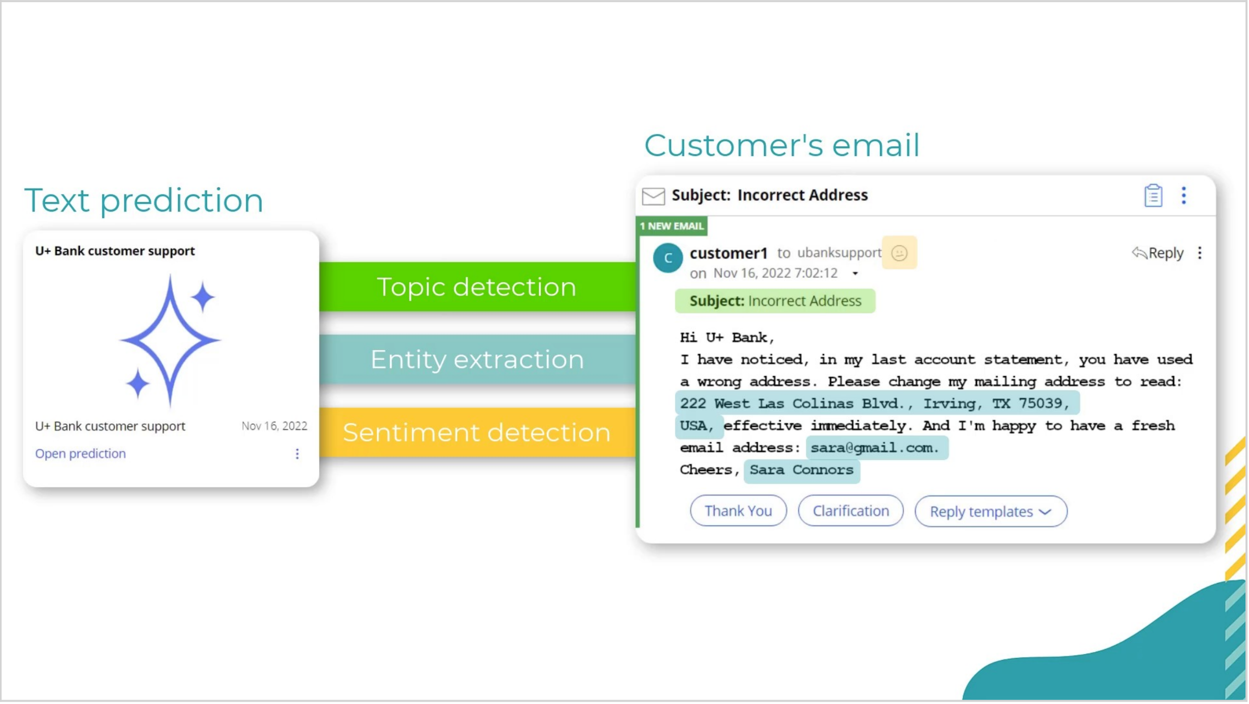
Suggested cases
Close Account, Account Address Change, Dispute Transaction (Case/Cover), Report a lost or stolen card

Suggested replies
Apology, Clarification, Thank You

A text prediction drives the email channel. Consider this email sent by the customer through the company's email channel. The text prediction aims to detect the topic of the email and route it to the appropriate department.

Apart from the topic detection model, the prediction also includes an entity extraction model that can extract entities from an email. It can automatically detect and extract certain entities such as an account number, email address, or street address. This functionality allows to automatically process or prioritize emails.

In addition to topic detection and entity extraction, Pega NLP detects the sentiment of an email based on its content. The sentiment score ranges from -1 to 1 and is negative, positive, or neutral.



Subject: Incorrect Address

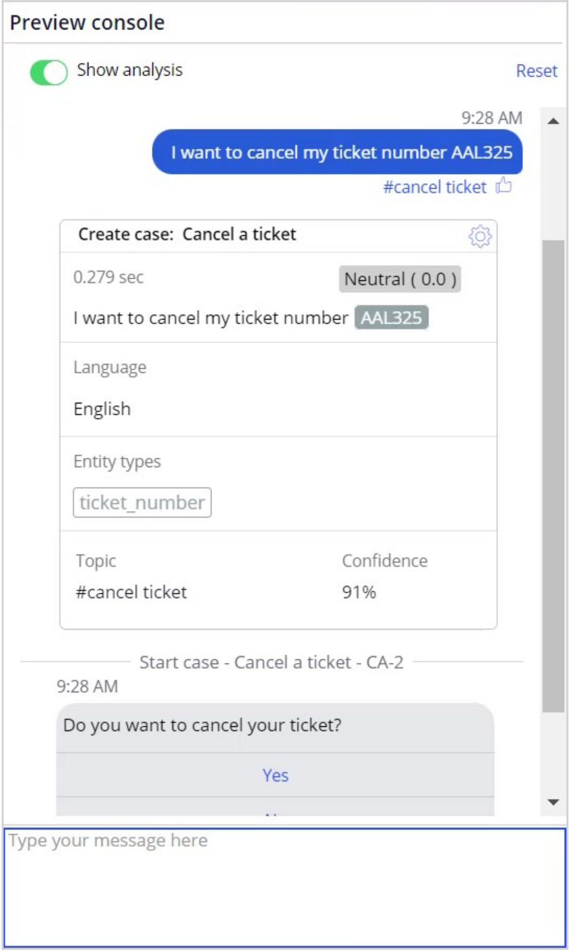
```
Hi U+ Bank,
I have noticed, in my last account statement, you have used
a wrong address. Please change my mailing address to read:
222 West Las Colinas Blvd., Irving, TX 75039,
USA, effective immediately. And I'm happy to have a fresh
email address: sara@gmail.com.
Cheers, Sara Connors
```

This functionality offloads work from customer service representatives, speeds up the process, and improves the customer experience.

Multiple channels can use the Pega NLP feature. For example, an airline sets up a chatbot channel to speed up customer service by automating the ticket cancellation process.

When Troy, a customer, sends a message to the Airline chatbot that he wants to cancel his ticket, the text prediction detects the topic of the message and runs the `Cancel a ticket` case type that an application developer preconfigured in the system. The case type contains the conversation flow that the chatbot uses in conversation with Troy. The chatbot uses entity extraction models to detect entities such as the ticket number, includes all necessary information in a case, and routes it to a correct customer service workbasket.

In App Studio, you can test the chatbot in the preview console.



Preview console

Show analysis

Reset

9:28 AM

I want to cancel my ticket number AAL325

#cancel ticket

Create case: Cancel a ticket

0.279 sec

Neutral (0.0)

I want to cancel my ticket number AAL325

Language

English

Entity types

ticket_number

Topic	Confidence
#cancel ticket	91%

Start case - Cancel a ticket - CA-2

9:28 AM

Do you want to cancel your ticket?

Yes

Type your message here

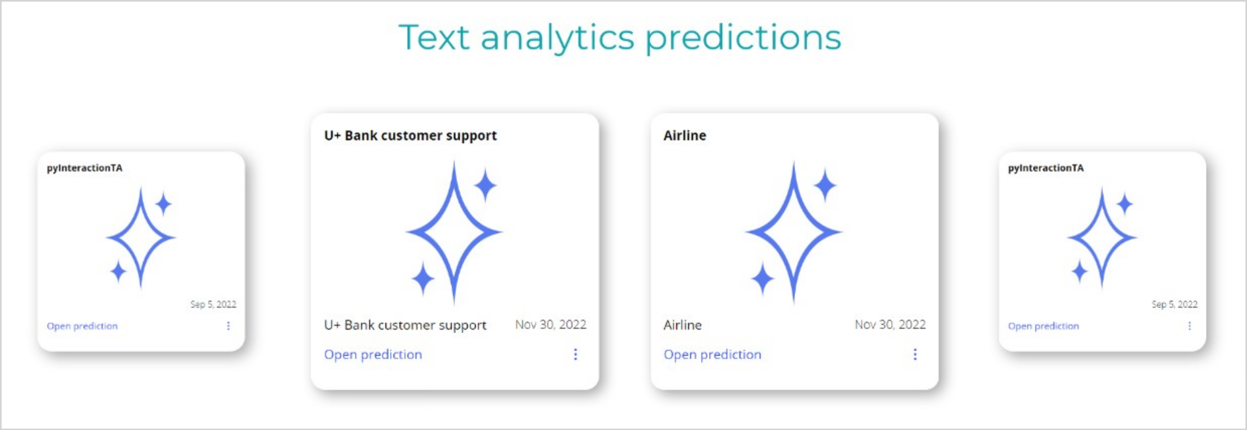
You can turn on the Show Analysis switch to see the chatbot details. Notice the customer's message. The topic detection model detects the topic of this message with 91 percent confidence, and the entity extraction model identifies the ticket number. Chatbot collects this information, includes it in a case, and then automatically routes it to a customer service representative for further processing.

Predictive models drive predictions. A data scientist manages predictions in Prediction Studio.

Prediction Studio is the dedicated workspace in Pega Platform where you manage the life cycle of predictive models and the predictions that they drive.

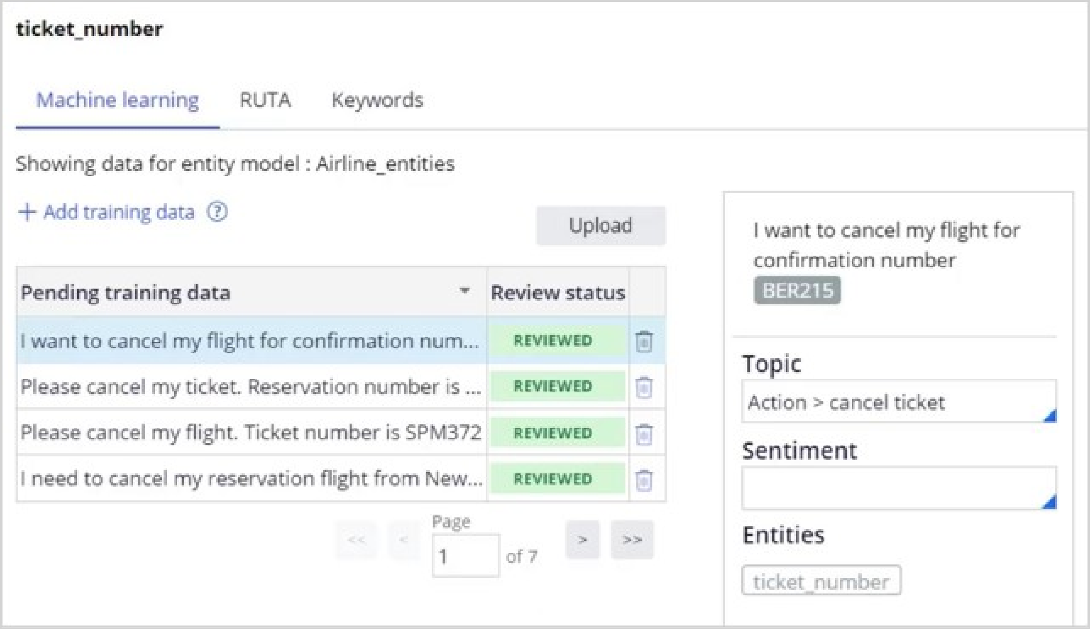
The workspace provides data scientists with everything they need to monitor and change predictions.

This is the Predictions landing page, where you manage predictions. Pega NLP uses Text analytics predictions.



Data scientists build topic detection model in a text prediction by using training data that they add manually or as a batch.

Data scientists also configure entity extraction in a text prediction. Several methods are available, including machine learning, Ruta scripts, or keyword-based entity extraction. It is also possible to use Google AutoML models. The machine learning-based model is the most versatile but requires training data to function. When adding entity extraction training data, you can also specify the topic of each message to further enhance the topic detection model.



Showing data for entity model : Airline_entities

Pending training data	Review status
I want to cancel my flight for confirmation num...	REVIEWED
Please cancel my ticket. Reservation number is ...	REVIEWED
Please cancel my flight. Ticket number is SPM372	REVIEWED
I need to cancel my reservation flight from New...	REVIEWED

Page 1 of 7

I want to cancel my flight for confirmation number BER215

Topic: Action > cancel ticket

Sentiment:

Entities: ticket_number

You have reached the end of this video. You have learned:

- How Pega NLP allows you to improve business processes by using predictions.

- How topic and sentiment detection models and entity extraction models drive text analytics predictions.
- That machine learning models require training data.

Text analytics for email routing

Description

Humans can effortlessly interpret a single tweet but are unable to parse a large volume of information efficiently. Businesses are exploring ways to use machine learning to extract meaningful information from a large number of text messages. Learn how a text prediction can work to detect topics, extract entities, and identity the sentiment for incoming emails.

Learning objectives

- Explain text analytics
- Describe practical applications where text analytics can be used
- Describe the role of machine learning in text analytics
- Explain how text predictions are trained on classified messages

Email routing

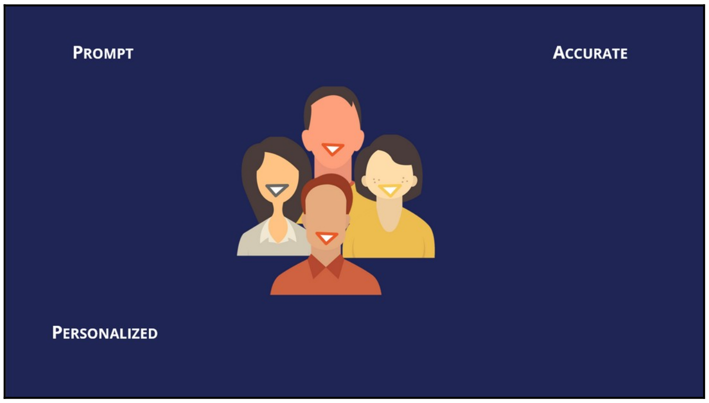
Introduction

A company has an email channel where its customers can send any email — from service requests to compliments to sales inquiries — to the product support team. This email variety makes it difficult to provide each customer with a prompt and personalized response. Pega Infinity™ uses AI-powered text analytics to perform intelligent email routing.

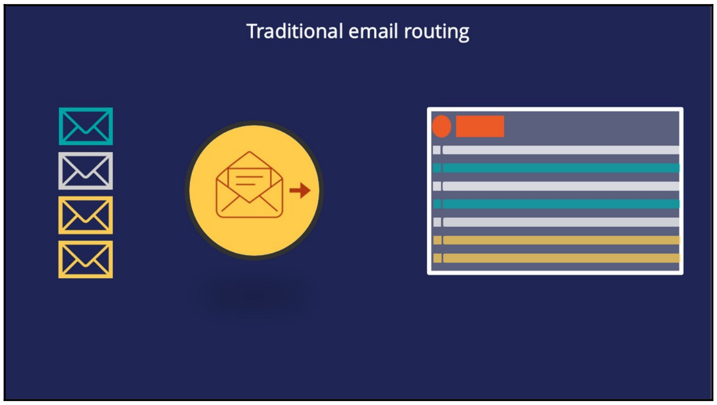
Transcript

In this video, you will learn how Pega Infinity™ uses AI-powered text analytics to do intelligent email routing.

Customer satisfaction is a reflection of what a customer expects from a company vs. what they experience from the company. Meeting, or even better, exceeding customer expectations means addressing their service requests and complaints promptly, accurately, and with personalized solutions. Doing this will ensure the customer has a great experience.

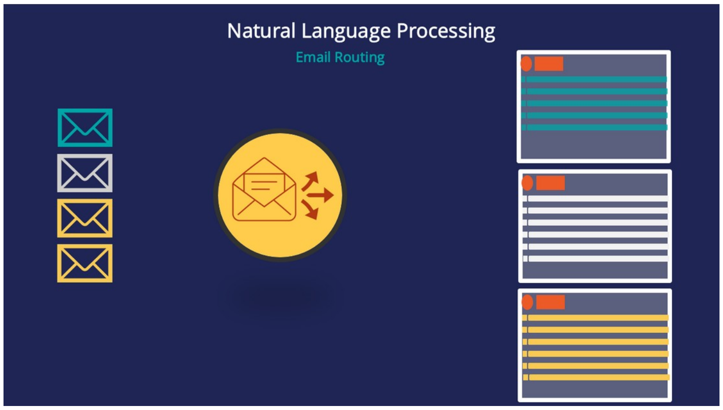


Assume a company has an email channel in which its customers can send any type of email —from service requests to compliments to sales inquiries—to the product support team. These emails are often routed to the same container and are uncategorized. This makes it difficult to provide each customer with a prompt and personalized response.

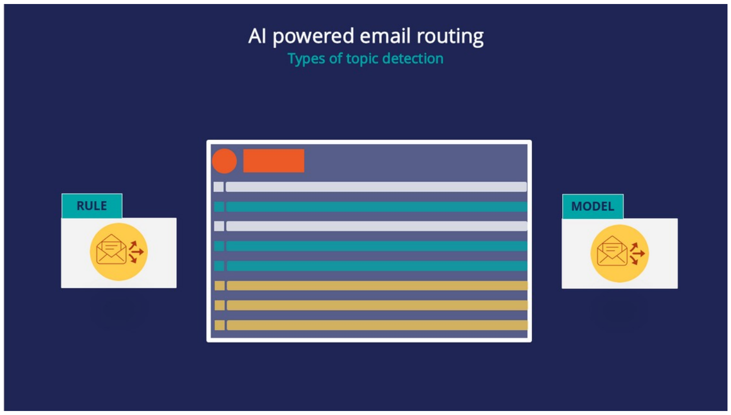


In Pega Infinity, this issue can be addressed by routing the emails using artificial intelligence. Pega Infinity uses AI-powered Natural Language Processing to detect the topic of an email and route the email to the appropriate container.

Consider emails from customers requesting an account address change, making a compliment, or requesting a new credit card. With the help of AI-powered text analytics, Pega Infinity is able to read and understand the content of each email and route it appropriately. This means customer service representatives can be alerted to any account-related service requests and resolve them quickly.



Email routing is done using the topic detection mechanism. The two types of topic detection are rule-based and model-based.

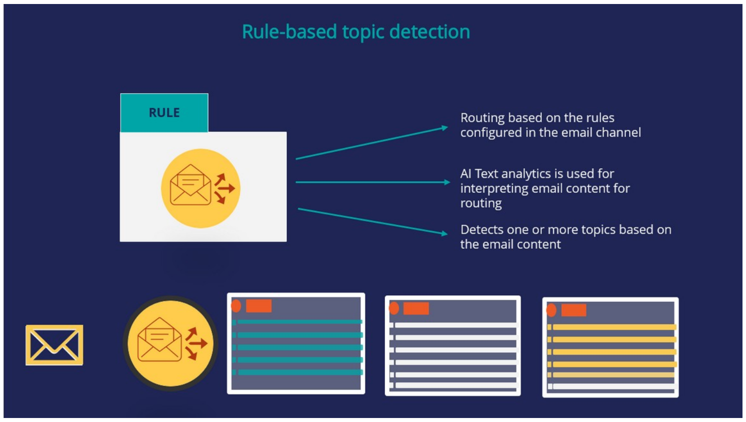


AI powered email routing

Types of topic detection

In rule-based topic detection the routing is based on the rules configured in the email channel. AI-powered text analytics is used to detect the topic of the email, and the channel rules route it to the right container. This type of topic detection may detect one or more topics if the email contains words associated with more than one topic.

Let’s consider a service request email from a customer. The email content is analyzed and routed to the right container. If an email from another customer contains words that are associated with two topics, the rule-based topic detection detects both topics. The email can then be routed to two different containers depending on how the channel rules are configured.



Rule-based topic detection

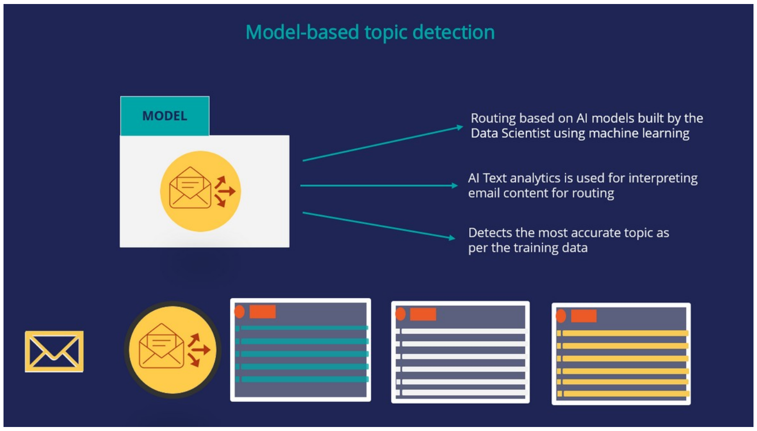
Routing based on the rules configured in the email channel

AI Text analytics is used for interpreting email content for routing

Detects one or more topics based on the email content

In model-based topic detection the routing is based on AI models built by a Data Scientist using machine learning. Building these models requires a training data set and a test data set. The data sets consist of a list of emails and the associated topic for each email. This type of topic detection identifies the most accurate topic based on the AI model and training set used by the Data Scientist.

Let’s consider the same service request email from the customer. The email’s content is analyzed and routed to the right container. If an email from another customer contains words associated with two topics during the training of the models, the model-based topic detection detects both topics but typically with a different accuracy factor. In this case, the topic with the highest accuracy factor is chosen.



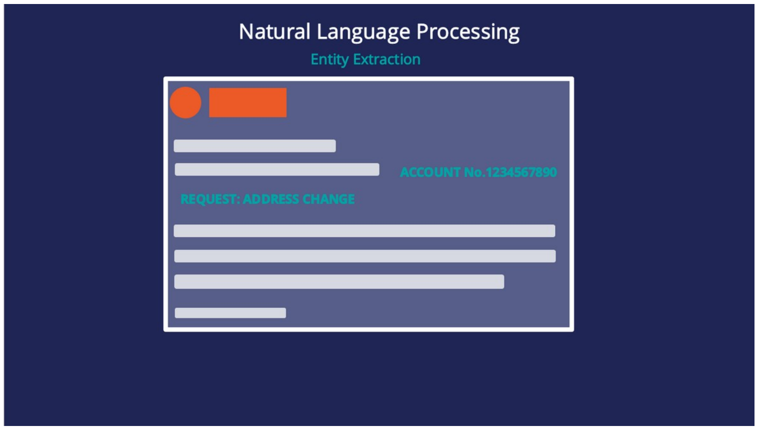
Model-based topic detection

Routing based on AI models built by the Data Scientist using machine learning

AI Text analytics is used for interpreting email content for routing

Detects the most accurate topic as per the training data

Pega Infinity also enables you to extract entities from an email. This means that when an email is sent, certain entities such as account number, email address, street address, etc. can be automatically detected and extracted. This allows certain emails to be automatically processed or given priority.



Natural Language Processing

Entity Extraction

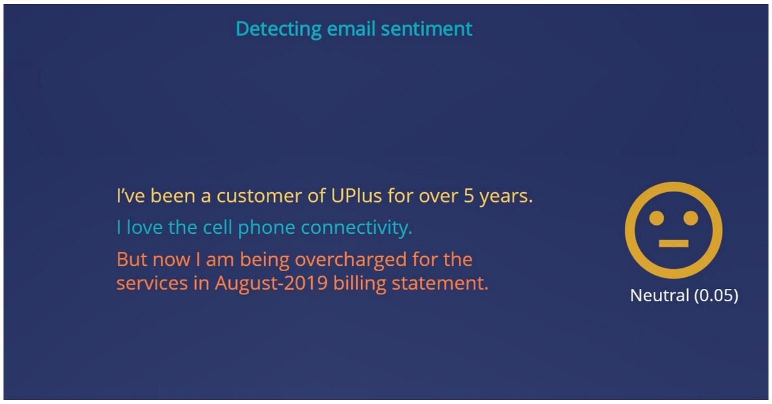
ACCOUNT No.1234567890

REQUEST: ADDRESS CHANGE

Besides Topic detection and Entity Extraction, Pega Infinity uses its AI-powered text analytics to enable you to detect the sentiment of an email based on its content.

Suppose a customer sends the following email to customer service.

The email’s content is a combination of different types of sentences. The first sentence has a neutral sentiment. In the second sentence, the customer expresses his satisfaction with the connectivity, so it has a positive sentiment. The third sentence is negative, as the customer is complaining about an overcharge. The overall sentiment of the email is determined based on the weight of the content sentiments. In this specific example the overall sentiment is neutral with a sentiment score of 0.05.

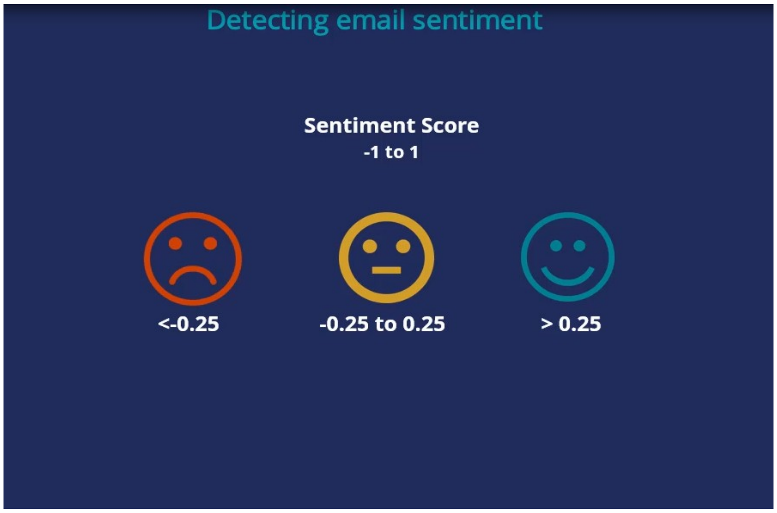


Detecting email sentiment

I've been a customer of UPlus for over 5 years.
I love the cell phone connectivity.
But now I am being overcharged for the services in August-2019 billing statement.

Neutral (0.05)

The sentiment score is a value between -1 and 1. In the out-of-the-box configuration, a sentiment score <-0.25 results in a Negative sentiment, a sentiment score between -0.25 and 0.25 results in a Neutral sentiment, and a sentiment score above 0.25 results in a positive sentiment.

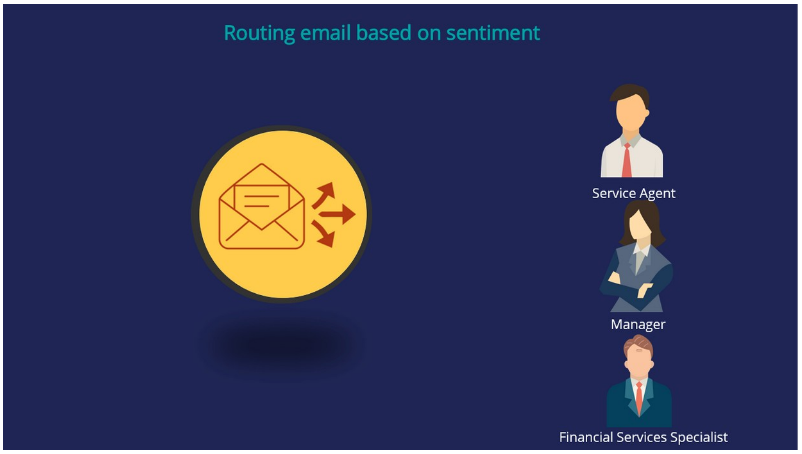


Detecting email sentiment

Sentiment Score
-1 to 1

<-0.25 **-0.25 to 0.25** **> 0.25**

Once the email sentiment is detected, you can configure the email channel to route a specific topic with a specific sentiment to a specialized agent for a quick and personalized response. For example, you could route an address change with a neutral sentiment to a Service Agent, a complaint email with a negative sentiment to a Manager, and a credit card inquiry with a negative sentiment to a Financial Services Specialist.


Routing email based on sentiment
Service Agent
Manager
Financial Services Specialist

In summary, Pega Infinity’s AI-based email routing capability enables customer service representatives to be more productive, reduces request processing time, and improves the customer experience by providing prompt and personalized service.

Training a topic model to improve email routing

Introduction

U+ Bank uses Pega Customer Service™ to route incoming emails to the appropriate department based on the topic of the email. For several use cases (for example, an address change), emails are routed based on keywords that are detected in the message. To improve the email routing, learn how to train the text prediction with a data set that contains classified messages.

Transcript

This demo shows you how to train a text prediction to improve email routing.

U+ Bank uses machine learning to route inbound messages in the email channel to the appropriate department based on the topic of the email.

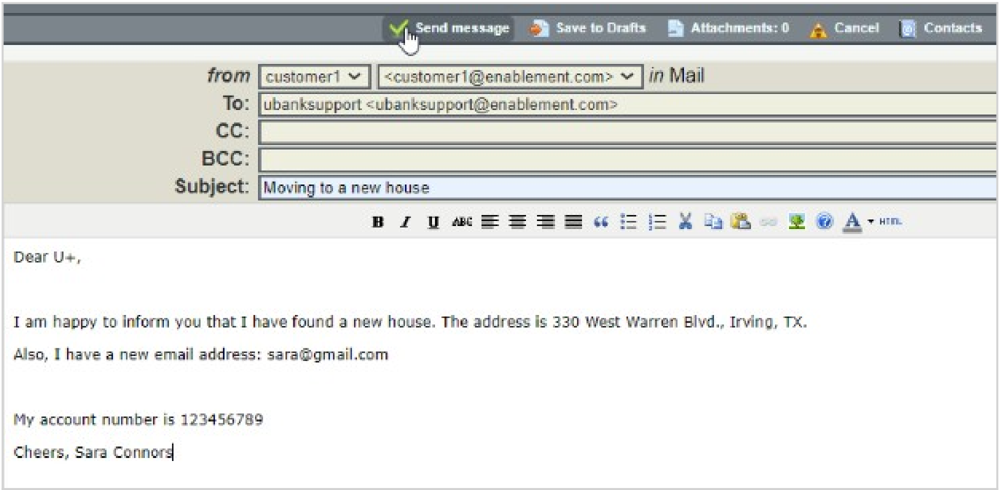
A text prediction that aims to detect the topic of the message drives the routing.

U+ Bank customer support

U+ Bank customer support

Mar 11, 2021

For example, when Sara writes an email to inform U+ Bank that she has moved to a new house, the text prediction detects an address change as the topic.



from customer1 <customer1@enablement.com> in Mail
To: ubanksupport <ubanksupport@enablement.com>
CC:
BCC:
Subject: Moving to a new house

Dear U+,

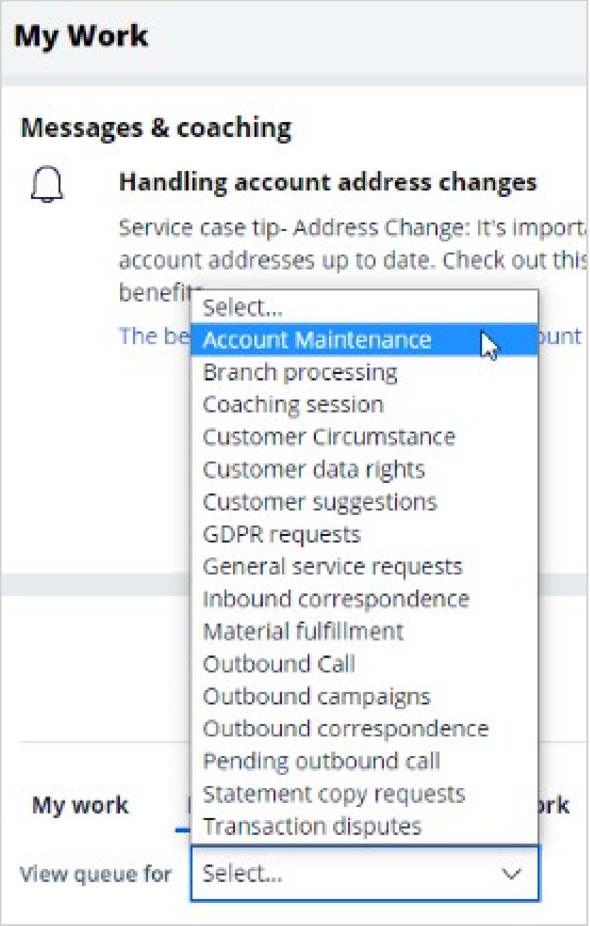
I am happy to inform you that I have found a new house. The address is 330 West Warren Blvd., Irving, TX.

Also, I have a new email address: sara@gmail.com

My account number is 123456789

Cheers, Sara Connors

The **Account maintenance** department receives all emails where an address change is detected as the topic of the message.


My Work

Messages & coaching

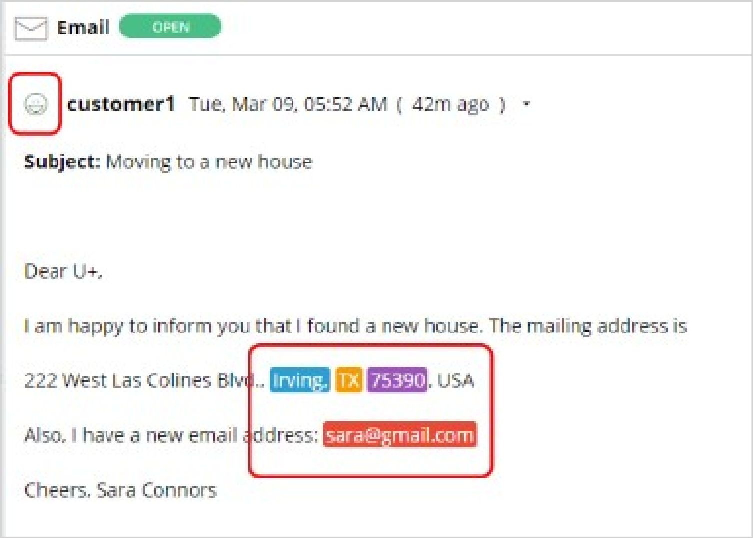
Handling account address changes

Service case tip- Address Change: It's import
account addresses up to date. Check out this
benefit

The be unt

View queue for Select...

The text prediction also detects entities such as a ZIP Code and an email address and the overall sentiment of the message.



Email OPEN

customer1 Tue, Mar 09, 05:52 AM (42m ago)

Subject: Moving to a new house

Dear U+,

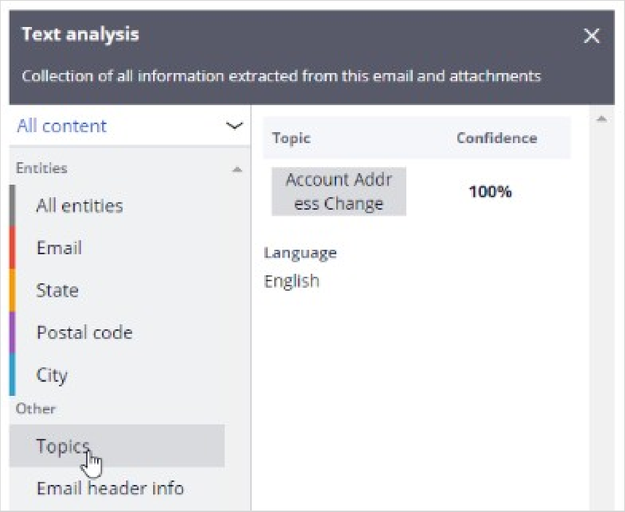
I am happy to inform you that I found a new house. The mailing address is

222 West Las Colines Blvd., Irving, TX 75390, USA

Also, I have a new email address: sara@gmail.com

Cheers, Sara Connors

Notice that the topic Account Address Change is detected with 100% confidence.



Text analysis

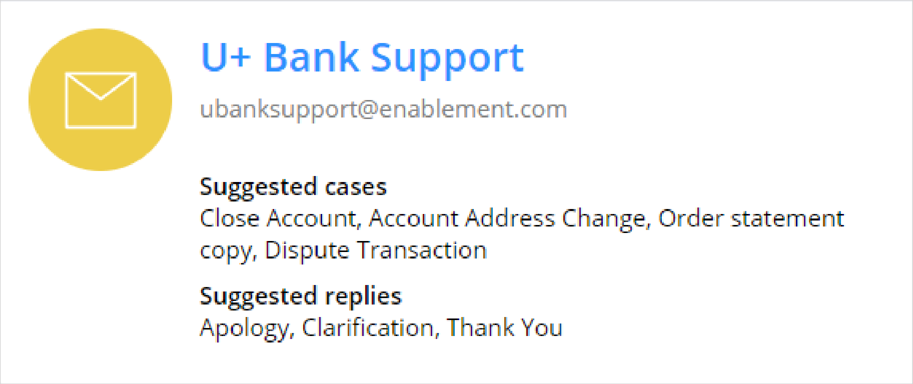
Collection of all information extracted from this email and attachments

Topic	Confidence
Account Address Change	100%

Language

English

The intelligent routing is set up in the email channel, in App Studio.



U+ Bank Support

ubanksupport@enablement.com

Suggested cases

Close Account, Account Address Change, Order statement copy, Dispute Transaction

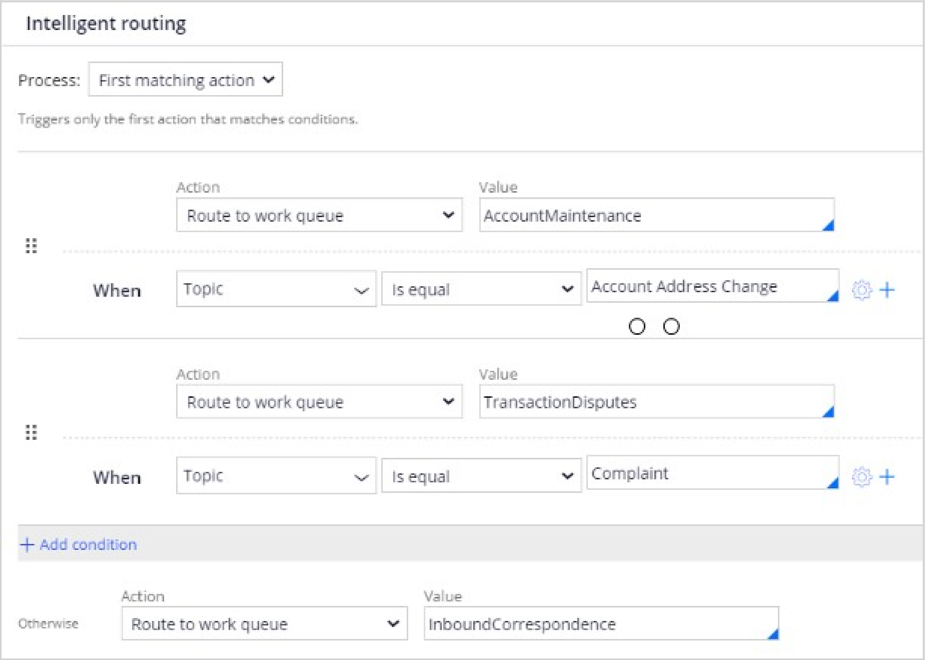
Suggested replies

Apology, Clarification, Thank You

An email is routed to the work queue of the Account maintenance department when the detected topic is an address change.

Complaints are routed to the **Transaction Disputes** department.

If the address change and complaint topics are not detected, the email is routed to a default work queue.



Intelligent routing

Process: First matching action

Triggers only the first action that matches conditions.

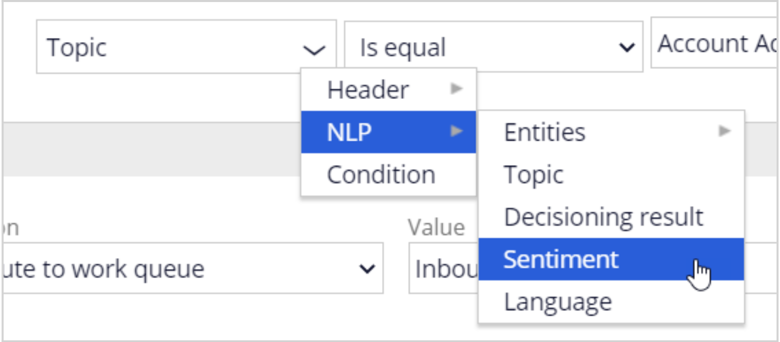
Action: Route to work queue
Value: AccountMaintenance
When: Topic | Is equal | Account Address Change

Action: Route to work queue
Value: TransactionDisputes
When: Topic | Is equal | Complaint

Otherwise
Action: Route to work queue
Value: InboundCorrespondence

Note that in the **When** condition of the email routing, you use the outcome of the text prediction.

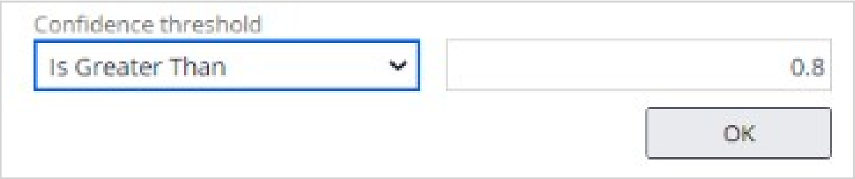
Optionally, you can also use the sentiment of the email in the routing conditions.



Topic | Is equal | Account Ac

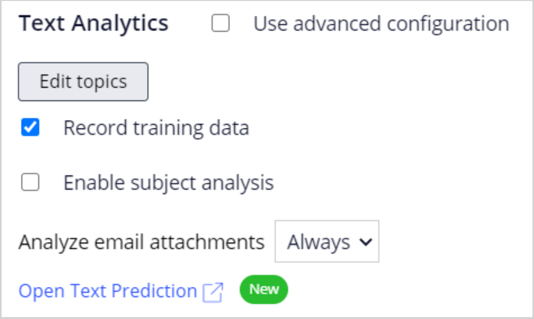
n | Value
ute to work queue | Inbou

You can set a threshold for the confidence with which a topic must be detected to trigger routing.



Confidence threshold
Is Greater Than | 0.8

Every channel is associated with a text prediction.

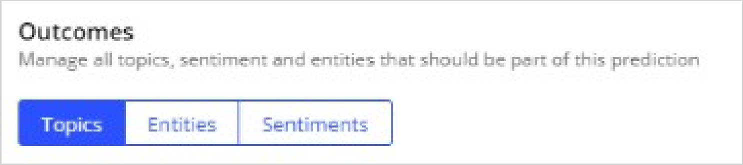

Text Analytics ☐ Use advanced configuration

☑ Record training data

☐ Enable subject analysis

Analyze email attachments: Always

A text prediction is driven by predictive models that detect topics, entities, and sentiments.

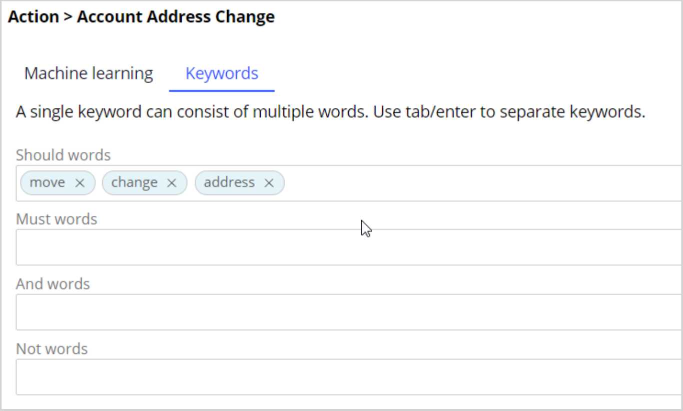

Outcomes

Manage all topics, sentiment and entities that should be part of this prediction

Many entity extraction models, and a sentiment model, are available out of the box.

For topic detection, Prediction Studio supports keyword models as well as machine learning models based primarily on training data.

A keyword model uses **Should words**, **Must words**, **And words**, and **Not words**.


Action > Account Address Change

A single keyword can consist of multiple words. Use tab/enter to separate keywords.

Should words: move, change, address

Must words:

And words:

Not words:

If any of the **Should words** appear in a piece of text, topic detection assigns that text to the corresponding topic. To achieve accurate results, create an exhaustive list of **Should words**.

Only if all **Must words** appear in a piece of text, will the topic detection assign that text to the corresponding topic.

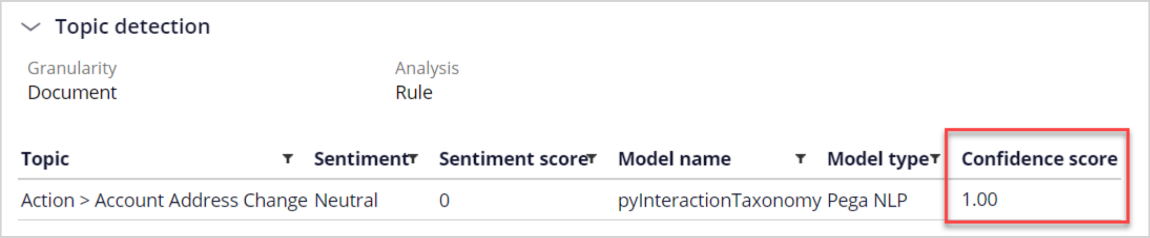
Use **And words** to distinguish between similar topics while using identical **Must words**.

If a **Not word** appears in a piece of text, the text is not assigned to the corresponding topic.

You can test the output of a prediction on a sample message.

Two topics are correctly detected: an address change and a complaint.

Notice that the confidence score for both topics is **1**. The keyword model performs a Boolean match based on the presence or absence of words and detects the topic with absolute certainty.



Topic detection

Granularity: Document

Analysis: Rule

Topic	Sentiment	Sentiment score	Model name	Model type	Confidence score
Action > Account Address Change	Neutral	0	pyInteractionTaxonomy	Pega NLP	1.00

It is highly recommended to use a machine learning model for topic detection. It determines the confidence score based on evidence.

A machine learning model is based on training data that consists of categorized texts.

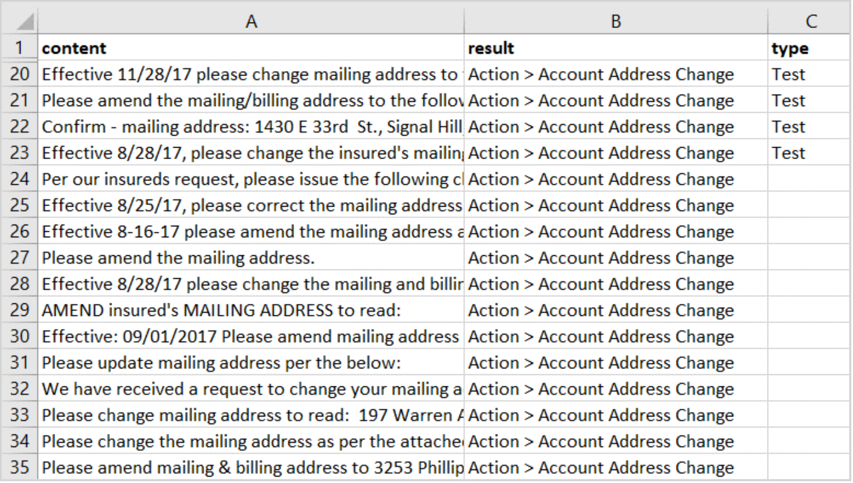
You can get training data from two sources. The first is training data accumulated through the application that is running in production.

When a customer service representative corrects the topic of an incoming email, this change is added to the training data.

After the messages are reviewed and approved, they are used to rebuild the topic model.

Also, you can choose to import data, provided you have accumulated data in the past.

The file must contain the message and the associated topic.



	A	B	C
1	**content**	**result**	**type**
20	Effective 11/28/17 please change mailing address to	Action > Account Address Change	Test
21	Please amend the mailing/billing address to the follov	Action > Account Address Change	Test
22	Confirm - mailing address: 1430 E 33rd St., Signal Hill	Action > Account Address Change	Test
23	Effective 8/28/17, please change the insured's mailin	Action > Account Address Change	Test
24	Per our insureds request, please issue the following c	Action > Account Address Change	
25	Effective 8/25/17, please correct the mailing address	Action > Account Address Change	
26	Effective 8-16-17 please amend the mailing address a	Action > Account Address Change	
27	Please amend the mailing address.	Action > Account Address Change	
28	Effective 8/28/17 please change the mailing and billir	Action > Account Address Change	
29	AMEND insured's MAILING ADDRESS to read:	Action > Account Address Change	
30	Effective: 09/01/2017 Please amend mailing address	Action > Account Address Change	
31	Please update mailing address per the below:	Action > Account Address Change	
32	We have received a request to change your mailing a	Action > Account Address Change	
33	Please change mailing address to read: 197 Warren A	Action > Account Address Change	
34	Please change the mailing address as per the attache	Action > Account Address Change	
35	Please amend mailing & billing address to 3253 Phillip	Action > Account Address Change	

Notice that training data for the address change and the complaint topics are in the pending state.

You can now rebuild the models.

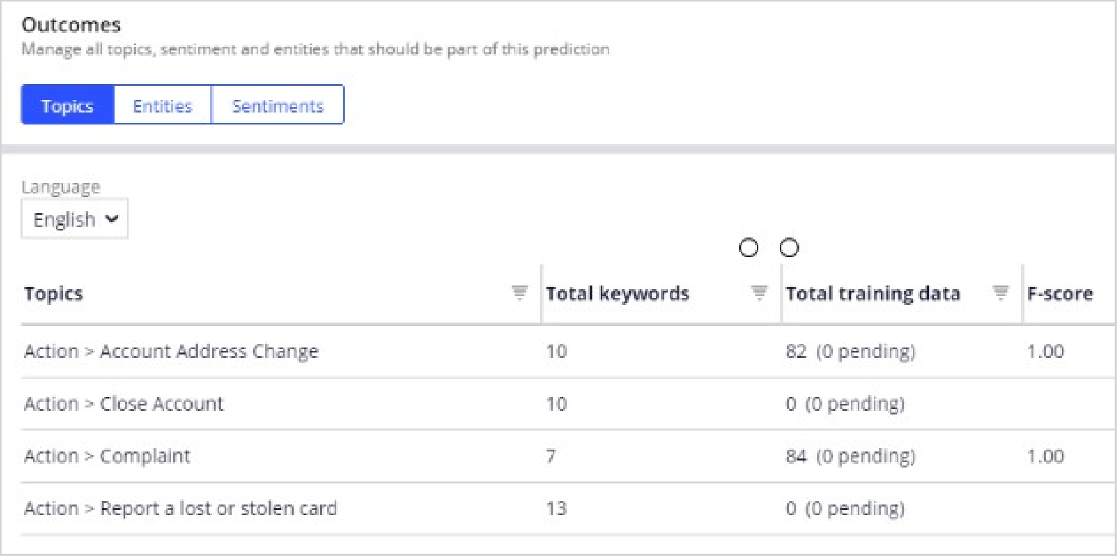
You can select individual models or rebuild all models across model types and languages.

While training is in progress, you have the option to cancel.

When the process completes, you can view the latest model report on the **Models** tab.

The report contains the validation data, the confusion matrix, and the score sheet.

The topic detection now uses machine learning models based on the training data and the keywords provided for the address change and complaint topics.



Outcomes

Manage all topics, sentiment and entities that should be part of this prediction

Topics	Total keywords	Total training data	F-score
Action > Account Address Change	10	82 (0 pending)	1.00
Action > Close Account	10	0 (0 pending)	
Action > Complaint	7	84 (0 pending)	1.00
Action > Report a lost or stolen card	13	0 (0 pending)	

The **Should words** and **Must words** act as positive features for matching text to a topic, while the **Not words** act as negative features.

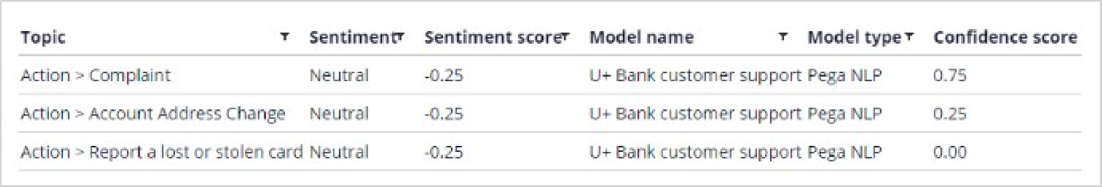
But the training and testing data have the greatest impact on your machine learning model, while keywords have a smaller impact.

Once the models are rebuilt, you can test the prediction.

Notice that the topics are detected with a confidence score below one.

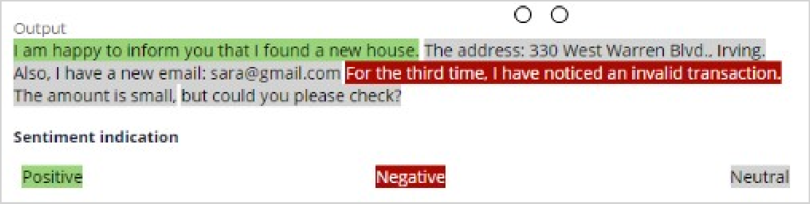
In this case, the topic message is recognized as a complaint with high confidence. The address change topic is detected, but with a lower confidence score.

When multiple topics are detected in a message, comparison of the confidence scores allows selection of the one with the highest priority.



Topic	Sentiment	Sentiment score	Model name	Model type	Confidence score
Action > Complaint	Neutral	-0.25	U+ Bank customer support	Pega NLP	0.75
Action > Account Address Change	Neutral	-0.25	U+ Bank customer support	Pega NLP	0.25
Action > Report a lost or stolen card	Neutral	-0.25	U+ Bank customer support	Pega NLP	0.00

The other outcomes of the text prediction are the entities detected, such as an email address and the sentiment of the message.



Output

I am happy to inform you that I found a new house. The address: 330 West Warren Blvd., Irving. Also, I have a new email: sara@gmail.com For the third time, I have noticed an invalid transaction. The amount is small, but could you please check?

Sentiment indication

Positive Negative Neutral

You have reached the end of this demo. What did it show you?

- How Pega Customer Service routes incoming emails to the appropriate department based on the topic of the email.
- How text predictions work to predict the topic and sentiment of a message and detect entities.
- How to train a topic model and use machine learning to identify the topic correctly.

Using entity extraction with chatbot channel

Description

Better understand the key features and benefits of entity extraction with chatbot channel. Use this module to learn how to enable the chatbot to handle ticket cancellation requests by training a text prediction that drives the chatbot. The chatbot detects the topic of a message, extracts all relevant entities, and creates a case that the Customer Service Representative (CSR) in the CSR portal can handle later.

Learning objectives

- Describe how Pega Chatbot uses natural language processing to determine the topic of inbound message and extract entities
- Configure the chatbot channel
- Explain the out-of-the-box entity extraction models
- Train the topic detection and entity extraction models

Training predictions for chatbot

Introduction

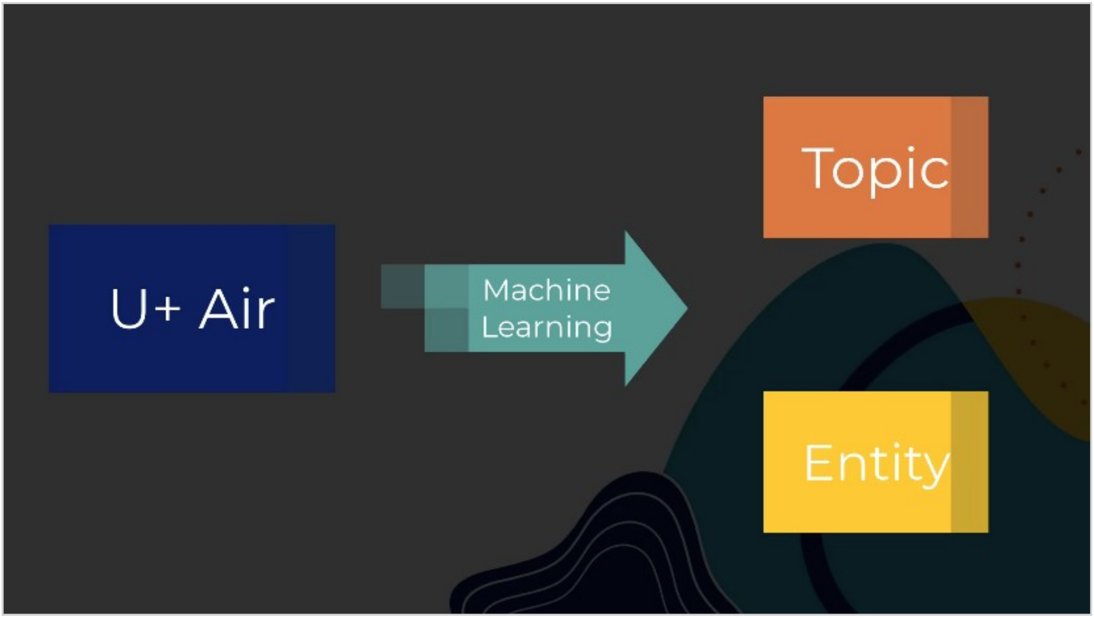
Web Chatbot is a feature in Pega Customer Service™ that allows your customers to enter questions or requests in a chat window that is accessible from a customer-facing web page, and then receive answers and guidance.

In a web chatbot session, the chatbot responds to a customer request, such as *I want to cancel a ticket*, by detecting the right topic with machine learning. The chatbot then walks the customer through a set of questions. The chatbot gathers the required information by leveraging entity extraction models. Learn how to train these topic detection and entity extraction models.

Transcript

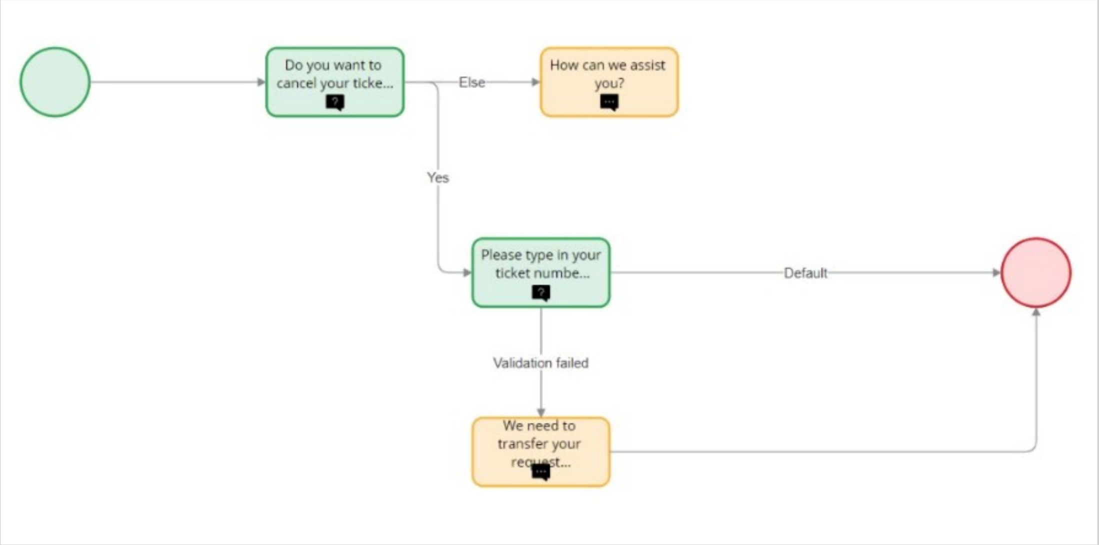
This demo shows you how to train a text prediction to enable the chatbot to handle ticket cancellation requests.

U+ Air uses machine learning to identify the topic of the incoming message and detect entities such as ticket numbers.



A text prediction that aims to detect the topic and entities included in the message drives the chatbot.

For example, when Troy, a customer, sends a message to the Airline chatbot that he wants to cancel his ticket, the text prediction detects the topic of the message and runs the *Cancel a ticket* case type that an application developer preconfigured in the system. The case type contains the conversation flow that the chatbot uses in conversation with Troy. Notice the conversation elements such as *Ask a question* or *Send a message*. Observe how the chatbot follows the conversation flow during interaction with Troy.


Do you want to cancel your ticke...
Else
How can we assist you?
Yes
Please type in your ticket numbe...
Default
Validation failed
We need to transfer your request...

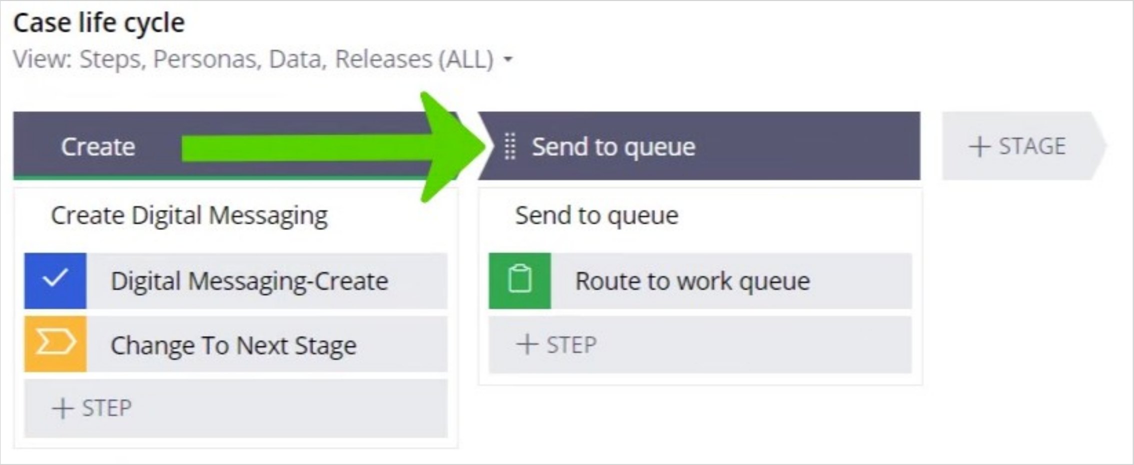
The prediction detects the ticket number entity that Troy provides, and then asks for confirmation.

Notice that the chatbot skips the *Please type in your ticket number* question because the text prediction detected that Troy provided the ticket number in the first message.

See how the chatbot reacts if Troy does not provide the ticket number in the first message.

Instead of skipping the question, the chatbot asks Troy to provide the ticket number.

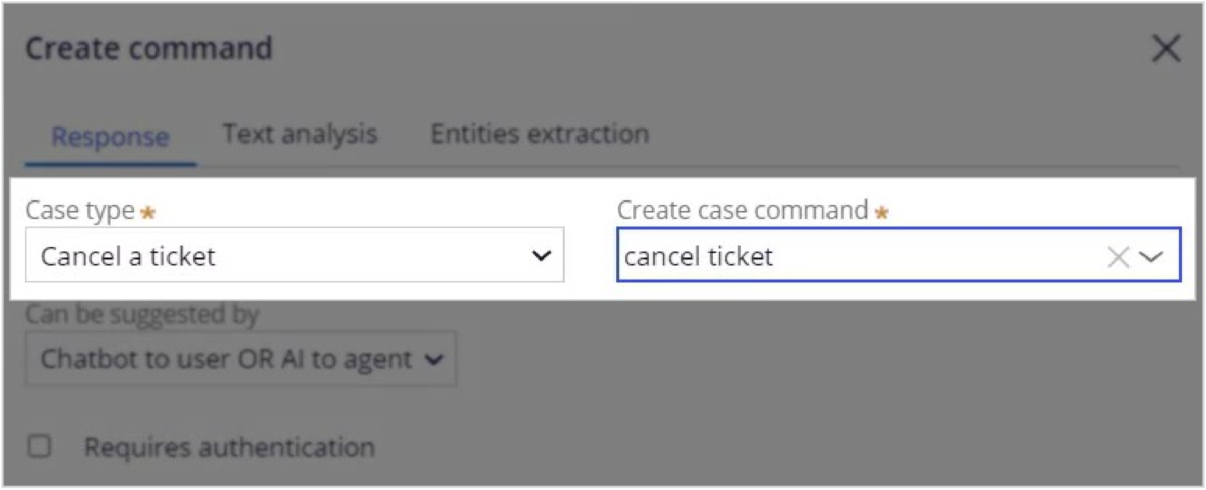
Once the conversation ends, the chatbot saves the necessary information and routes the case to an appropriate **Customer Service** workbasket.



Note that the case contains the ticket number extracted by the text prediction during the conversation with Troy.

To make the chatbot available to customers, you first configure the digital messaging channel in App Studio.

Notice that the *Cancel a ticket* case type is specified. This is the same case type that contains the conversation flow that the chatbot uses in the conversation with Troy. The **Create case command** represents the topic in the text prediction. The chatbot triggers the case type creation when the text prediction detects this specific topic.



Create command

Case type *: Cancel a ticket

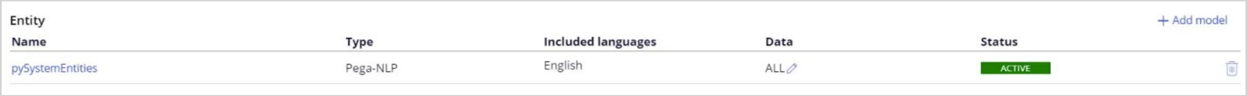
Create case command *: cancel ticket

Can be suggested by: Chatbot to user OR AI to agent

☐ Requires authentication

The topic detection and entities extraction are driven by a text prediction that is automatically generated after chatbot channel creation. Data scientist builds the models in the prediction in Prediction Studio. Models learn with training data. In this scenario, the training data contains three topics: *action book ticket*, *action cancel ticket*, and *action reschedule*. Besides topic detection models, the prediction contains entity extraction models that extract necessary information from the incoming chat message.

Notice the existing *pySystemEntities* model. It is an out-of-the-box entity extraction model that contains basic entities; you cannot modify it. To add a new entity type such as ticket number, you create a new entity extraction model.



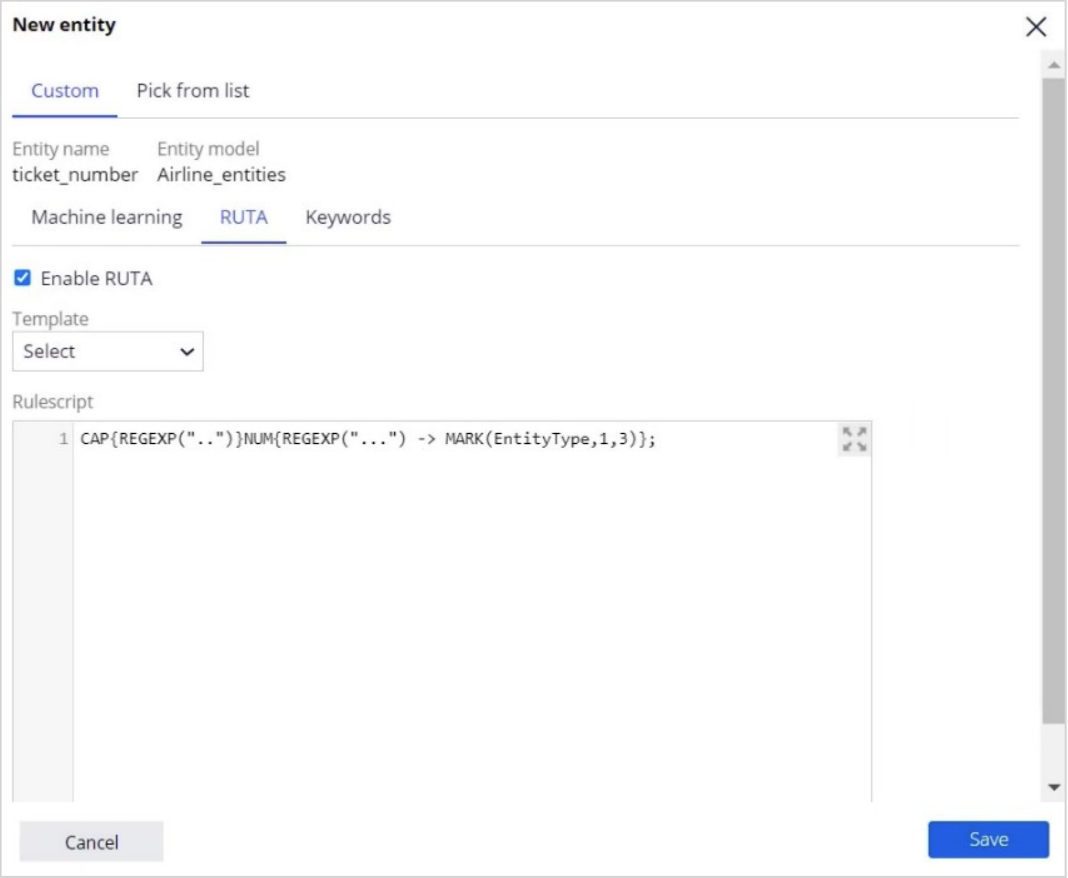
Entity

Name	Type	Included languages	Data	Status
pySystemEntities	Pega-NLP	English	ALL	ACTIVE

After creation of the new model, you can create a new entity.

U+ Air determines that using Rule-based Text Annotation (Ruta) scripts is the best choice to detect the strict ticket number pattern. Ruta is a rule-based script language which detects keywords and phrases that follow specific patterns.

Notice the rulescript that detects the ticket number. This script detects ticket numbers that follow a strict pattern of two letters followed by three digits. In case of a rescheduled ticket or human error, the application also needs to detect unusual ticket number patterns. For example, a dash in between the letters and numbers, or three letters and three digits. Using Ruta with machine learning can achieve this business outcome.



Entity name: ticket_number
Entity model: Airline_entities

☑ Enable RUTA

Template: Select

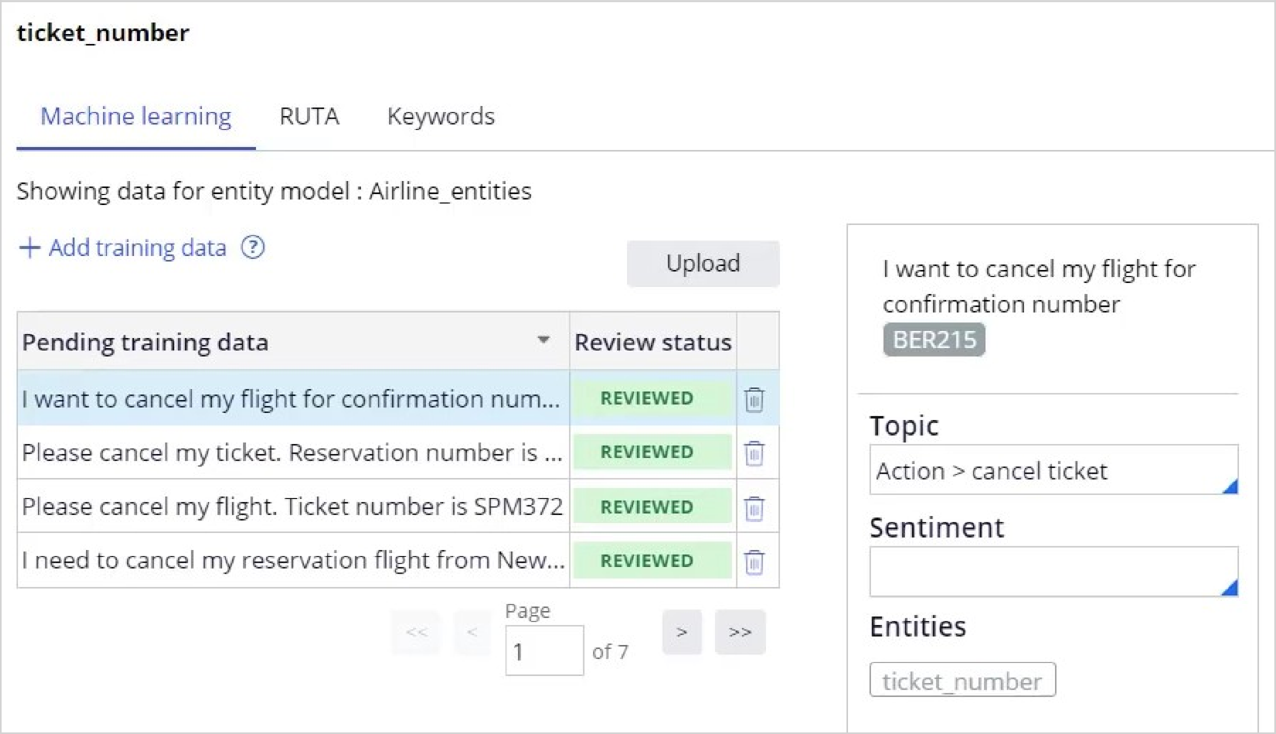
Rulescript

```
CAP{REGEXP("..")}NUM{REGEXP("...") -> MARK(EntityType,1,3)};
```

A machine learning-based model must be built to enable machine learning entity extraction. You add training data to the ticket number entity to build the model; you can add training data one record at a time.

A training data record is a typical chat message that the customer sends. After adding a training data record, you manually identify the entities you want the model to recognize. In this case, JK-294 is a ticket number entity. Optionally, to improve the topic detection model, you can specify the topic of the message that provides additional training data.

You can also use a batch import from a dataset file to provide additional training data. Notice the tagging that identifies a specific entity in each record. It is possible to have multiple tags to train multiple entity types simultaneously.



ticket_number

Showing data for entity model : Airline_entities

Pending training data	Review status
I want to cancel my flight for confirmation num...	REVIEWED
Please cancel my ticket. Reservation number is ...	REVIEWED
Please cancel my flight. Ticket number is SPM372	REVIEWED
I need to cancel my reservation flight from New...	REVIEWED

Page 1 of 7

I want to cancel my flight for confirmation number BER215

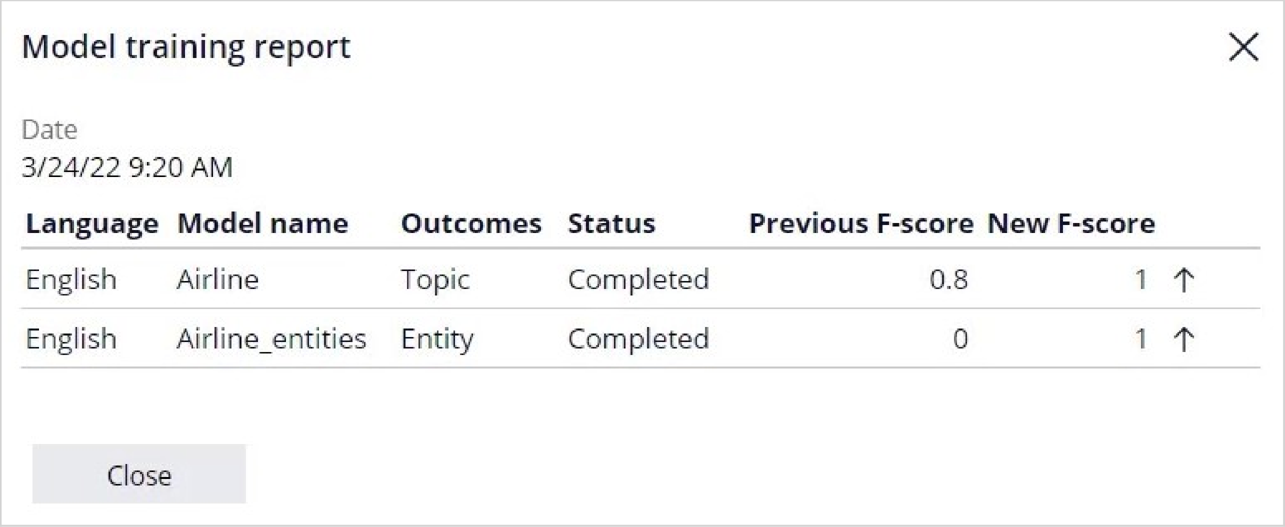
Topic: Action > cancel ticket

Sentiment:

Entities: ticket_number

After uploading the training data file, you can review each record and ensure that it was correctly imported. The correct ticket number entity and topic are flagged.

After uploading the dataset, build the models. Notice the New F-score for the Airline and Airline_entities models. Now, the entity extraction model can also detect unusual ticket number patterns.



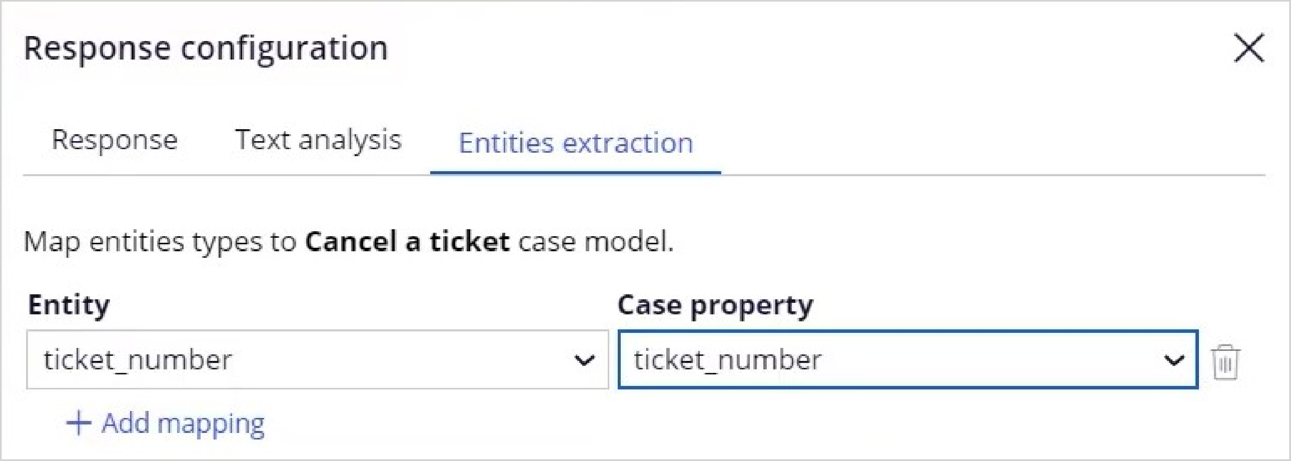
Model training report

Date
3/24/22 9:20 AM

Language	Model name	Outcomes	Status	Previous F-score	New F-score
English	Airline	Topic	Completed	0.8	1 ↑
English	Airline_entities	Entity	Completed	0	1 ↑

To enable the chatbot to use the extracted entity in the created case, an application developer maps the entity to a case property in a case command.

The system uses this property in the *Cancel a ticket* case type.

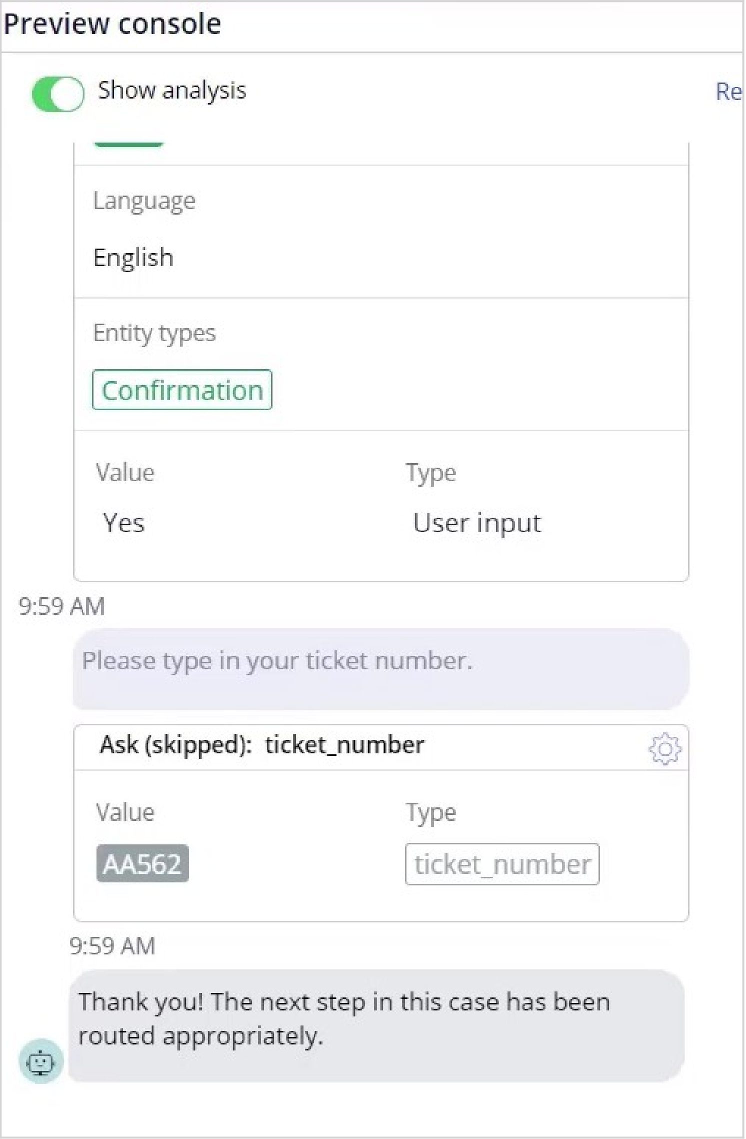


Response configuration

Map entities types to **Cancel a ticket** case model.

Entity	Case property
ticket_number	ticket_number

With all models in the text prediction built, and all necessary properties mapped, it is possible to see in full detail how Pega natural language processing (NLP) works. The digital messaging interface has a built-in preview console that simulates the chatbot. After sending the *I want to cancel my ticket number AA562* message, you can turn on the **Show Analysis** switch to see the chatbot details. In this case, the chatbot detects the topic of the message with 94% confidence. The confidence score, which is a value from 1 to 100, represents the confidence of the prediction. The chatbot detects the ticket number entity in the first message, skips the ticket number question, and then routes the case to a correct workbasket.



Preview console

Show analysis

Re

Language

English

Entity types

Confirmation

Value	Type
Yes	User input

9:59 AM

Please type in your ticket number.

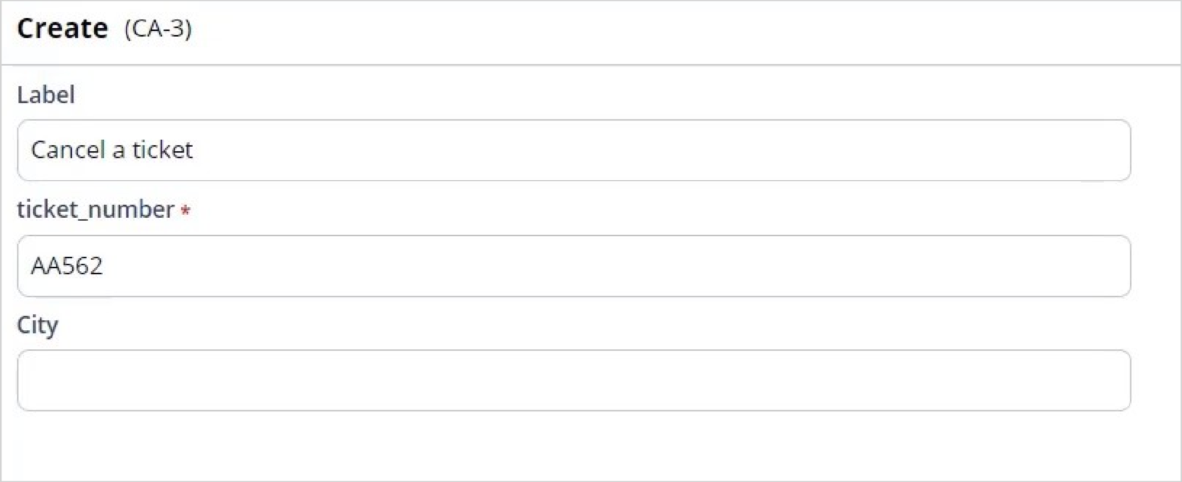
Ask (skipped): ticket_number

Value	Type
AA562	ticket_number

9:59 AM

Thank you! The next step in this case has been routed appropriately.

The case that the chatbot creates is available for review by the customer service representative in the CSR portal. Notice that because of the detected topic, the system creates a *Cancel a ticket case*. The case contains the ticket number extracted by the text prediction during the conversation with the customer.



Create (CA-3)

Label

Cancel a ticket

ticket_number *

AA562

City

You have reached the end of this video. What did it show you?

- How to create a chatbot channel.
- How to create a topic detection model.
- How to create an entity extraction model with machine learning and Ruta script.