



AI for 1:1 Customer Engagement

STUDENT GUIDE

© Copyright 2023
Pegasystems Inc., Cambridge, MA
All rights reserved.

This document describes products and services of Pegasystems Inc. It may contain trade secrets and proprietary information. The document and product are protected by copyright and distributed under licenses restricting their use, copying, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

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.

PegaRULES, Process Commander, SmartBPM® and the Pegasystems logo are trademarks or registered trademarks of Pegasystems Inc. All other product names, logos and symbols may be registered trademarks of their respective owners.

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: AI for 1:1 Customer engagement

Product: Pega Customer Decision Hub™ '23

URL: <https://academy.pega.com/mission/ai-11-customer-engagement/v3>

Date: 22 October 2023

Contents

Customer Decision Hub overview.....	5
1:1 Customer Engagement paradigm.....	6
The cross-sell on a website use case.....	15
Critical roles in implementing Pega AI.....	20
Exploring Next-best-Action decisions.....	22
Monitoring the health of the system using Impact Analyzer.....	28
Customer Decision Hub predictions.....	35
Customer Decision Hub predictions.....	36
Customizing Prediction Studio.....	39
Adaptive models.....	41
Adaptive analytics.....	42
Action arbitration with AI.....	47
Predictors of adaptive models.....	51
Monitoring adaptive models.....	54
Regular monitoring of adaptive models.....	55
Monitoring predictions.....	57
Monitoring adaptive models.....	63
Exporting adaptive model data.....	70
Exporting historical data.....	71
Exporting selected model data for external analysis.....	75
Data scientist tools for Customer Decision Hub.....	79
Creating predictions.....	85
Creating a churn prediction using a scorecard.....	86
Predictive models.....	90
Building models with Pega machine learning.....	94
Importing predictive models.....	101
Using machine learning services.....	106
Creating a prediction using an ML model.....	113

MLOps.....	115
Change management process.....	116
Challenging a predictive model.....	120
Shadowing an Adaptive Model.....	125
Creating and understanding decision strategies.....	127
Decision strategies.....	128
Creating a decision strategy.....	134
Decision strategy execution.....	140
Defining prediction patterns.....	151
Creating parameterized predictors.....	152
Using predictions in engagement strategies.....	157
Model governance.....	163
Model transparency.....	164
Explainable AI.....	166
Ethical bias.....	168
Detecting unwanted bias in engagement policies.....	170

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.

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.

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.



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.



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



**Should we
arbitrate?**

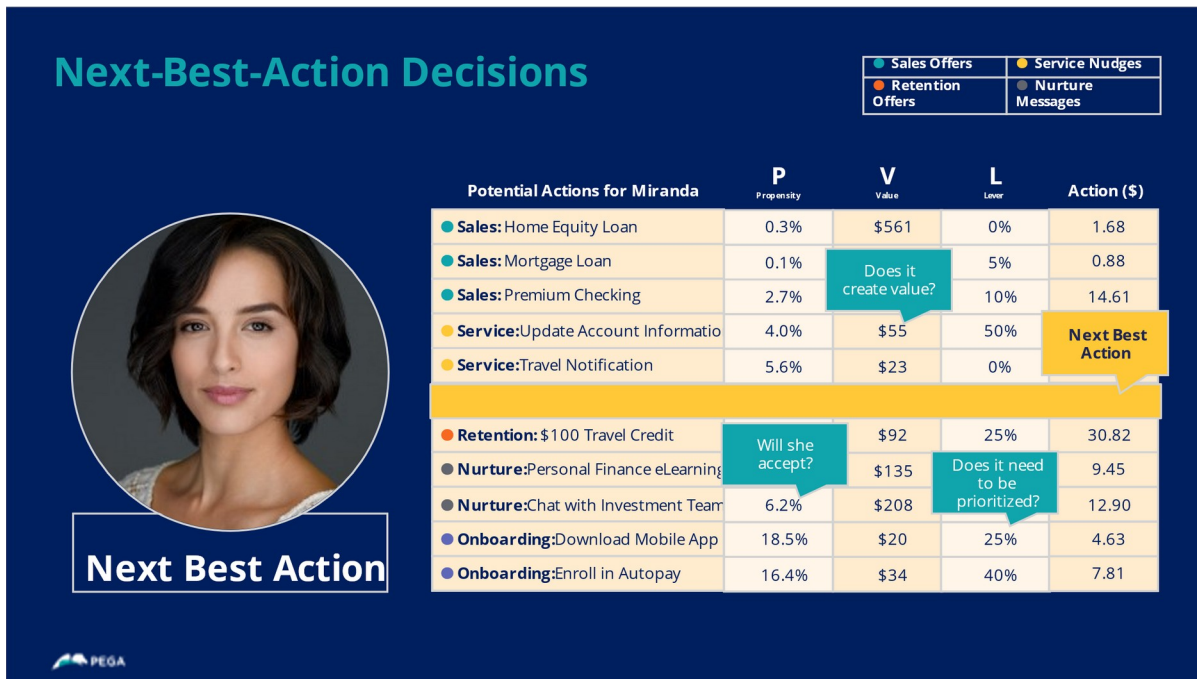
Potential Actions for Miranda	Eligible?	Applicable?	Suitable?	Arbitrate?
● Sales: Rewards Card	✓	✓	X	No
● Sales: Home Equity Loan	✓	✓	✓	Yes
● Sales: Mortgage Loan	✓	✓	✓	Yes
● Sales: Premium Checking	✓	✓	✓	Yes
● Service: Update Account Information	✓	✓	✓	Yes
● Service: Fraud Alert	✓	X	—	No
● Service: Travel 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

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

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.



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

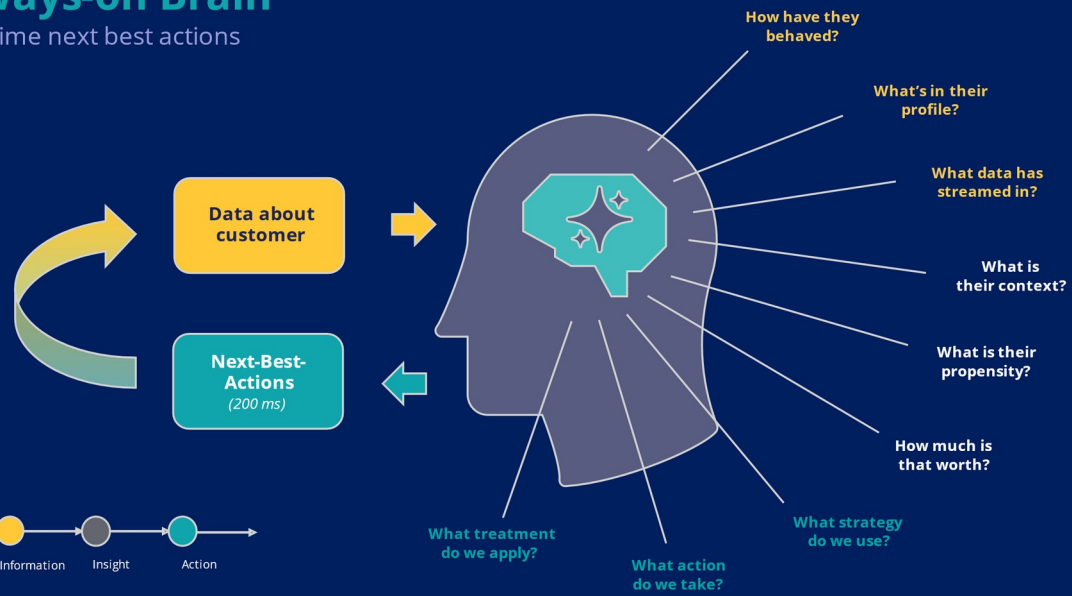
It uses data about the customer, including past interactions, as input.

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



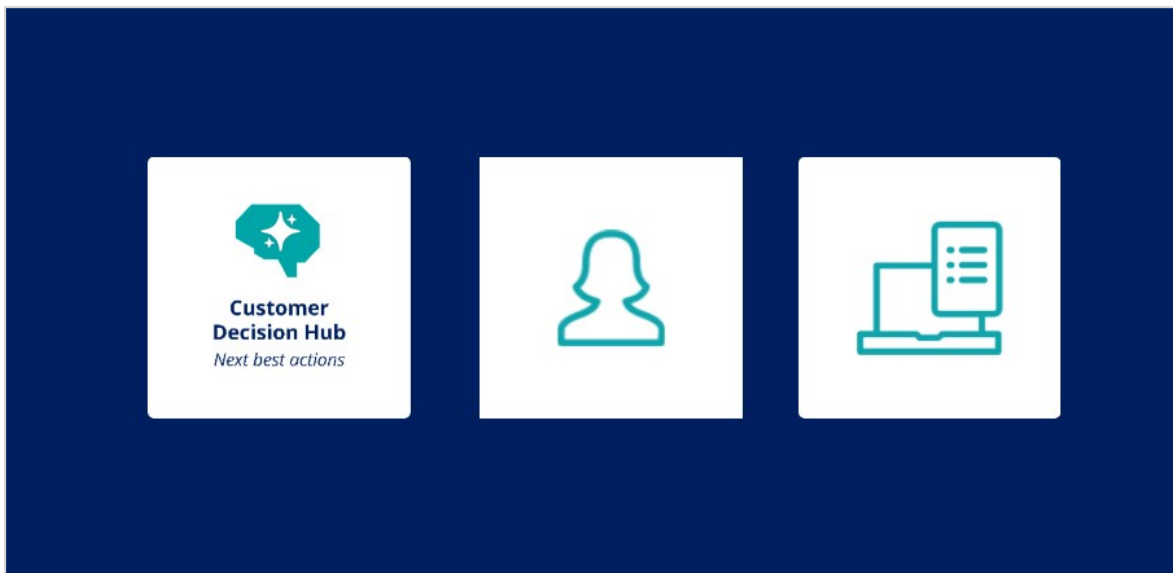
You have reached the end of this video.

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.

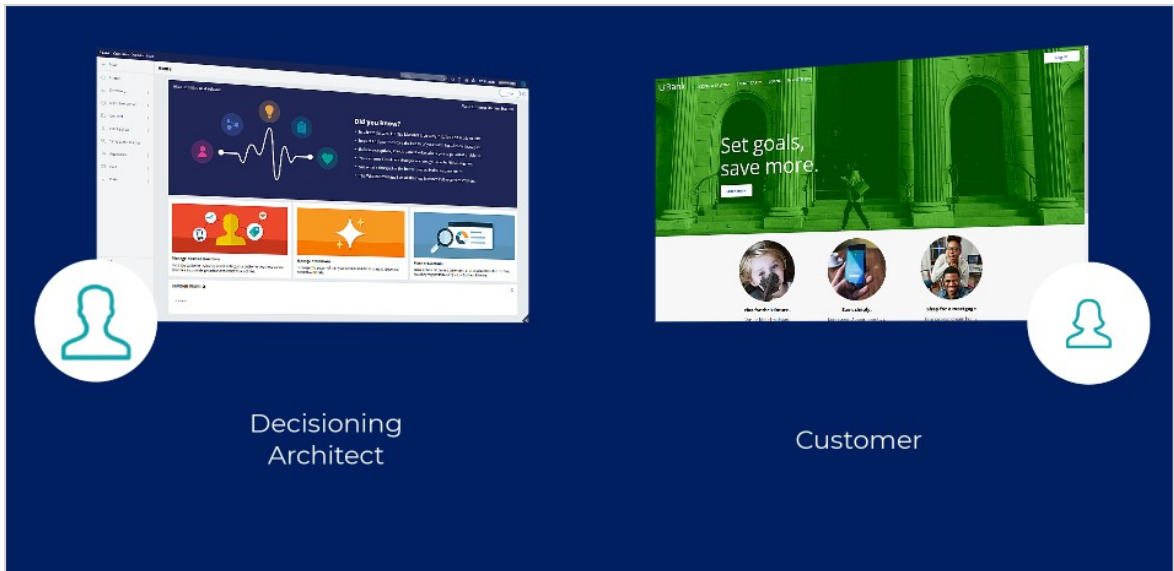
Transcript

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.



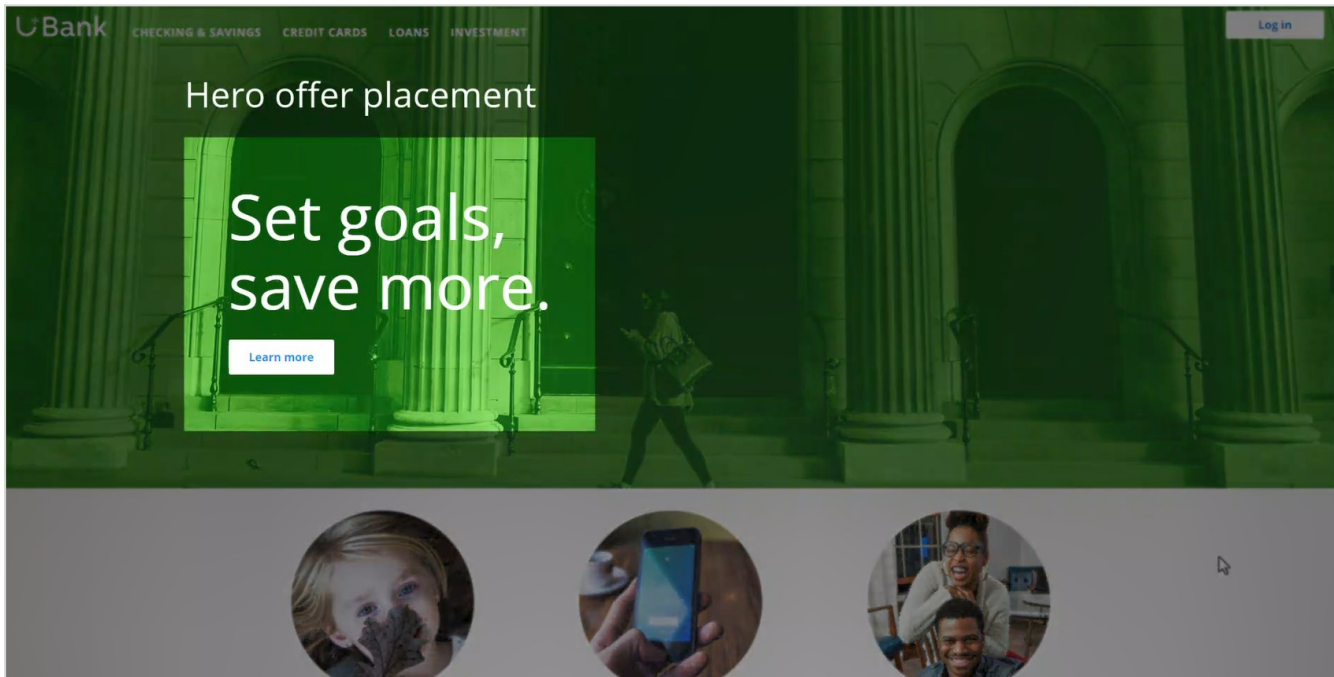
Consider the following web channel scenario, which is a typical cross-selling use case.

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.

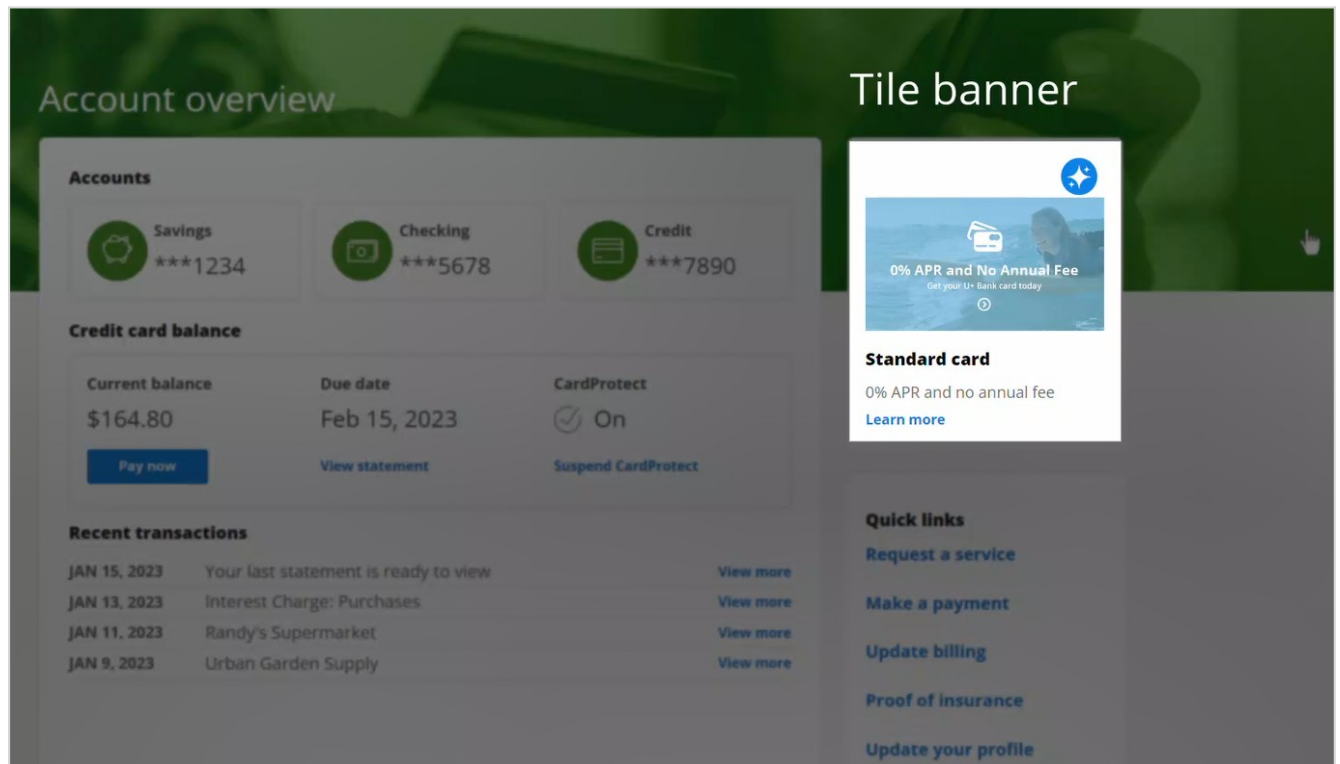


U+ is a retail bank that wants to use its website as a marketing channel to improve One-to-one Customer Engagement, drive sales, and deliver next best actions in real time. The bank has decided to use Pega Customer Decision Hub™ to recommend more relevant banner ads to its customers when they visit the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






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.

Credit cards

Featured offers

card for every occasion

Featured cards

 <p>U+ Bank PLATINUM</p> <p>1234 5678 1234 5678</p> <p>06/21 NAME SURNAME</p>	 <p>U+ Bank CASH REWARDS</p> <p>1234 5678 1234 5678</p> <p>06/21 NAME SURNAME</p>	 <p>U+ Bank TRAVELER</p> <p>1234 5678 1234 5678</p> <p>06/21 NAME SURNAME</p>
<p>Platinum Card</p> <p>Make everyday purchases and earn 4x points on U+ Bank Platinum.</p> <p>Apply now</p> <p>View Details</p>	<p>Cash Rewards Card</p> <p>Get 4% cash back on dining, 3% on gas, and 2% on other purchases.</p> <p>Apply now</p> <p>View Details</p>	<p>Traveler Card</p> <p>Avoid foreign transaction fees, and earn points on airline purchases.</p> <p>Apply now</p> <p>View Details</p>

Cash back you deserved

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.

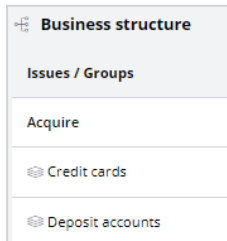
The CDH Implementation team is responsible for implementing a Pega Customer Decision Hub application to meet business requirements, and ensuring that the system is configured to support the business processes. The team typically includes System Architects who are familiar with Pega Platform™ and the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

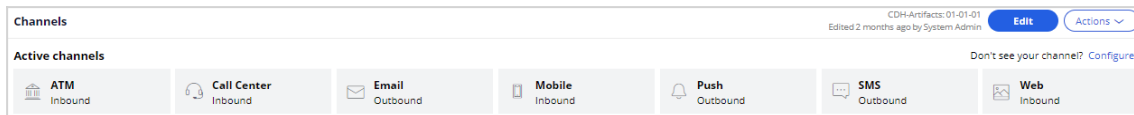
Finally, the Project Governance team manages the project, arbitrates any debates between the different interests among these groups, and ultimately has decision-making authority. It is important to ensure that all perspectives are accounted for, even in organizations with a few people as the main stakeholders. 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.

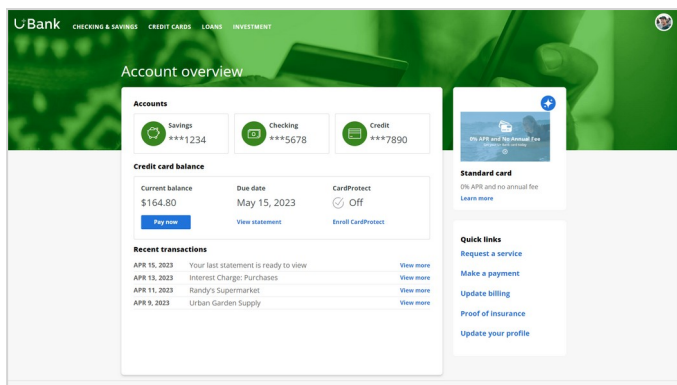


The issue and objectives will also determine which **channels** they use to communicate with customers.



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.



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

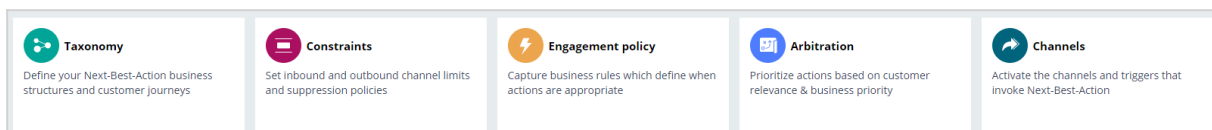
Next-Best-Action Designer organizes the high-level sequence of steps needed to customize the next-best-action strategy to suit the business requirements. Pega Customer Decision Hub™ uses these settings to generate an underlying next-best-action strategy framework and the enterprise-level strategies that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



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.



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.

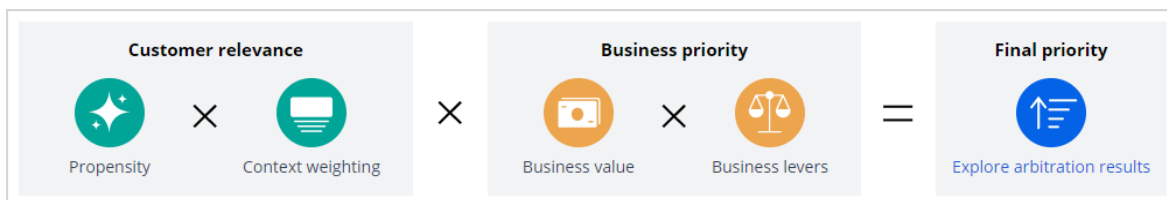
Issue Group
Grow / Credit cards

> 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.

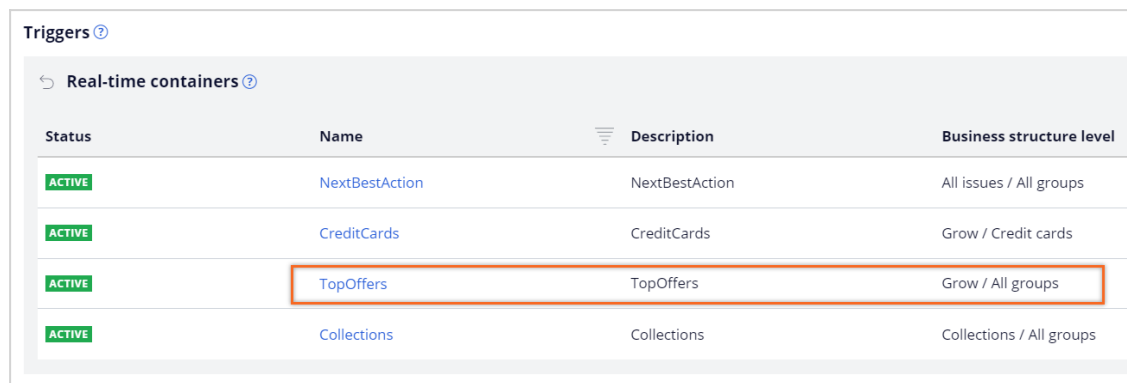
Propensity is the likelihood that a customer responds positively to an action, and AI calculates the propensity. For example, clicking an offer banner or accepting an offer in the contact center are considered positive behaviors.

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

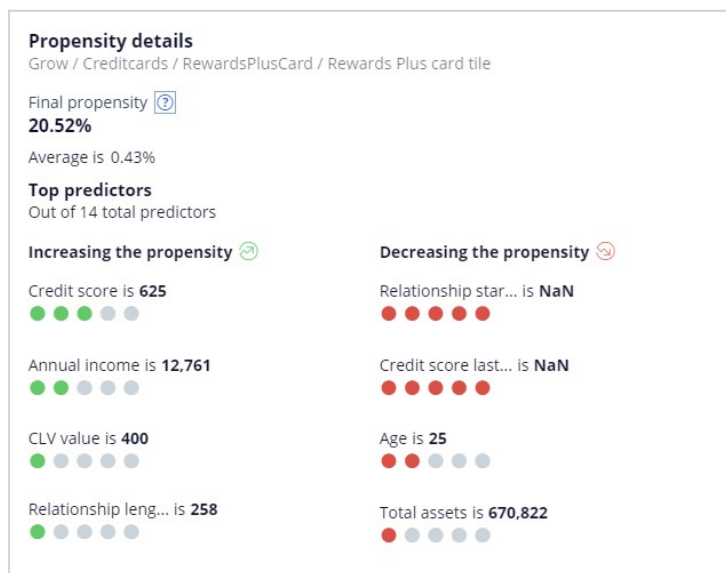
You can explore the arbitration matrix in the Customer Profile Viewer report to understand what actions the system offers to customers. For example, U+ Bank uses Customer Decision Hub to decide which credit card offer to show when customers log in to the bank's website. Joanna is a U+ Bank customer.

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 Premier 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.



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.

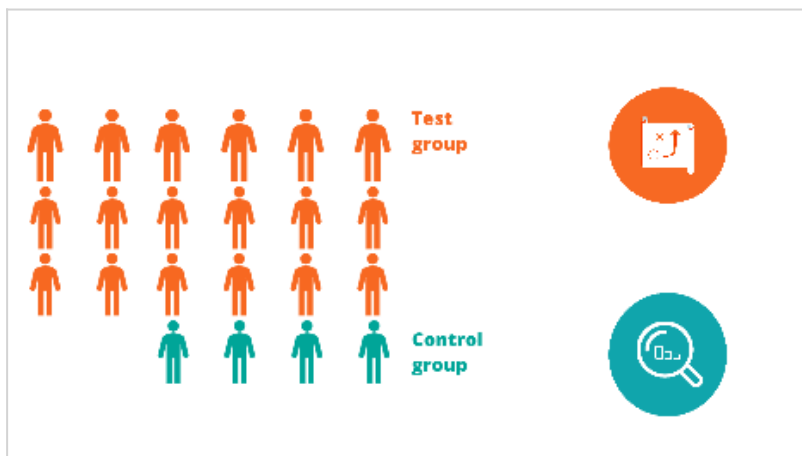
Monitoring the health of the system using Impact Analyzer

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.

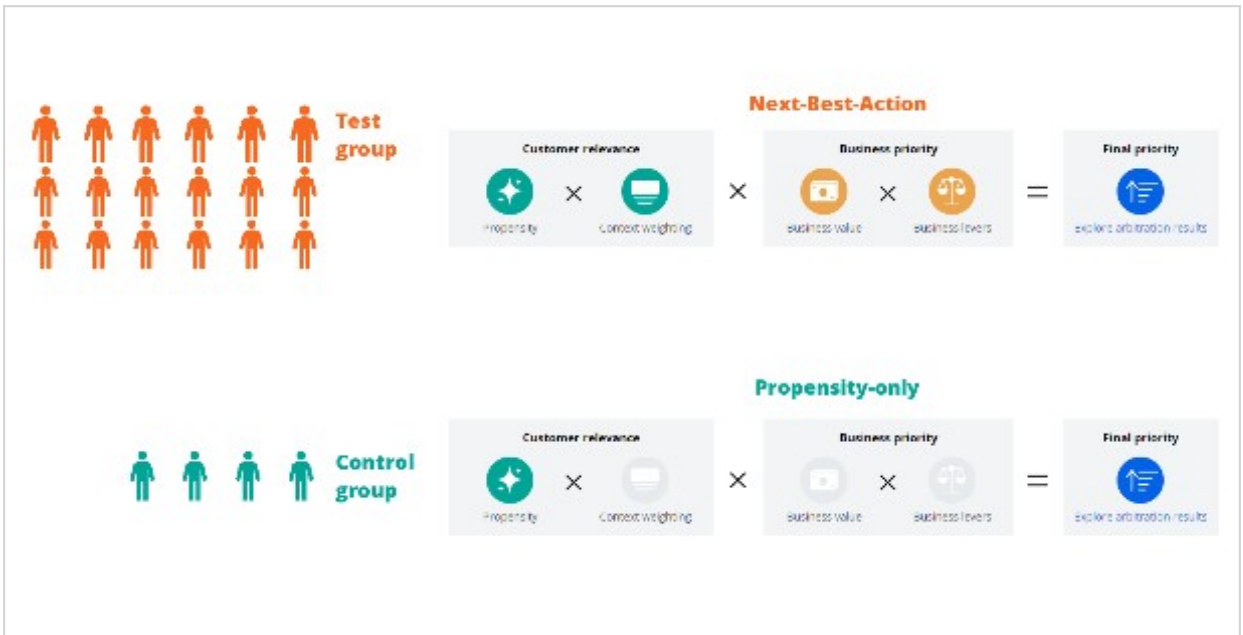
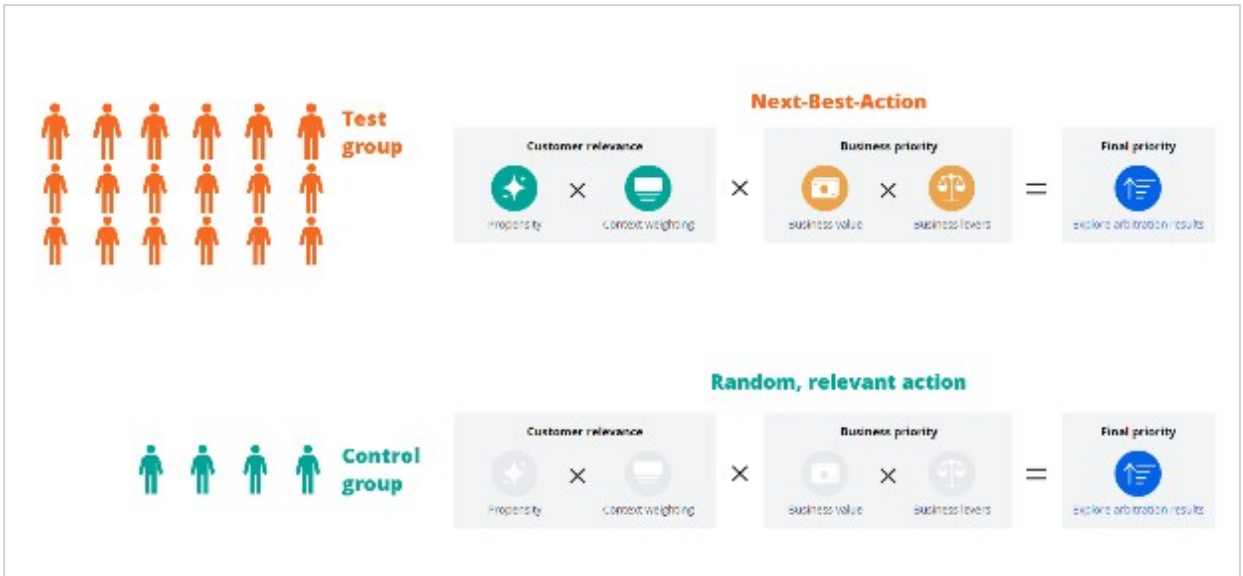
Transcript

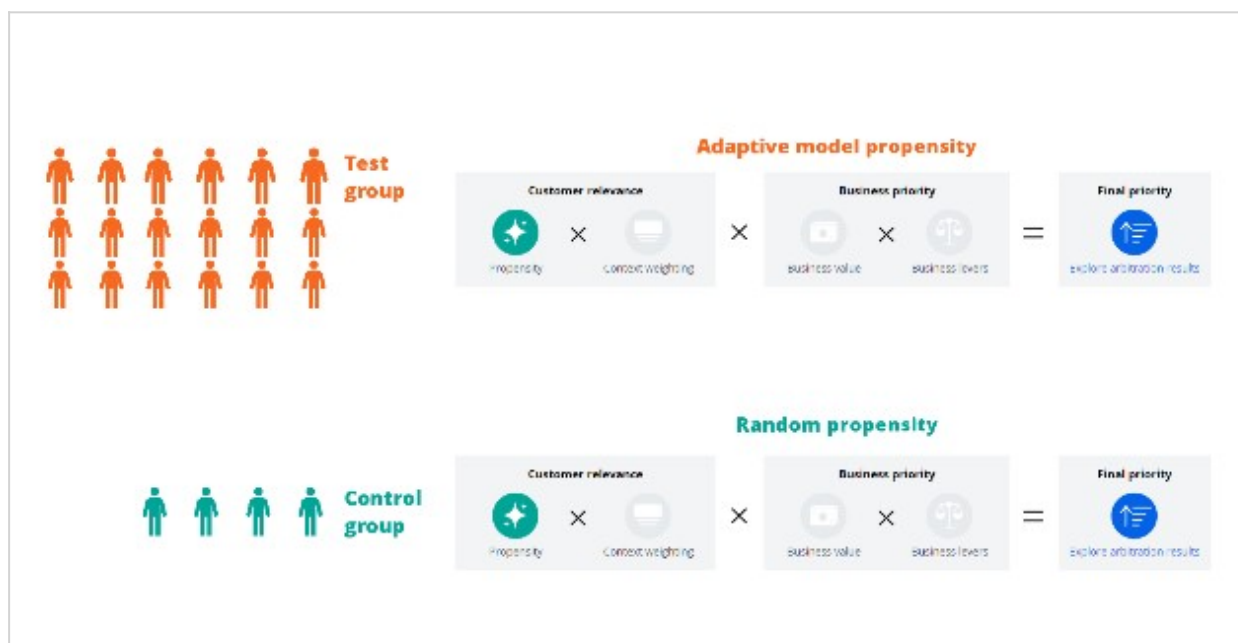
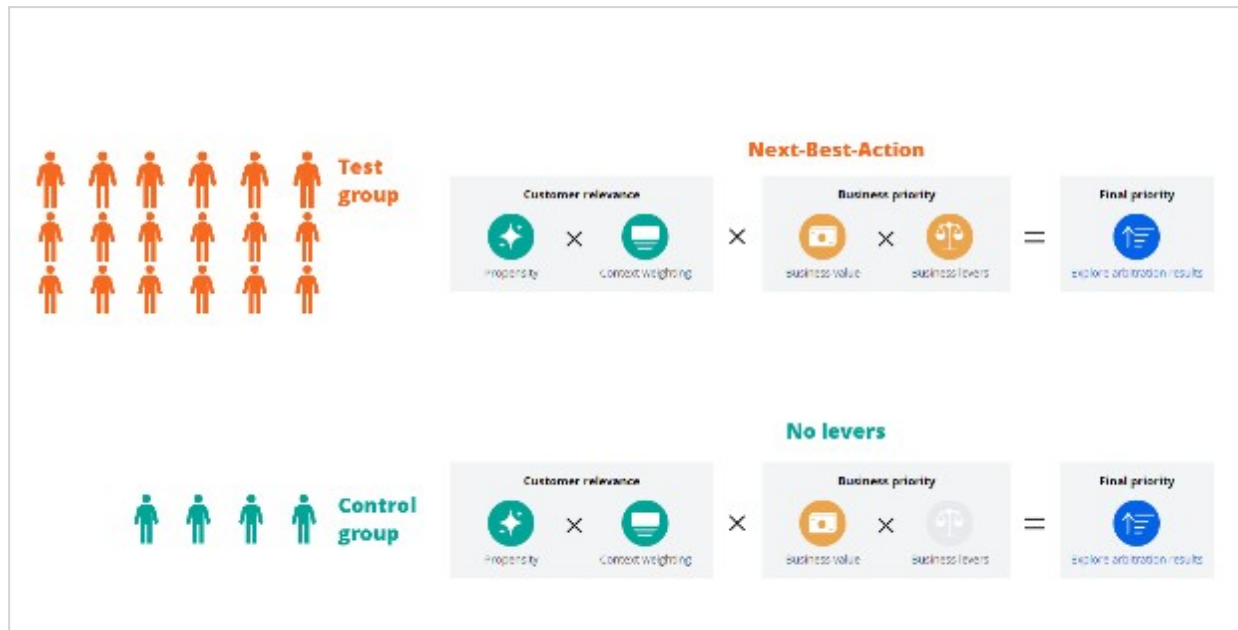
Impact Analyzer is a feature of Pega Customer Decision Hub™ that uses experimentation to track the effectiveness and overall health of next best actions. By utilizing this tool, you can gain insights into the effects of arbitration. Impact Analyzer conducts experiments with alternative prioritization 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.

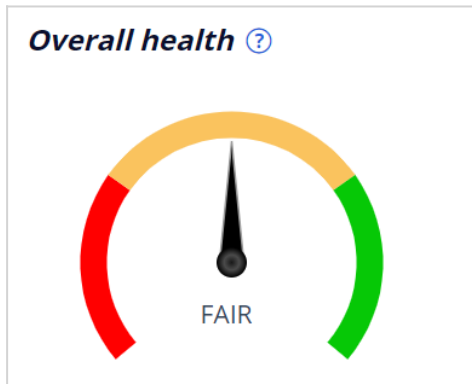




Impact analyzer continuously monitors the health of the next best actions in the experiment, and presents the results in the form of two key performance indicators (KPIs): value lift, and engagement lift. Value lift measures how much a business can make from suggesting certain next best actions to customers, 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

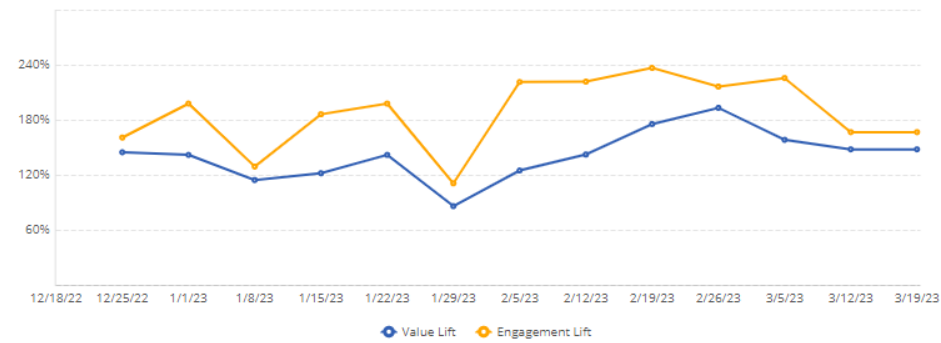
HIGH CONFIDENCE

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*)

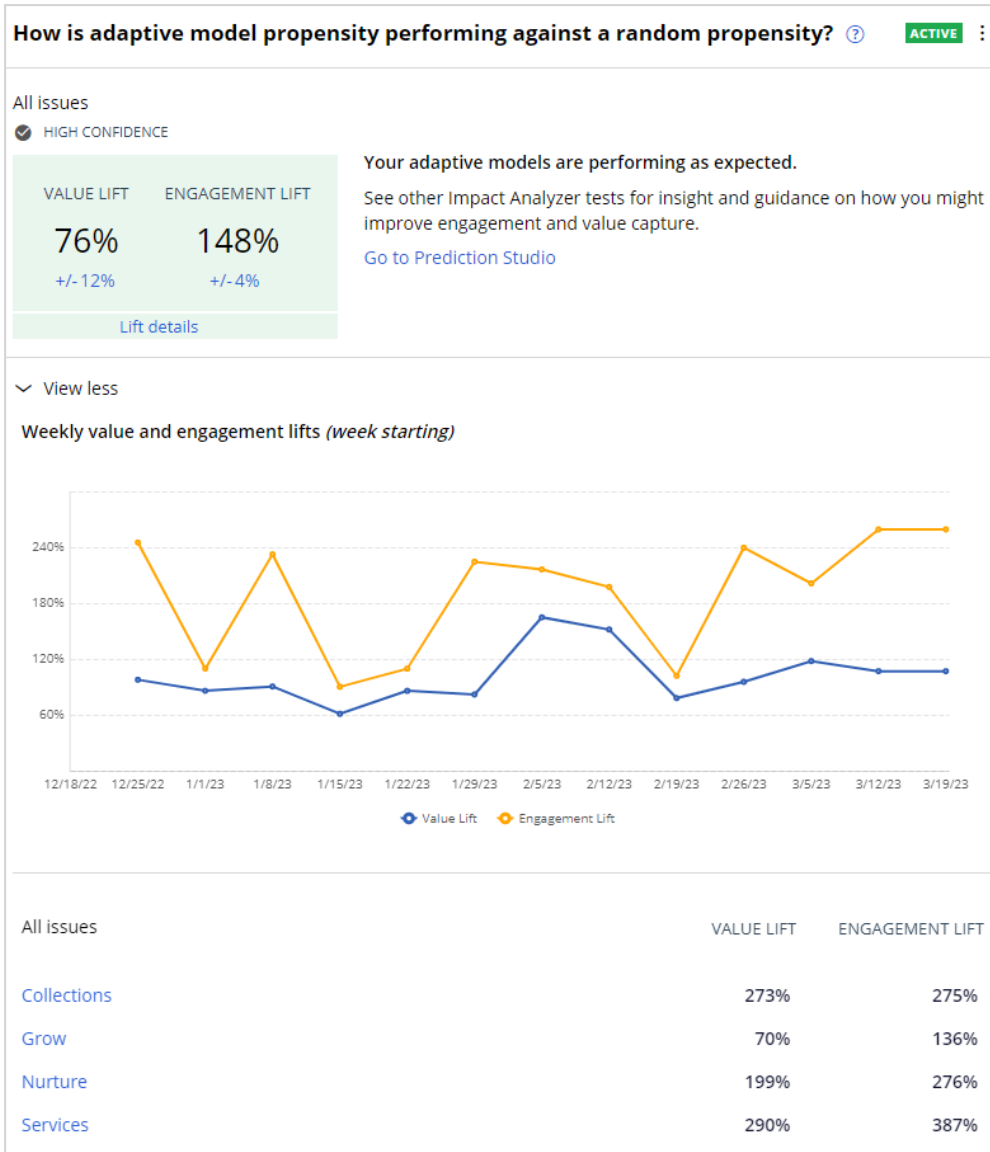


All issues	VALUE LIFT	ENGAGEMENT LIFT
Collections	251%	254%
Grow	121%	156%
Nurture	171%	215%
Services	335%	335%

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?*



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.

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.

Customer Decision Hub predictions

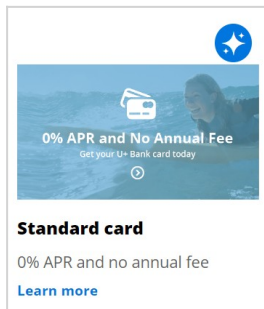
Prediction Studio is the workspace in which you manage the predictions that Pega Customer Decision Hub™ uses to optimize customer interactions.

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

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 ⓘ	Negative outcome ⓘ
<input type="button" value="Add outcome"/>	<input type="button" value="Add outcome"/>
<input type="text" value="Clicked"/>	<input type="text" value="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!

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, which helps businesses to implement adaptive models for predicting customer behavior, such as interacting with online offers. By continuously learning from customer responses, ADM 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.

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.

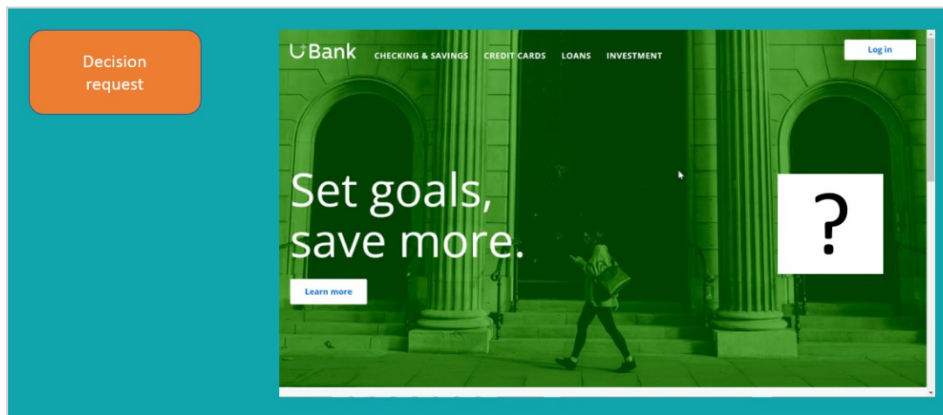
Adaptive analytics

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.

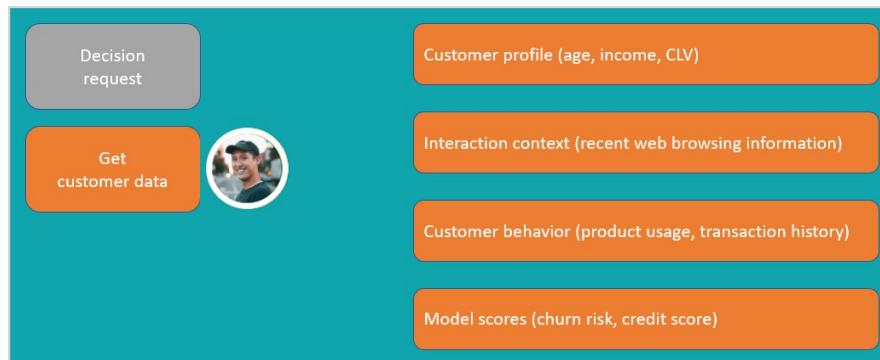
Transcript

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.

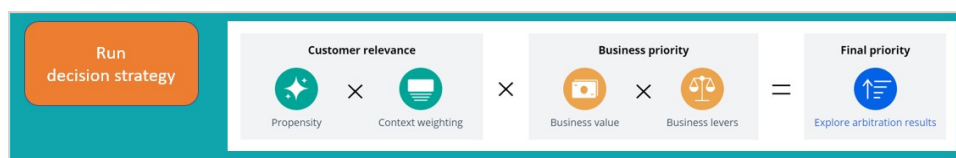
Adaptive Decision Manager (ADM) is a component of Pega Decision Management that businesses can use to implement online, adaptive models that drive predictions about customer behavior, like clicking or ignoring a web banner that offers a credit card on a bank's website. When customer Troy logs in to the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



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.

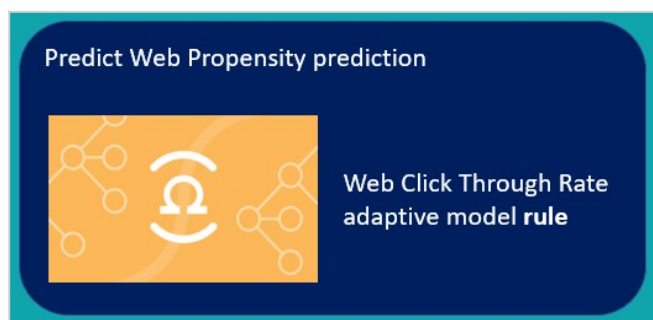


The result of the decision strategy, the Next-Best-Action for customer Troy, is the Standard Card, which is then displayed on the website. There are two possible outcomes of the interaction.



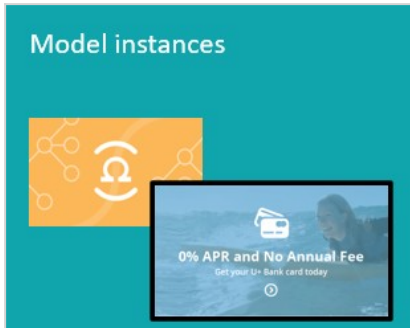
If Troy is interested and clicks on the web banner, ADM records the outcome as target behavior. If he ignores the banner, ADM records the outcome as alternative behavior. The prediction predicts the probability that a customer shows the target behavior.

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.

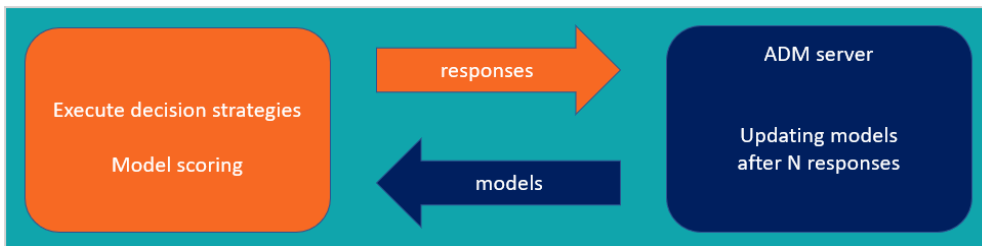


A data scientist configures the settings of both the prediction and the adaptive model rule, including the customer fields that are available as features to the model rule. Customer fields that are unsuitable as features, for example the customer ID, should be excluded.

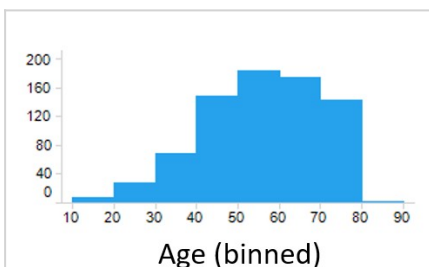
An adaptive model rule typically generates many adaptive model instances without human intervention, because each unique combination of an action, treatment, direction, and channel, will generate a model the first time a decision strategy runs that references the model.



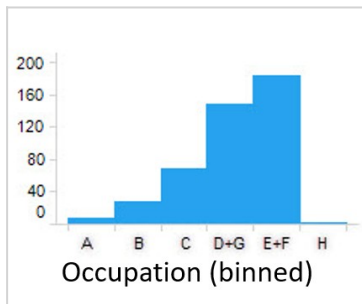
ADM captures the responses and updates the adaptive model instances regularly, so they continuously learn from customer responses and adapt to account for changing customer interests and needs. The ADM server is physically separated from the nodes that process decision strategies and model executions,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



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.

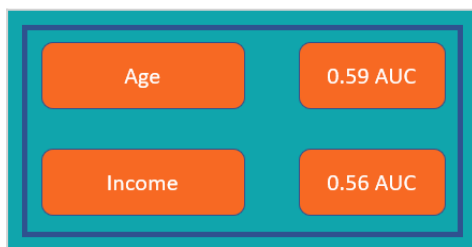


The granularity of the binning is a trade-off between performance and the statistical robustness of the predictor. Appropriate default settings for binning are provided and should only be changed by an experienced data scientist for specific use cases.

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.

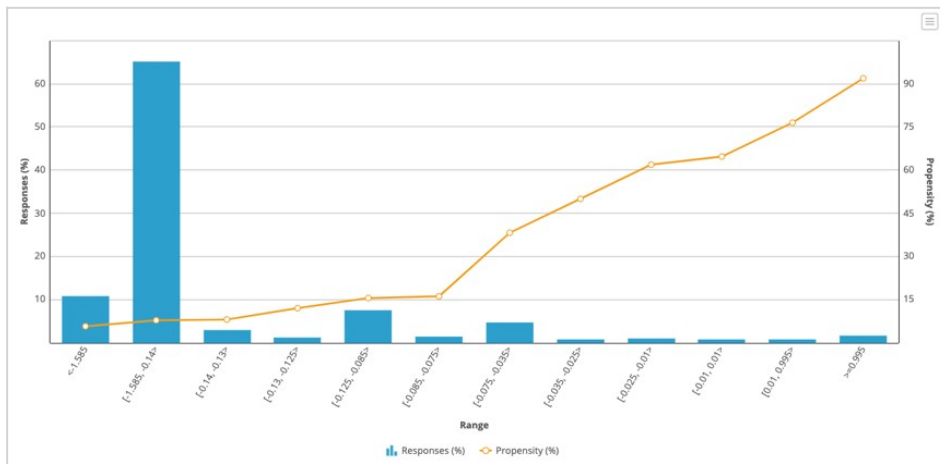
$$P(A|B) = (P(B|A) \times P(A))/P(B)$$

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.

$$\text{Score (Log odds)} = \ln(P/(1-P))$$

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.

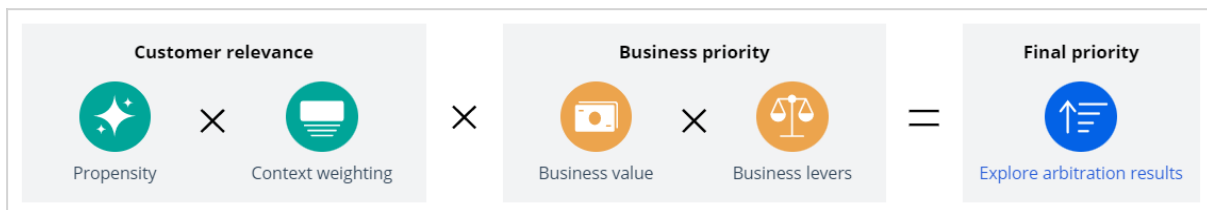
Action arbitration with AI

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

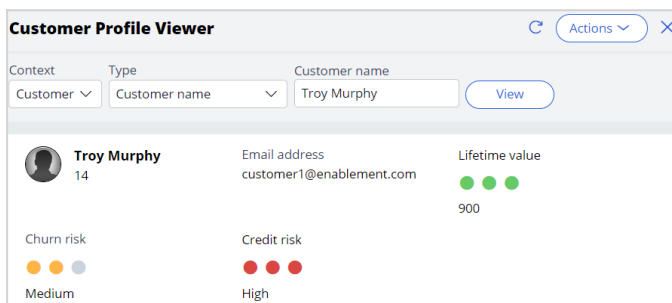
U+ Bank wants to optimize the cross-selling of their credit cards on the web by using Pega Customer Decision Hub to show a credit card offer in a web banner when a customer logs in to their account. Customer Decision Hub uses the Predict Web Propensity prediction to arbitrate between the offers for which 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.



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

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 Time unit

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.

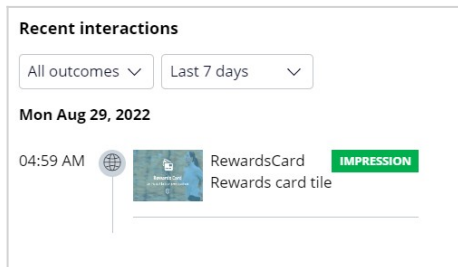
Every hour, Adaptive Decision Manager updates the models to learn from the recent interactions.

Action	Direction	Channel	Adaptive model rule	Model technique	Last updated
{"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.



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.

For example, data that directly or indirectly reveal a customer's geographic location, such as the ID of the customer's most-used ATM or their zip code, can be a proxy for off-limits data such as income.

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 do you have any other recommendations on the predictors used by adaptive models?

Well, my advice is to document critical information about your predictors.

You can then better understand the predictors that you are using and make informed decisions about which predictors to include in your models.

This information should include:

- The source of the predictor data
- The type of predictor: is it regular, parameterized, or based on interaction history summaries?
- The business meaning of the predictor
- The data type of the predictor
- The range of values that the predictor can take
- And any implementation notes, especially whether empty values are allowed.

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.

Also, I need to document my candidate predictors to understand their value.

Yes, that sums it up!

Thanks, Bella! This has been very helpful.

My pleasure.

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.

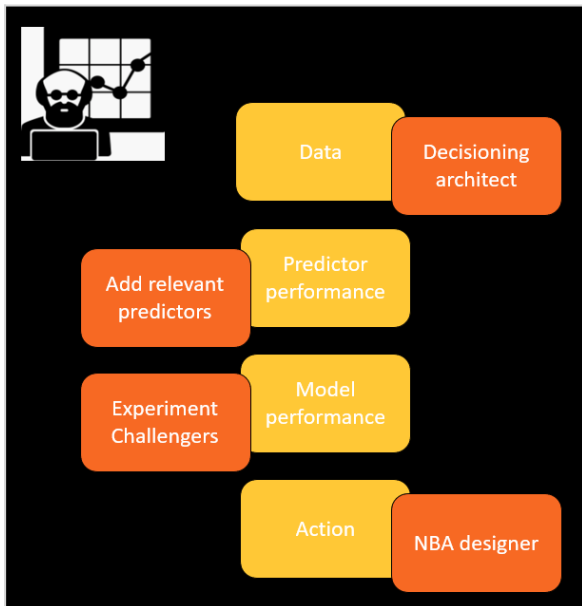
Monitoring predictions

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.

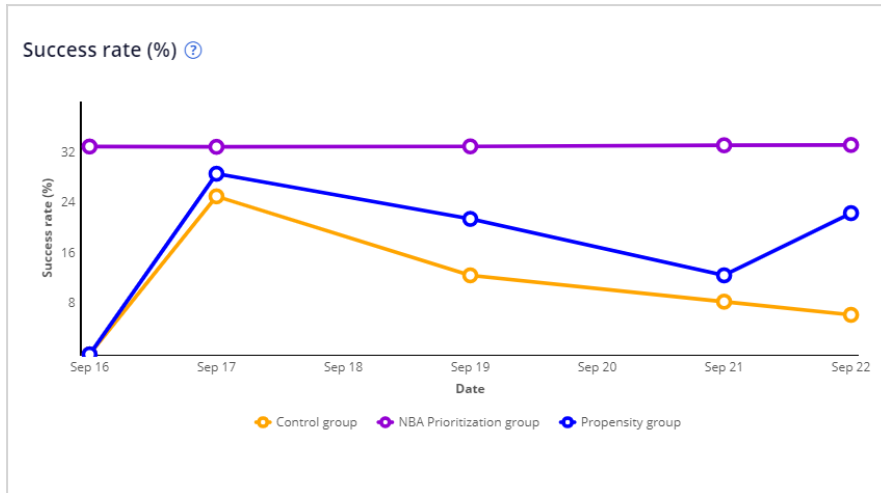
Transcript

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.

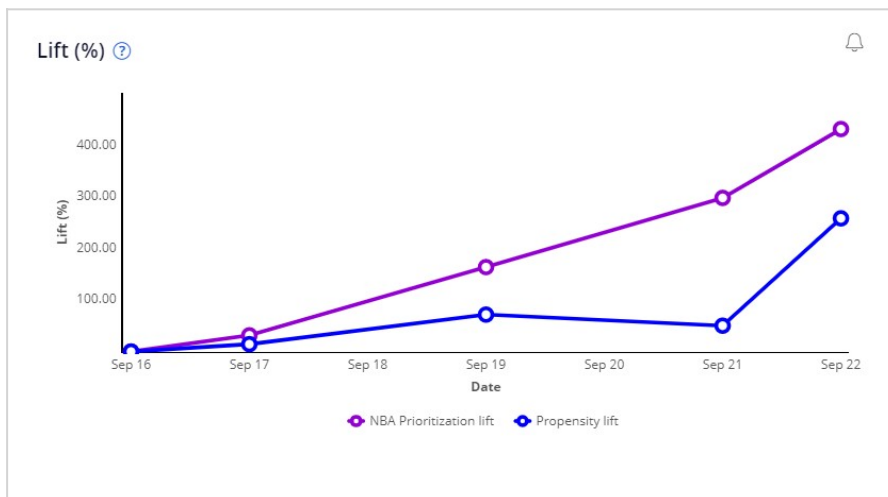


For the analysis of predictions, several charts are available. As an example, U+Bank uses the *Predict Web Propensity* prediction to calculate the likelihood that a customer will click on a web banner that offers one of the credit cards for which the customer is eligible. Success rate measures how successful 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.

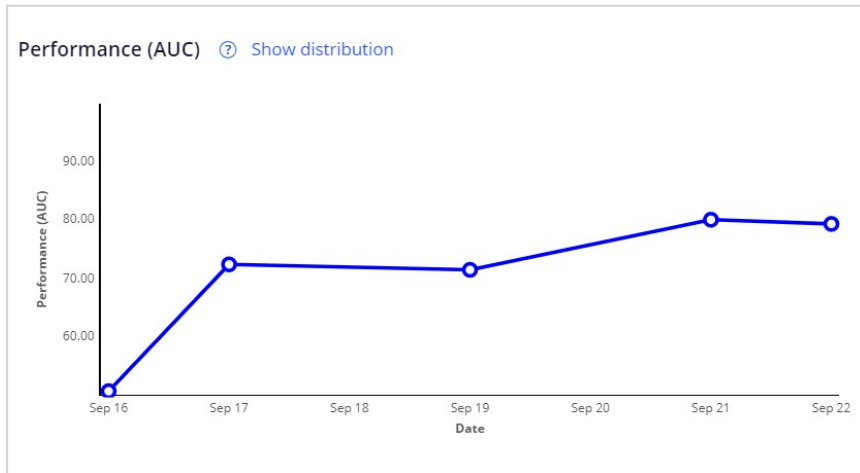
Lift is the impact on engagement that the AI generates. The lift metric can tell us how much better predictions made by a model are when compared to a series of random predictions for customers in the control group.



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

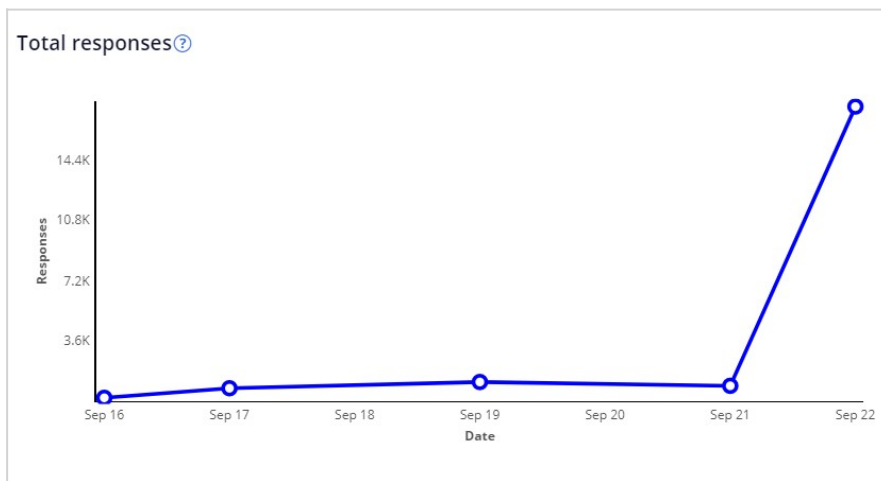
if the model is still immature, or if the control group is not representative enough of the actual population.

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



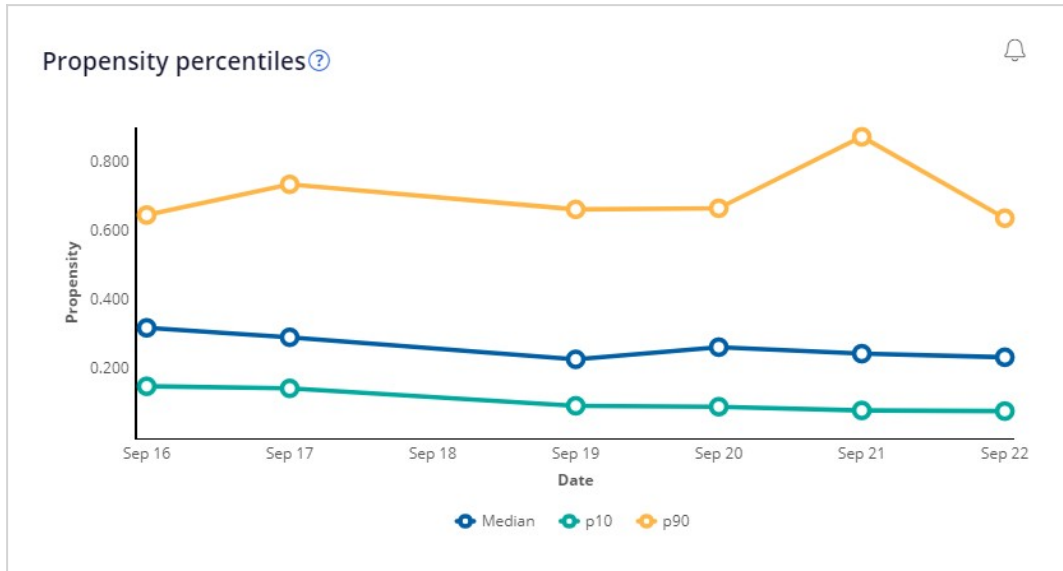
The prediction needs attention when the performance decreases over time. This may indicate issues with the predictor set of the models.

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



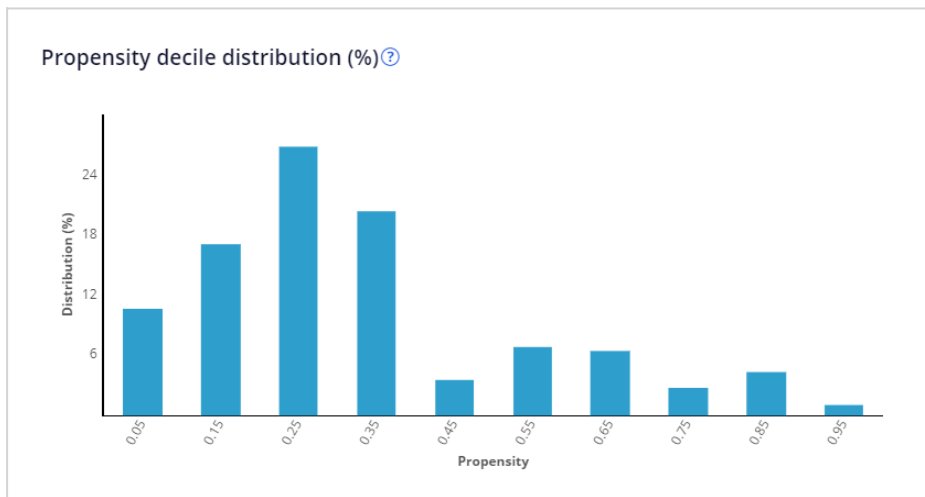
The prediction needs attention when the number of responses received is zero for a week or more. This may have multiple causes and needs to be looked at by the Decisioning Architect.

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).

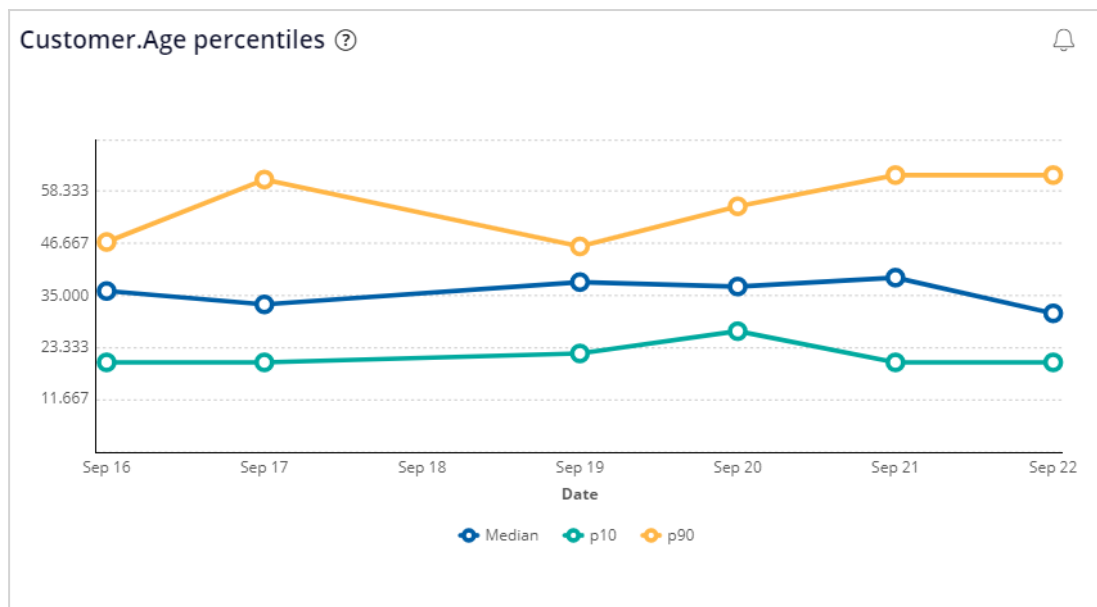


For inbound use cases, the propensities peak at higher values than outbound use cases and continue reducing over the higher value deciles, indicating that models are more confident in making inbound predictions than outbound predictions. The prediction needs attention when the propensity is very similar 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.

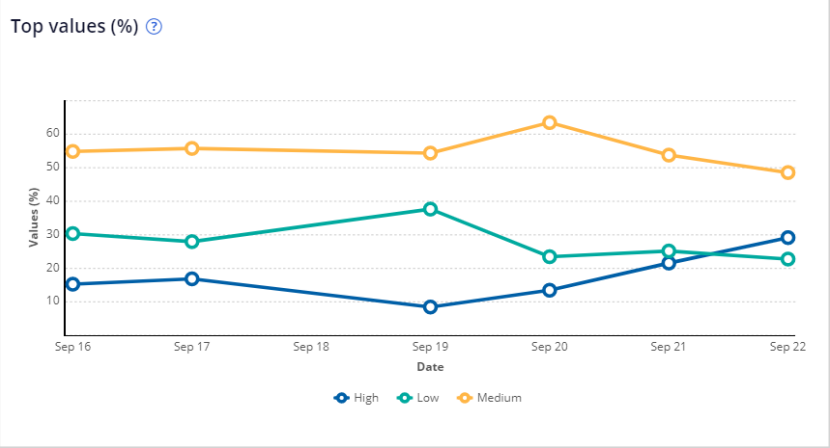
Best performing models					
Issue	Group	Action	Direction	Channel	Performance (AUC)
Grow	Creditcards	{"pyName":"PremierRewardsCard","pyTreatment":"Premier Rewards card tile"}	Inbound	Web	83.981
Grow	Creditcards	{"pyName":"RewardsPlusCard","pyTreatment":"Rewards Plus card tile"}	Inbound	Web	75.271
Grow	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.

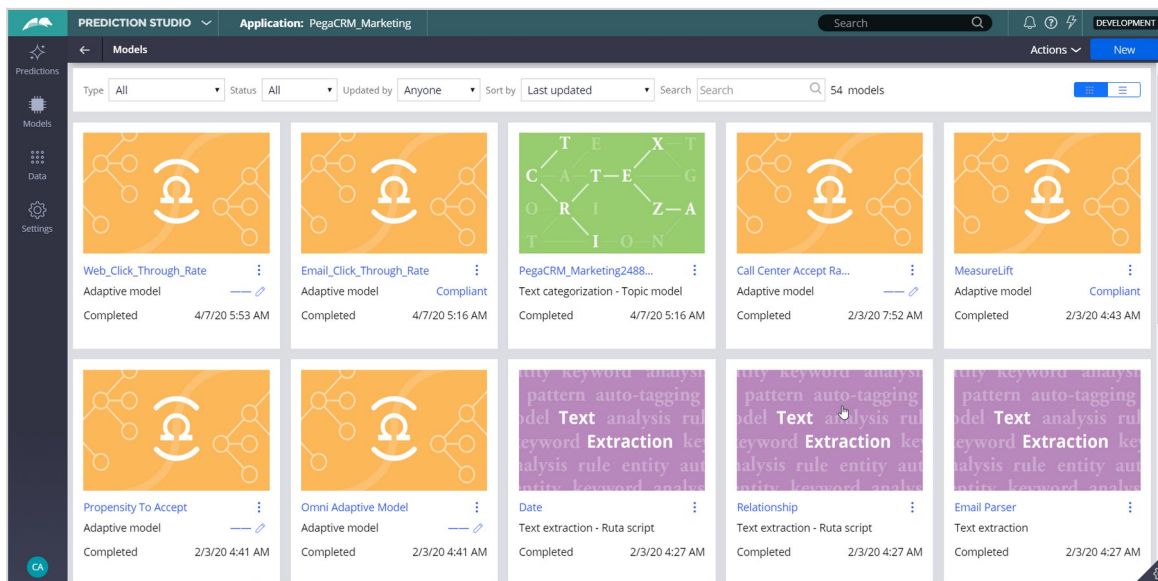
Monitoring adaptive models

Introduction

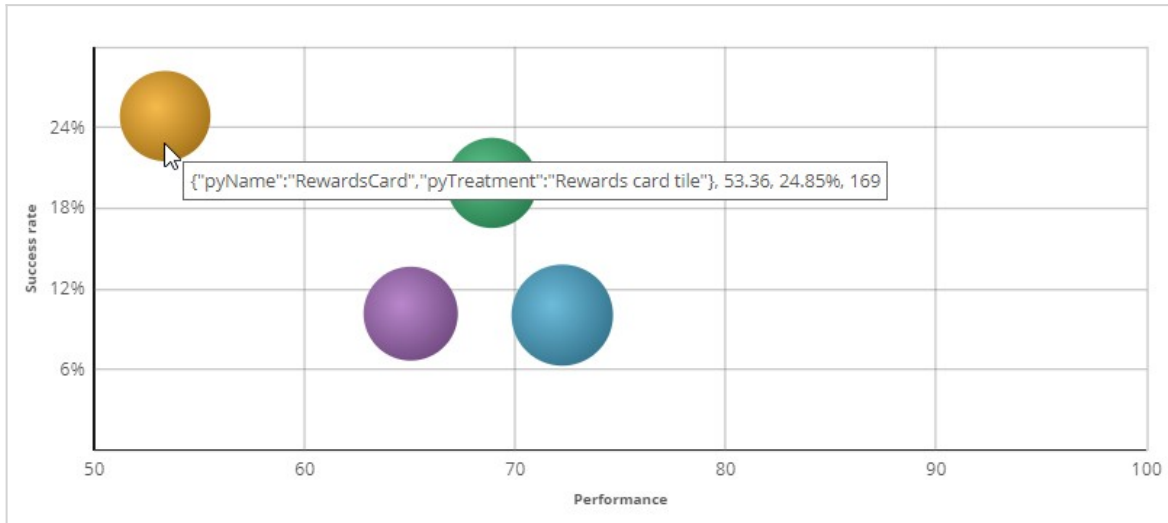
The predictive performance and success rate of individual adaptive models provide information that can help business users and decisioning consultants to refine the Next-Best-Actions of the company. Monitoring the health of adaptive models and their predictors is a regular data scientist task that can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

Transcript

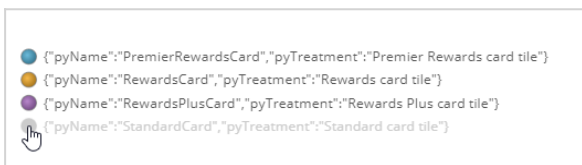
This demo will show how to inspect the health of your adaptive models and their predictors. This is a regular data scientist task. The predictive performance and success rate of individual adaptive models provide information that can help business users and decisioning consultants to refine the Next-Best-Actions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



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.

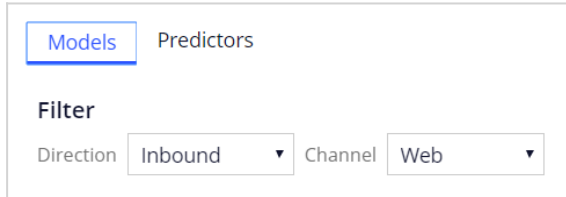


The Performance axis indicates the accuracy of the outcome prediction. The model performance is expressed in the Area Under the Curve (AUC) unit of measurement, which has a range between 50 and 100. The higher the AUC, the better a model is at predicting the outcome.

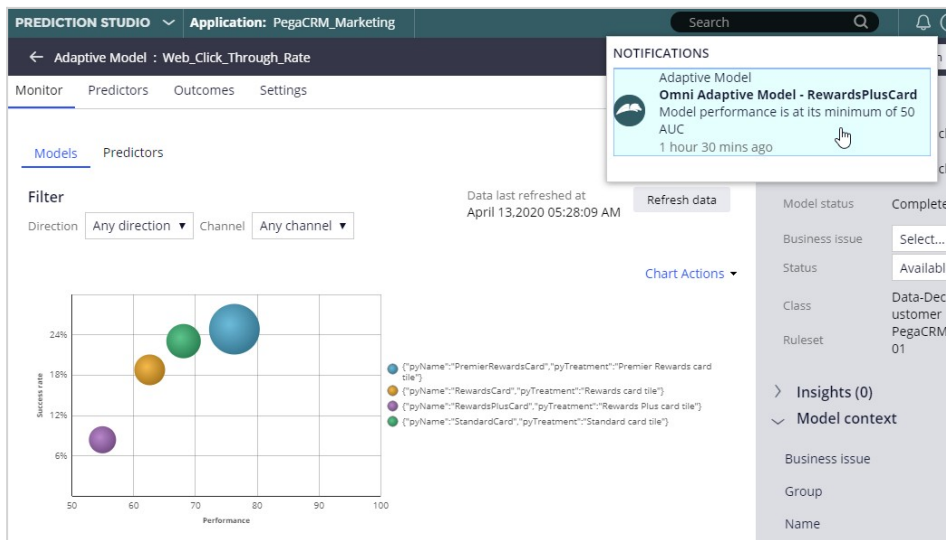
The Success rate axis indicates the success rate expressed in percentages. In this example, the success rate represents how often a web banner is clicked. The system calculates this rate by dividing the number of times a banner is clicked by the total number of times the banner was shown on the website.

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.



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



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	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.px.LastOutcomeTime.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

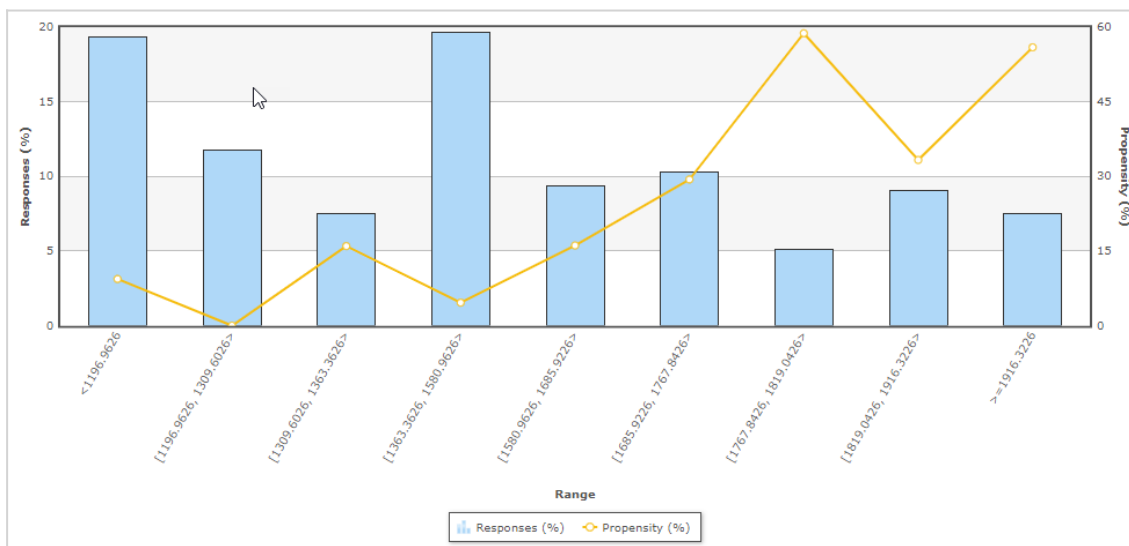
AUC

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.

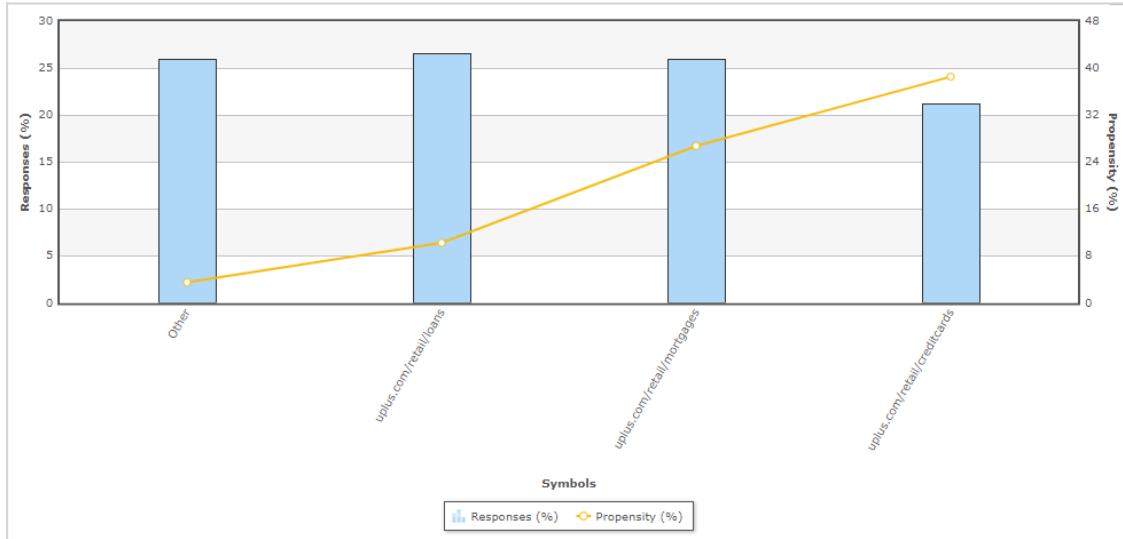
From the adaptive model table, you can drill down into a model report for a specific adaptive model.

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

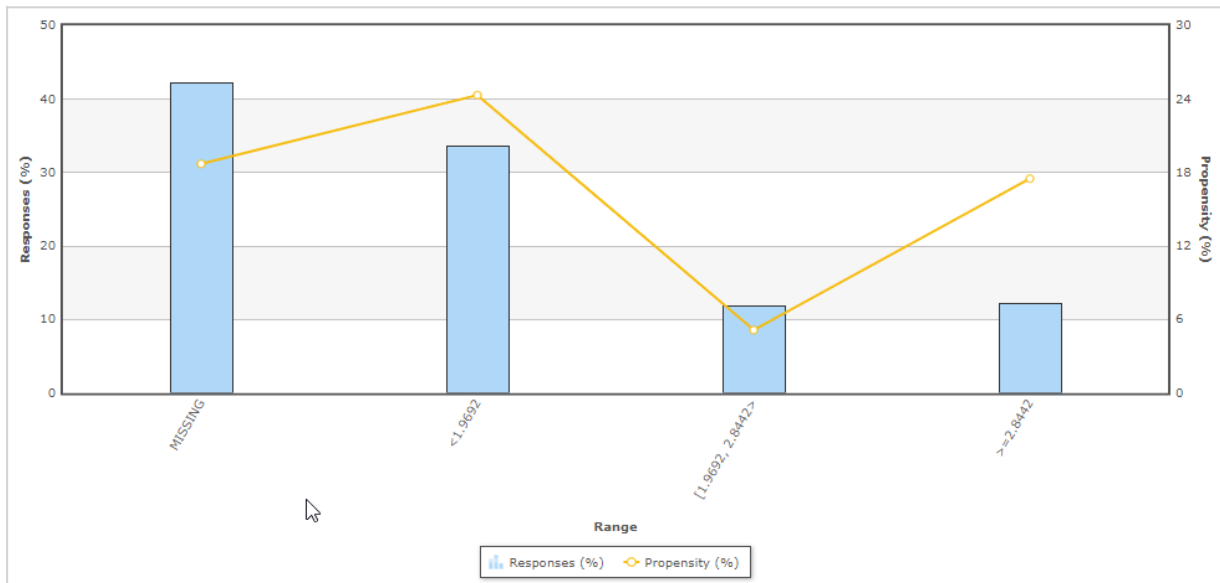
There are three tabs, reporting on predictors, the model score distribution, and the trend. In the predictors report, you can examine the performance of individual predictors. Let's examine a couple of them. In this case, the best-performing predictor is AverageSpent. This is a predictor of type numeric. 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.



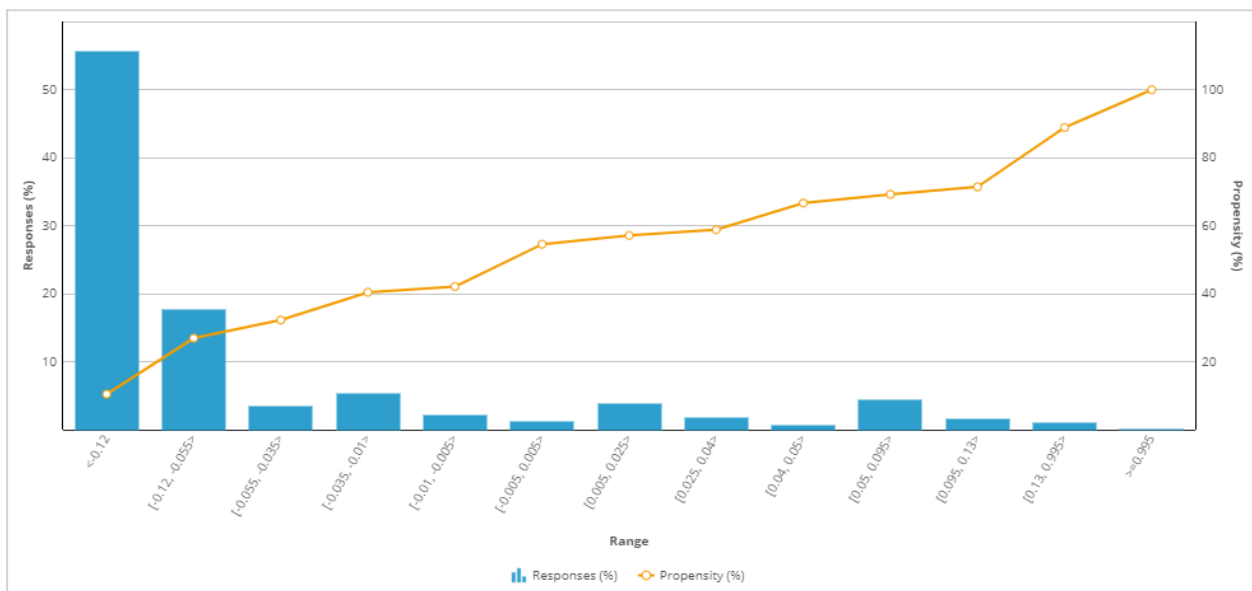
To further improve the predictive power of the models the system uses Interaction History summaries. In this example, the adaptive system established that the number of days since the offer was accepted is a well-performing predictor.



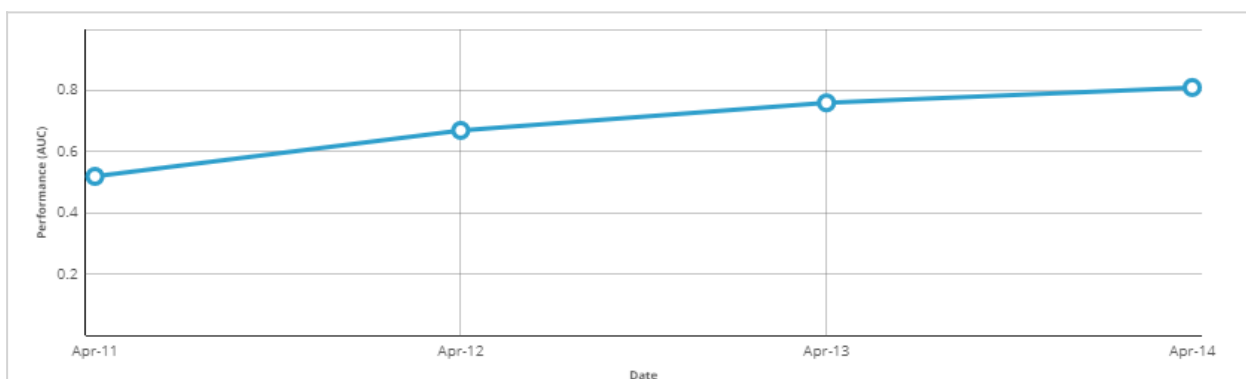
The system grouped three predictors that are correlated. It then marked two of them as inactive. Inactive predictors are not used in the propensity calculation.

▼ 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.

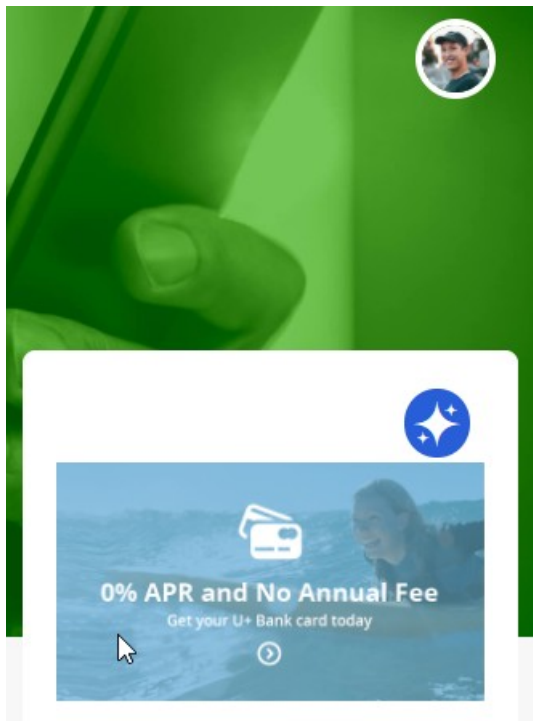
Exporting historical data

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.

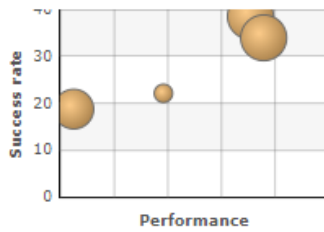
U+ Bank has implemented Pega Customer Decision Hub™ to show a personalized banner on their website that advertises credit card offers.



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.



[Web_Click_Through_Rate](#) ⋮

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 <input type="text" value="100.00%"/>
NoResponse, Impression	Sample percentage <input type="text" value="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.

Select repository type



Artifactory



S3



File system



Azure

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

New data set



Name *

Type *

Apply to *

Development branch

Add to ruleset

Ruleset version

Cancel

Create

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


File Mapping

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  Preview file

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

 Save as  Delete Actions 

File Mapping Specifications History

Data source

Files on repository
 Embedded file

- Run
- Refresh
- Delegate
- Add to favorites
- Export** 
- Import

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.

The screenshot shows a software interface with a 'Fields (7)' list on the left and a 'Details' panel on the right. The 'Details' panel displays a JSON object for 'Web_Click_Through_Rate'. The 'Customer.Age' field is highlighted in red in the field list, and its value '25.0' is highlighted in red in the JSON output.

```

"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

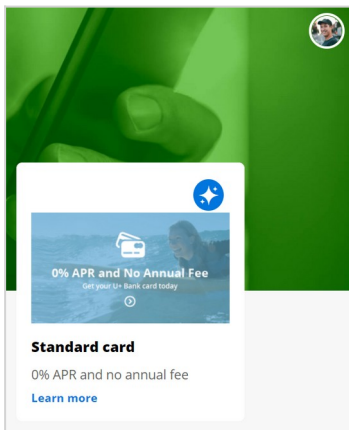
Data scientists can export snapshots of the adaptive models that drive Pega Customer Decision Hub™ predictions from the Adaptive Decision Manager (ADM) data mart for further analysis in their favorite analytical tools.

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.

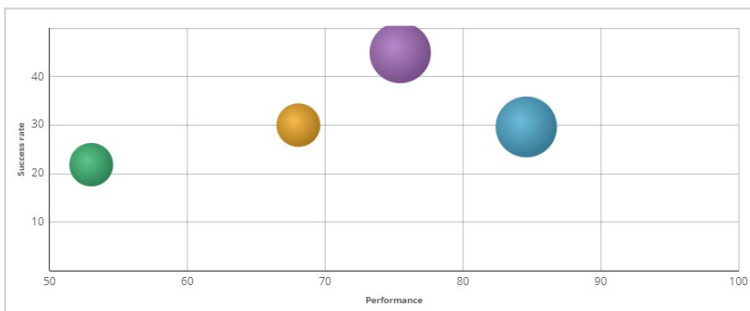
Transcript

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.



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.

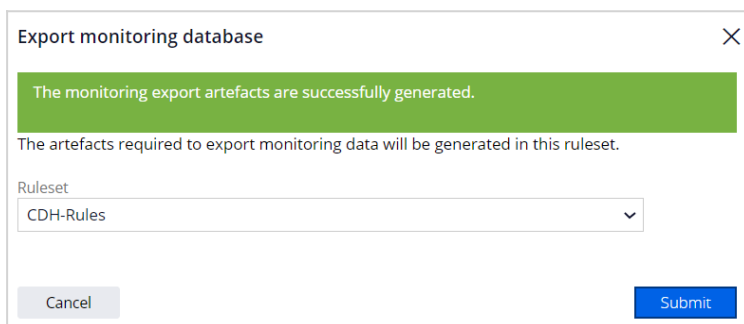
Snapshot Time	Model ID	Model Name	Configuration Name	Predictors	Performance	Etc.
20202212 11.49	5c2d8678-660c	Web Treatment A	Web Click Through Rate	153	0.74	
20202212 11.49	6h4s9812-546d	Email Treatment C	Email Click Through Rate	68	0.63	
20202212 11.49	7f3g3401-841a	Retail Treatment B	Retail Click 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.

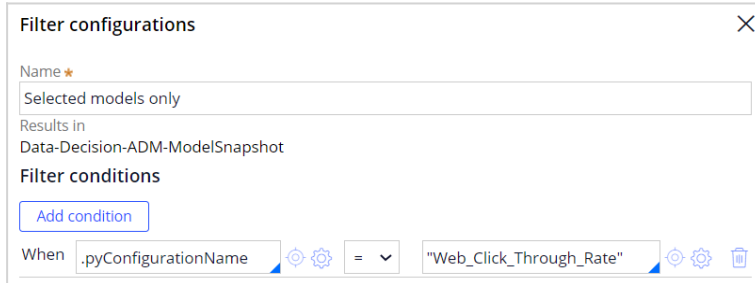
In the implementation phase of the project, you generate the artifacts required for the export of adaptive and predictive model snapshots, including the required data flows in Prediction Studio.



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*.



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", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.517635", "pyResponseCount": "416.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.518584", "pyResponseCount": "783.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.520264", "pyResponseCount": "988.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"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": "CDH", "pyPerformance": "0.500000", "pyResponseCount": "1.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.503333", "pyResponseCount": "116.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.642275", "pyResponseCount": "606.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.659855", "pyResponseCount": "683.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.657353", "pyResponseCount": "989.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"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", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.673089", "pyResponseCount": "571.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"pxApplication": "CDH", "pyPerformance": "0.714575", "pyResponseCount": "889.0", "pyPerformanceError": "0.0", "pyConfigurationName": "Web_Click_Through_Rate", "pyFactoryUpdateI
"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?

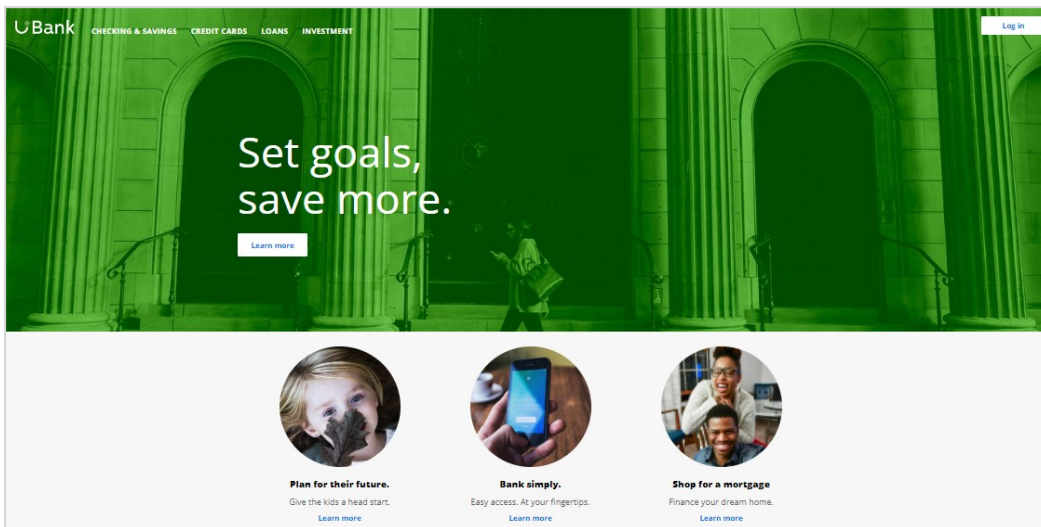
- How to generate the monitoring database export artifacts.
- How to configure the auto-generated data flows to export a subset of the ADM data mart.
- How to trigger the monitoring database export.

Data scientist tools for Customer Decision Hub

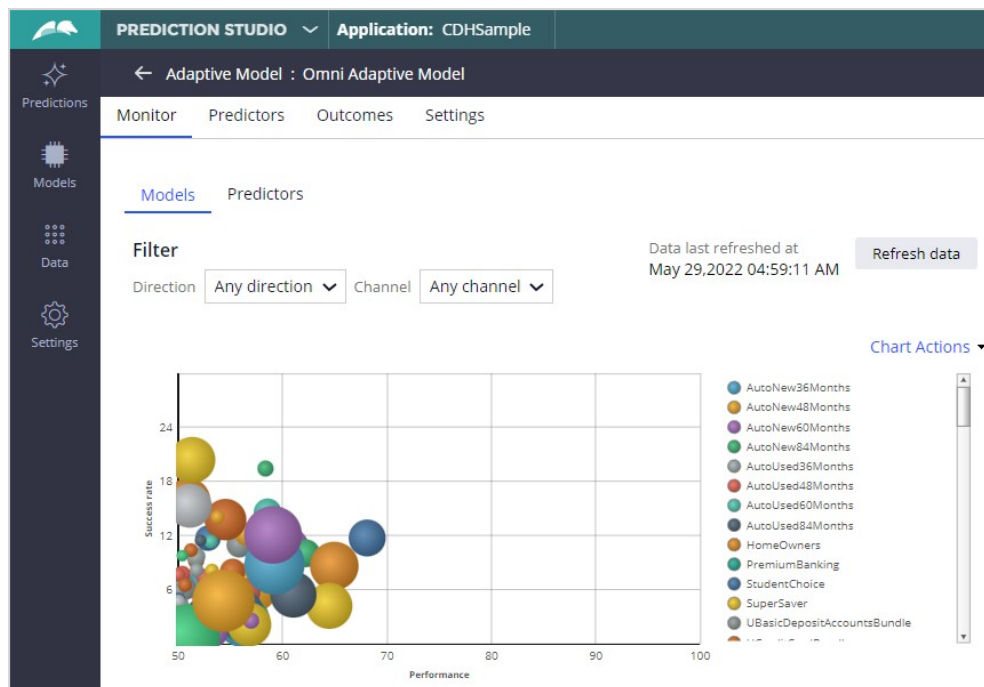
Pega has an open data model. Consequently, data scientists can export adaptive model data, predictor binning data, and historical data for further analysis. The open-source GitHub repository [Pega Data Scientist Tools](#) helps you to build meaningful plots and more.

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.



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.

```

M pip install pdstools
...
M 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.

```

M pip install pdstools
...
M from pdstools import ADMDatamart
M 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.

```

M CDHSample.combinedData

```

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

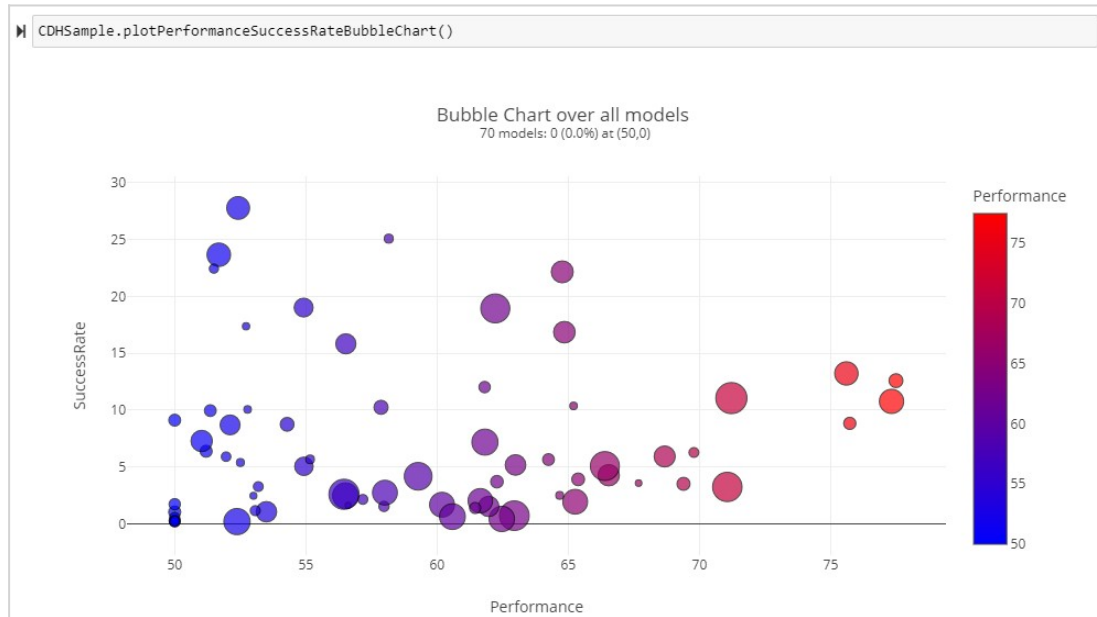
```

M CDHSample.AvailableVisualisations

```

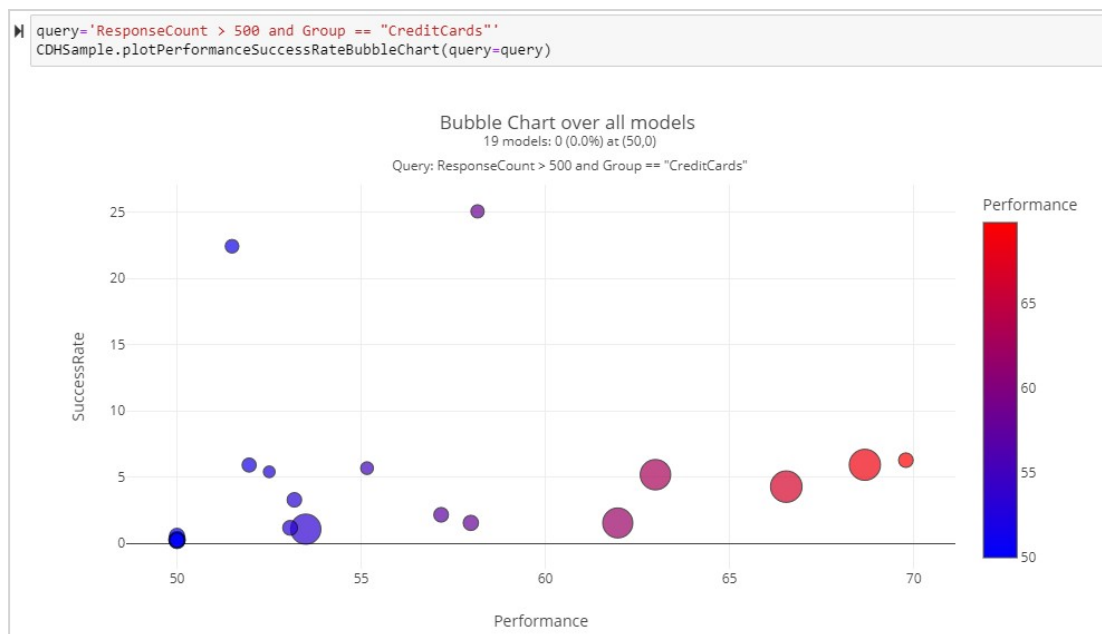
	modelData	predictorData	Multiple snapshots
plotPerformanceSuccessRateBubbleChart	1	0	0
plotPerformanceAnd SuccessRateOverTime	1	0	1
plotOverTime	1	0	1
plotResponseCountMatrix	1	0	1
plotPropositionSuccessRates	1	0	0
plotScoreDistribution	1	1	0
plotPredictorBinning	1	1	0
plotPredictorPerformance	1	1	0
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.

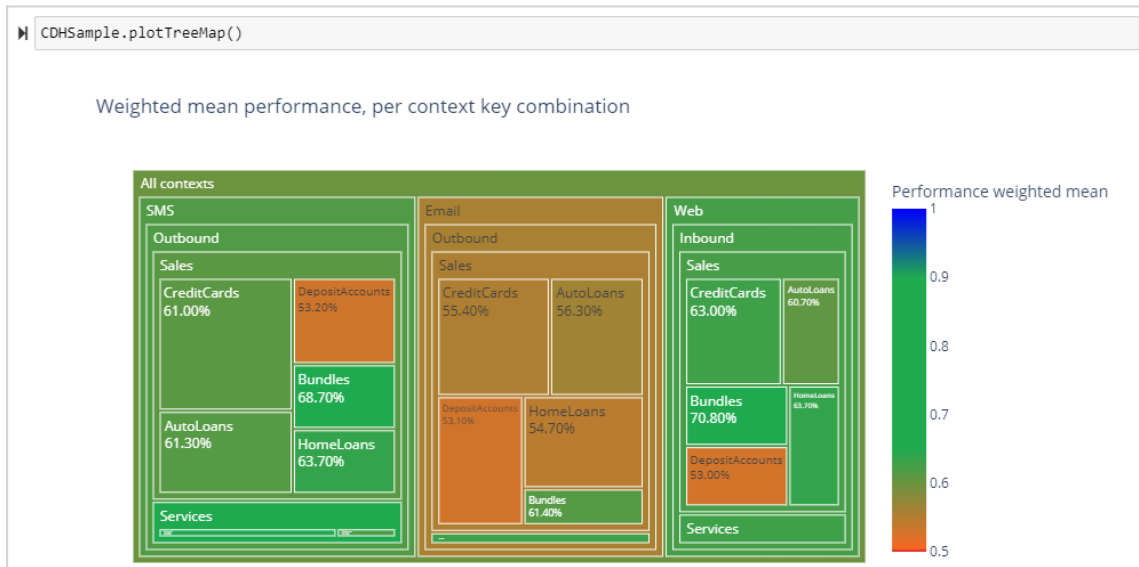


Zoom in on models that perform very well but have a low success rate, to report actions that need attention to the business.

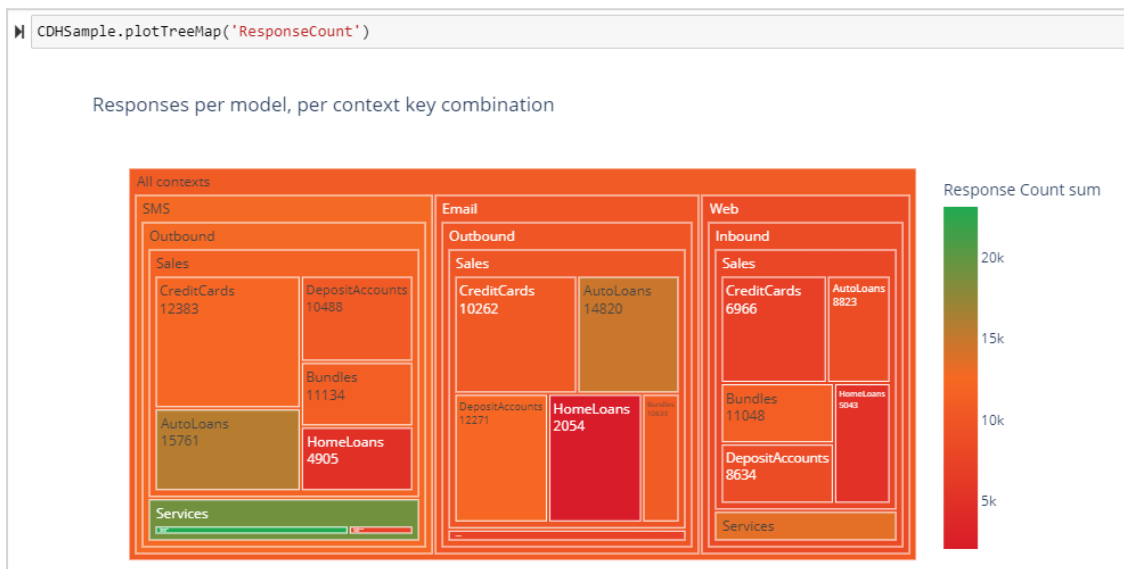
You can visualize a subset of the data by supplying a query argument. Let's only consider the models with a high response count within the CreditCards group.



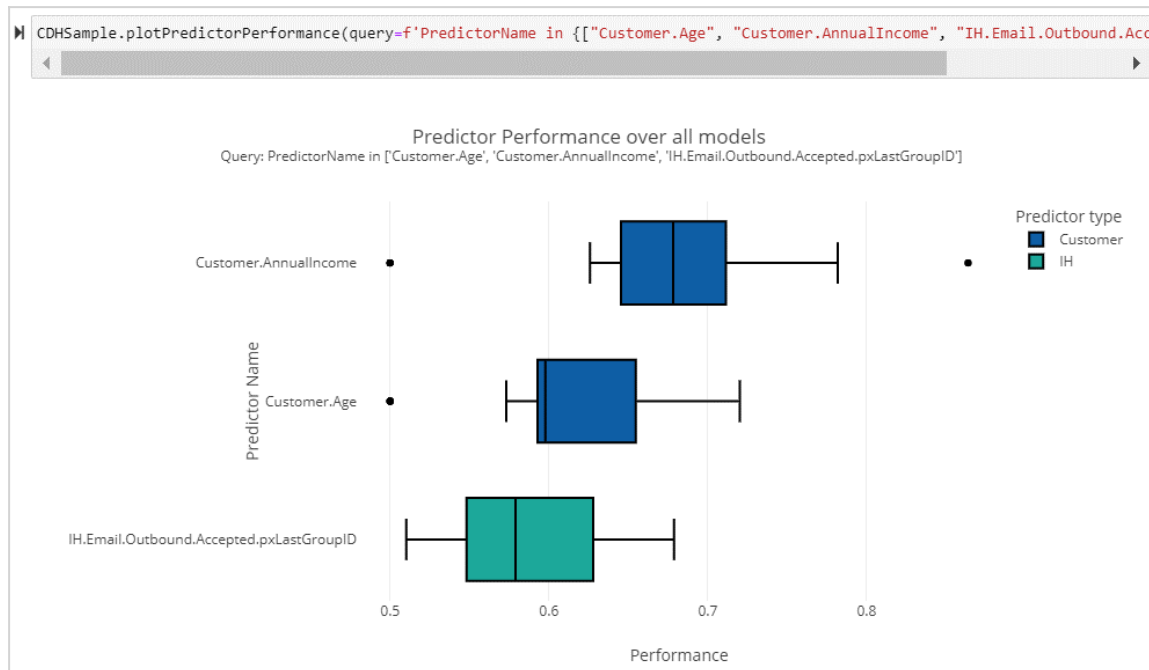
The bubble chart shows you which models perform well. You might, however, want to know if performance issues occur in a specific channel, issue, or group. The Treemap visualization offers insight in this situation.



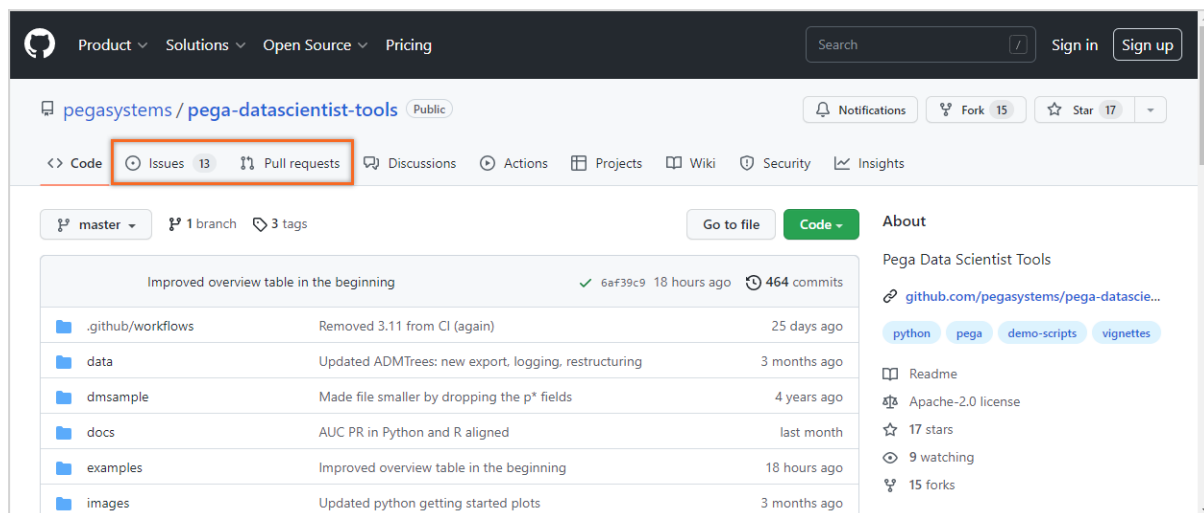
By default, the Treemap shows the performance, weighted by the response count. The number of model IDs within a combination of context keys determines the size of the squares. Besides performance, you can also use another variable, such as the SuccessRate or ResponseCount.



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.



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.



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 one of many business use cases that involve predictive models. Pega Customer Decision Hub™ provides predictions that use predictive models to improve one-to-one customer interactions. Learn how to create new predictions and the predictive models that drive them in Prediction Studio.

Learning objectives

- Create a new prediction that uses a predictive model.
- Build a scorecard to drive a churn prediction in Prediction Studio.
- Create a new predictive model with Pega Machine Learning.
- Import third-party predictive models in the PMML and H2O.ai formats.
- Use machine learning services such as Amazon SageMaker and Google machine learning.

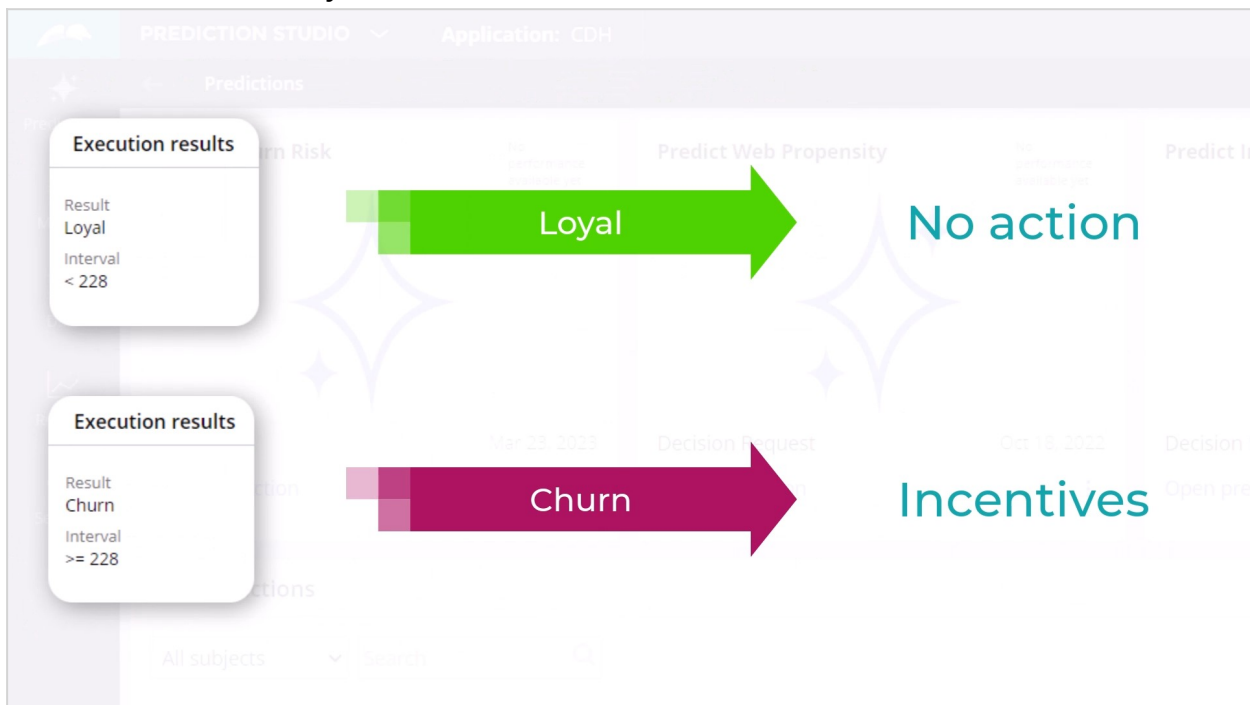
Creating a churn prediction using a scorecard

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

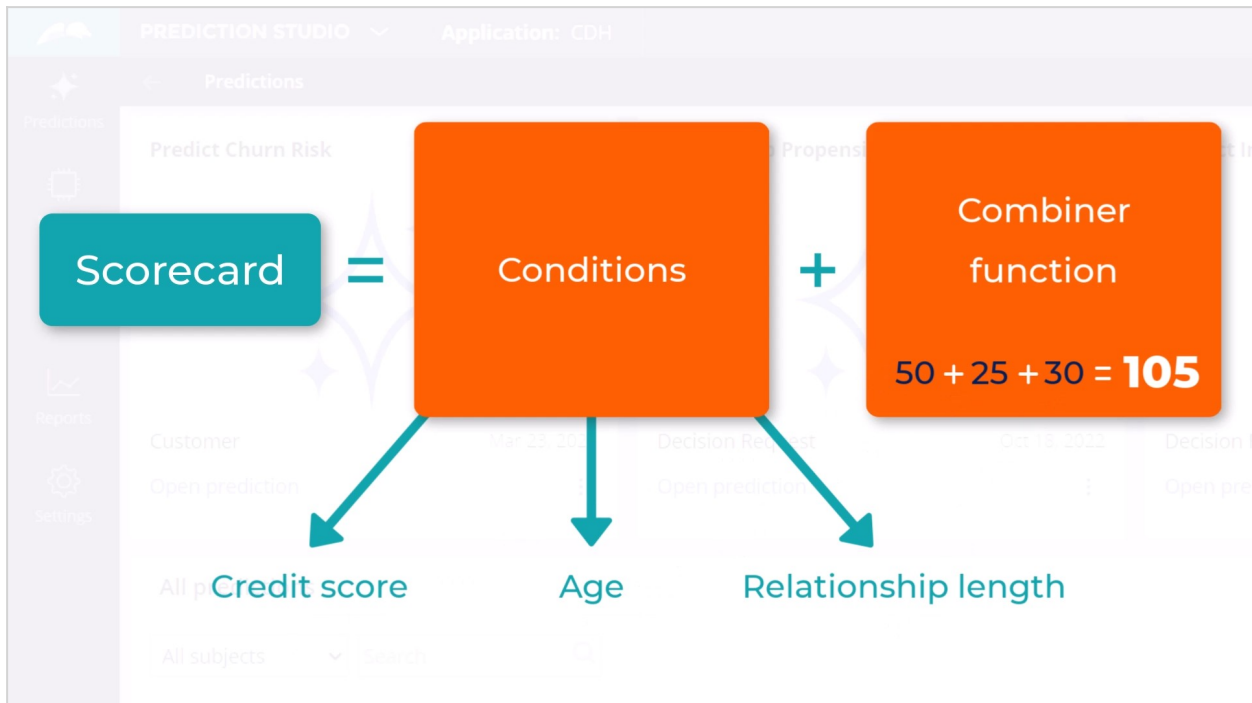
In this video, you will explore the use of scorecards to drive predictions in Pega Customer Decision Hub™.

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.

The screenshot shows the "Create a prediction" dialog box in Prediction Studio. The dialog box has a close button (X) in the top right corner. It contains the following fields and options:

- Prediction name**: Predict Churn Risk
- Outcome**: Churn
- Churn**: Predict how likely a customer will churn.
- Subject**: Customer

At the bottom of the dialog box, there are two buttons: "Back" and "Create".

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*.RelationshipLengthDays)	<= 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.

Refresh

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	Score
Loyal	50.0
Interval	Minimum score
< 122	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	Score
Churn	130.0
Interval	Minimum score
>= 122	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

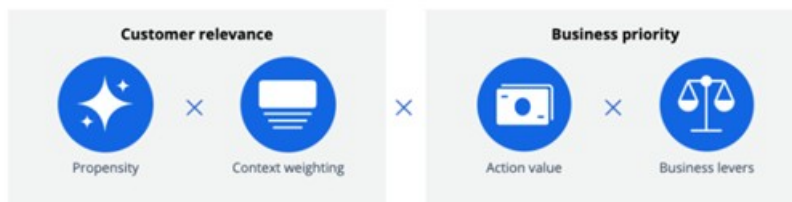
Enhance decision strategies with predictive models built on customer interaction data and let Pega Customer Decision Hub™ bring even more relevance to every customer engagement. Build models using Pega's machine learning capabilities, import models built with third-party tools, and incorporate the latest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

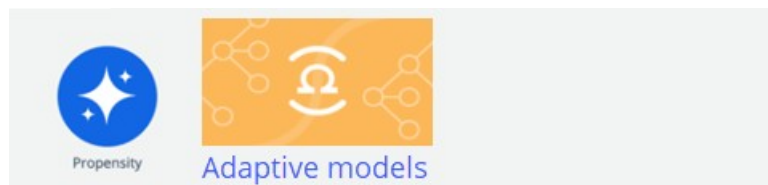
Next best actions balance customer relevance and business priorities by selecting the actions with the highest priority.

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.

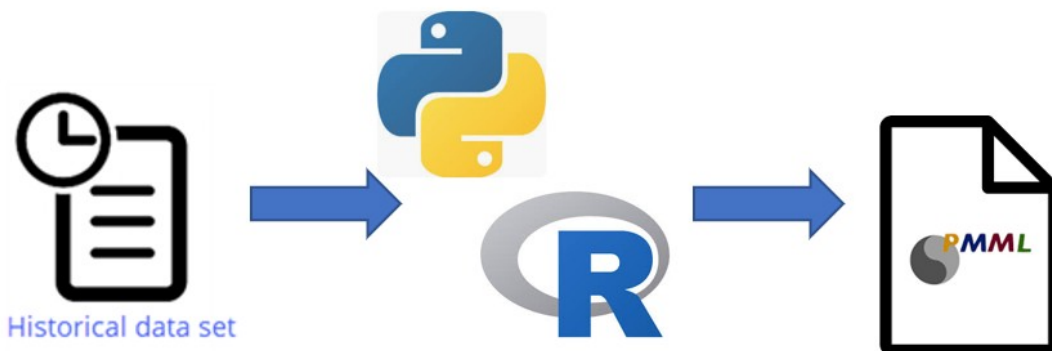
This model can then be used in decision strategies. When the decision strategies execute, the models are executed inside the Pega platform.



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 AI Platform.



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 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 ML and the Amazon SageMaker platform.

Building models with Pega machine learning

Introduction

Prediction Studio offers several options for leveraging customer data to create predictive models. Learn how to develop powerful and reliable models that can predict customer behavior, such as offer acceptance, churn rate, credit risk, or other types of behavior by using Pega machine learning.

Transcript

This demo will show you how to build a predictive model using Pega machine learning in Prediction Studio.

In an effort to proactively prevent churn, U+ Bank wants to predict the likelihood that a customer will leave the bank in the near future. When starting to build a new model, you will be presented with the option to create a model on a template that is used for streamlining model development. One of these is churn modeling.

New predictive model [X]

Name *
ChurnPegaML

Create model ?

Use Pega machine learning | Import model | Select external model

Category: Retention | Template: Churn Modeling

Churn Modeling
Aims at ordering cases in terms of their propensity to churn within a defined length of time. Score bands are created to enable cases with different levels of propensity to be selected or deselected. Behavior: Churn can be defined as closure of a relationship in a following period (e.g. within three months after the potentially predictive data was captured). Cases can be restricted to those who suffered some adverse experience or those who would be targeted by some competitive offer. Predictions: In addition to the probability of churn, a model may analyze and forecast the probability of each possible reason for dormancy and retention.

The model build itself consists of 5 steps: Data preparation, data analysis, model development, model analysis, and model selection. In the data preparation step, the data source containing the historical data is selected, the sample is constructed, and the outcome of the model is defined. The data source can be a csv-file, a database table, a data flow, or a data set.

Source selection

Select the data source for the creation of predictive models and preview the first 100 records.

CSV
 Database
 Data flow
 Data set

Upload flat file
 No file chosen

Separator character:
 Quote character:
 First line contains field names:

The preview of the first ten records in the data set allows you to verify that all fields will be correctly imported.

Preview for first 10 records of historical_data.csv

Field	Record 1	Record 2	Record 3
CustomerID	14	15	16
ACCOUNT_ID	---	---	---
Title	---	---	---
pyFullName	Troy Murphy	Barbara Stockton	Joanna Williams
Gender	M	F	F
Age	26	32	25

Next, construct the sample. Using a weight field is optional; it is only used when the data source contains such a field. If you do not specify the field, each case counts as one. The type of field to be sampled can be set to either numeric or categorical.

Select the fields to sample

Field	Type
CustomerID	Categorical
ACCOUNT_ID	Categorical
Title	Not used

By default, all fields are considered potential predictors. When setting predictors, it's important to use some common sense. For example, the customer ID is a random number and has no impact on the behavior to be predicted.

Likewise, the name of the customer has no predictive value. For such fields, change the type to 'Not used'. If the data contains a relatively small number of cases, you will want to use 100% of the records. If the data source is large, a sample will be sufficient.

Select sampling method

Uniform sampling
 Stratified sampling

Set sample size using

% or Cases

Next, you define the hold-out sets for validation and testing during model development. Your models will be trained with the remainder of the data. Once trained, the validation set is used to check for robustness of candidate models and to compare their performance.

Finally, the test set is used to analyze the performance characteristics of candidate models, and to select the best model.

Hold-out sets

Split the sample into a development, validation and test set. ?

Create hold-out sets by

Setting percentages for each set

User defined field

Retain % of the sample for validation (201 cases)

and % of the sample for testing (201 cases)

60.0 % of the sample for development (604 cases)

Finalize the data preparation step by defining the outcome to be predicted. You can predict a binary outcome, as in this example, or predict a continuous outcome. For a binary outcome type, the outcome field must be categorical. For a continuous outcome type, the outcome field must be numerical. Here you also map the values of the outcome field to the outcome category. With that, you specify how to differentiate between good and bad behavior.

Outcome definition

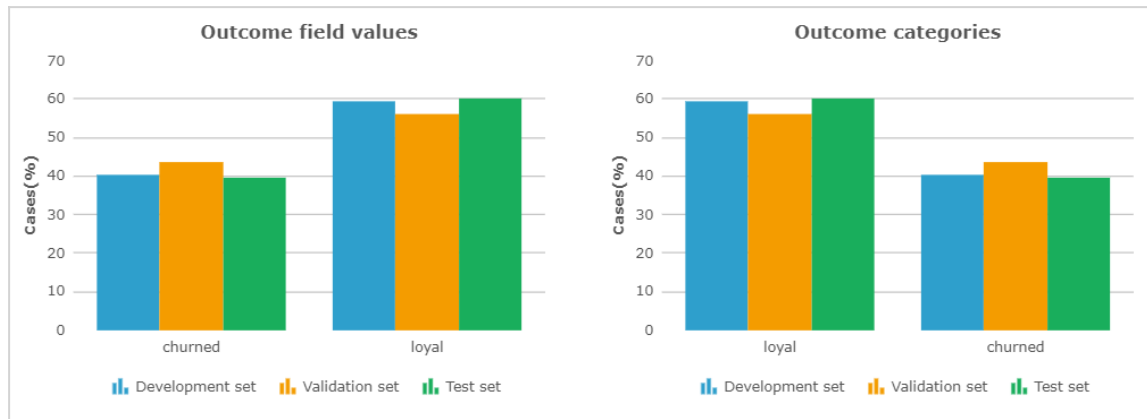
Define the outcome to be predicted. Predict a binary outcome, only categorical fields can be selected for this. Or predict a continuous outcome, only numeric fields can be selected for this.

Outcome type Outcome field to predict

Map possible values of outcome field to outcome category

	Value		Outcome category
Map	loyal	to	<input type="text" value="loyal"/>
Map	churned	to	<input type="text" value="churned"/>

It is worthwhile to verify that the customer distribution across the development data set is similar to the whole sample.



In the data analysis step, you analyze the individual predictors. By default, only predictors with a performance higher than 52 are included. For fields that have a very high performance, the Role is set to *value* to protect models from accidentally using predictors that might be directly correlated to the outcome.

Exclude predictors with a performance below

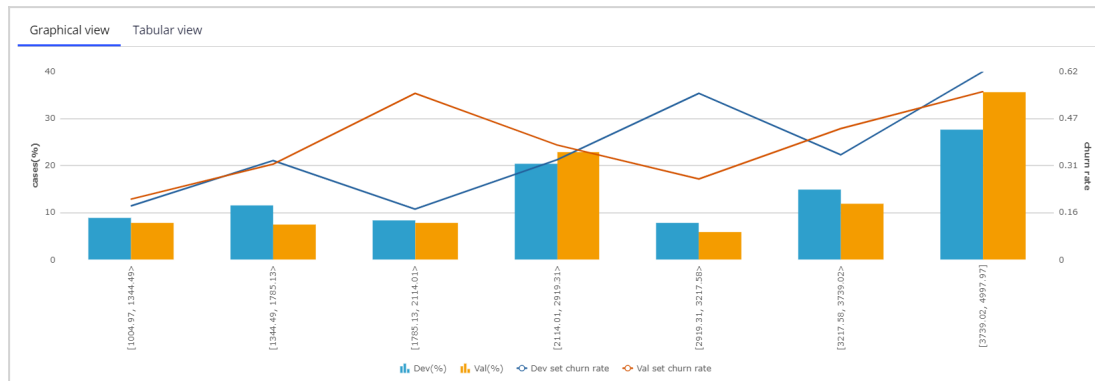
Change role ▾ Reports ▾ New virtual field Reset Show groups

<input type="checkbox"/> Predictor	Type	Role	Binned intervals	Grouped intervals	Grouped performance
<input type="checkbox"/> RiskScore	Numeric	VALUE	200	11	90.27
<input type="checkbox"/> AverageSpent	Numeric	PREDICTOR	200	7	68.28
<input type="checkbox"/> MonthlyPremium	Numeric	PREDICTOR	200	10	64.84
<input type="checkbox"/> Age	Numeric	PREDICTOR	70	8	63.87

You can also manipulate features to create a better predictor by creating a 'New virtual field'. This is a fundamental step towards having good models. $Income \cdot CLV$ is such a virtual field. The performance of this new predictor is higher than that of the individual fields.

<input type="checkbox"/> Predictor	Type	Role	Binned intervals	Grouped intervals	Grouped performance
<input type="checkbox"/> RiskScore	Numeric	PREDICTOR	200	11	90.27
<input type="checkbox"/> AverageSpent	Numeric	PREDICTOR	200	7	68.28
<input type="checkbox"/> $Income \cdot CLV$	Numeric	PREDICTOR	200	10	65.08
<input type="checkbox"/> MonthlyPremium	Numeric	PREDICTOR	200	10	64.84
<input type="checkbox"/> Age	Numeric	PREDICTOR	70	8	63.87
<input type="checkbox"/> Income	Numeric	PREDICTOR	200	10	63.74
<input type="checkbox"/> CLV_VALUE	Numeric	PREDICTOR	200	9	63.47

Data analysis creates a binned, ordinal view of individual predictors. Both Binning and Granularity are automatically set but can be manually adjusted.



As part of model development, the grouping and predictor selection process is automated. When multiple predictors are correlated, considering them all for the machine learning process will lead to unnecessary model complexity. It is best practice to select the best-performing predictor in each group.

Predictor grouping

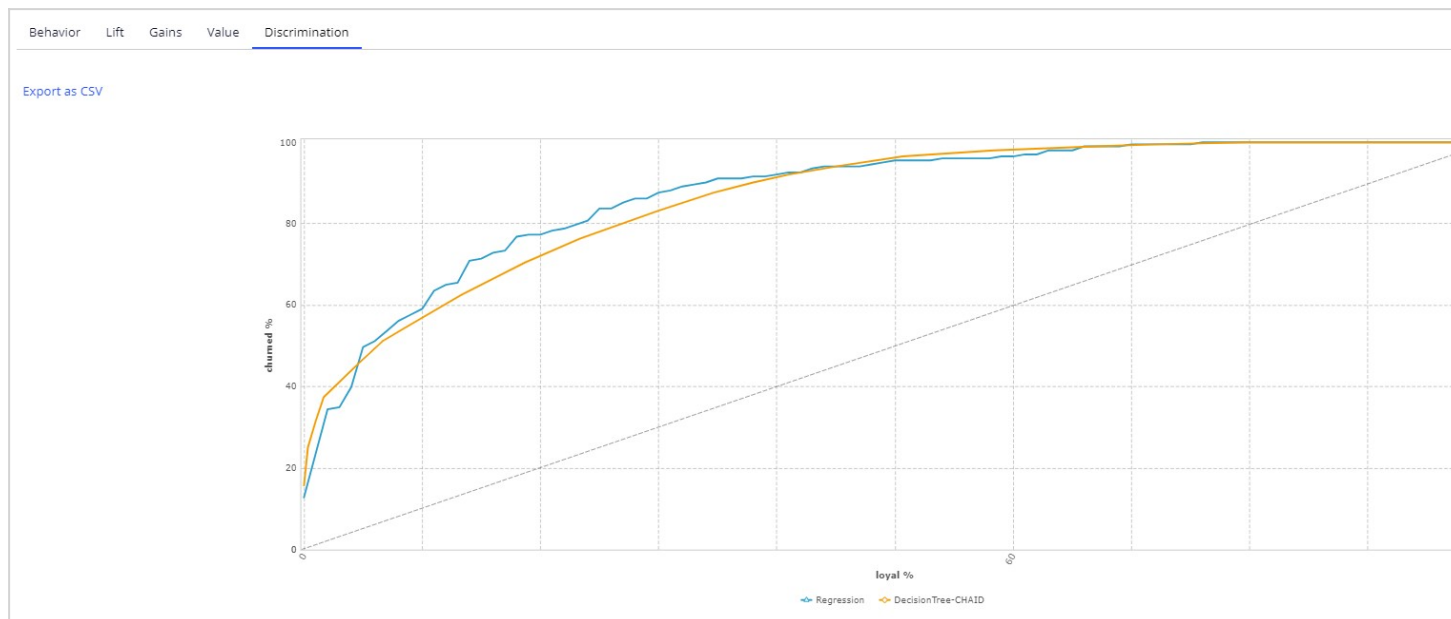
In predictor grouping, correlated predictors are grouped together. You can select to use all predictors (default) or alternatively, continue with the best predictor from each group.

Grouping level:

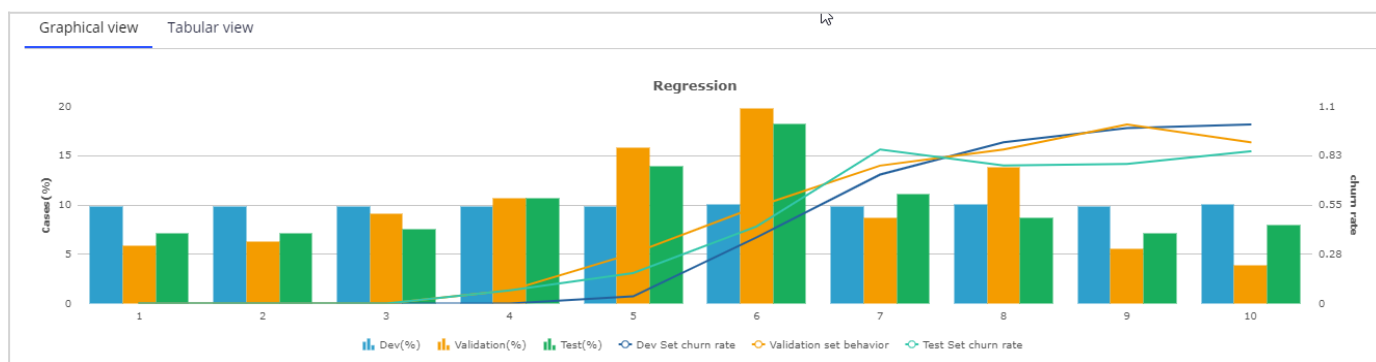
Prediction Studio provides a rich model factory that supports industry-standard models. You can create 4 types of models: Regression models, Decision tree models, Bivariate models, and Genetic algorithm models. By default, a Regression and a Decision tree model are automatically created. These models are highly transparent. Bivariate models and Genetic algorithm models have a lower transparency score.

Model Type	Transparency Score
Bivariate model Pega	3
Genetic algorithm Pega	2
Regression	4
Tree model	5

The purpose of the next step, Model Analysis, is to select the best model for your use case. In the 'Score comparison' step, you can compare the scores generated by the models in terms of behavior, lift, gains, and discrimination. Prediction Studio uses Area Under the Curve (or AUC) to measure the performance of predictors and models. You can describe AUC as the measure of how well the model is able to discriminate between good and bad cases. The value of AUC ranges from 50%: random distribution, to 100%: perfect discrimination.



In the 'Score distribution' step, the model scores are segmented based on a method you select. A typical example divides the scores into deciles: 10 classes with an equal number of cases.



The 'Score distribution' settings give several methods for defining these segments.

Score distribution settings
 Segmentation method
 Create bands with equal number of cases
 Create statistically significant bands
 Create monotonically increasing bands
 Create user defined bands

Max. # of bands ★
 Number ★
 or Percentage ★

Only count cases where the outcome equals

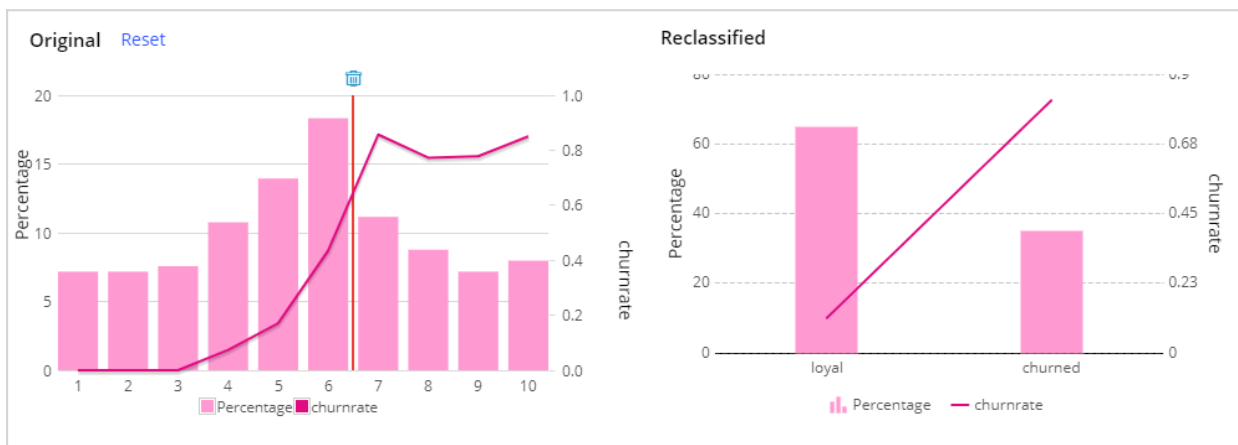
In the 'Class comparison' step, you can analyze and compare models after the score distribution has been adjusted.

Finally, you select the model that best fits your needs and specify the context in which to save it. The default context where the models are saved is the customer class. Before you

can save the model, check the mapping of the predictors to the properties in the customer class. If the properties exist and have a name similar to a predictor field name, they will be mapped automatically. You also have the option to create missing properties, but this should be discussed with the system architect beforehand.

Monitor Model <u>Mapping</u>		
The model predictors are automatically mapped to fields in the data model.		
<input type="button" value="Create missing fields"/> <input type="button" value="Refresh mapping"/>		
Predictor	Predictor Type	Field
Gender	Symbolic	.Gender
Age	Numeric	.Age
MaritalStatus	Symbolic	.MaritalStatus

If needed, you can adjust the score distribution segments by clicking on the original score distribution chart. In this example, two segments are appropriate: loyal and churned.



The model can now be saved and is ready for use in a decisioning strategy.

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

- How to create a predictive model in Prediction Studio using Pega machine learning.

Importing predictive models

Introduction

During a Pega Decision Management implementation project, you may discover that the company already uses predictive models. These assets can be reused in Pega Decision Management to help make customer predictions.

Transcript

This demo will show you how to import third-party predictive models into Prediction Studio and use them natively in Next-Best-Action strategies.

Prediction Studio supports two external model formats. First, you can import models in the Predictive Model Markup Language (PMML) format. PMML is an XML-based language aimed at easily sharing predictive models between applications. It is the de facto standard for representing not only predictive models, but also data, pre- and post-processing.

Additionally, you can import models built with H2O.ai, an open source machine learning and predictive analytics platform that allows you to build machine learning models on big data. The processes for importing PMML and H2O models are identical and start with creating a new predictive model strategy component.

Create model



Predictive model

Predict customer behavior such as offer acceptance, churn rate or credit risk based on customer data.



Adaptive model

Predict customer behavior using self-learning models.



Text categorization

Analyze and assign text to a specified category.



Text extraction

Analyze unstructured text to extract required words or phrases.

Prediction Studio offers three options for creating a predictive model: using Pega machine learning, importing a previously built model, or using an external model.

To leverage an existing model file, select the **Import model** option. Upload the PMML or H2O model file. The default context of the model is the **Customer class**, where the customer data model properties are stored. You can change this class if required.

New predictive model

Name *

ChurnPMML

Create model ?

Use Pega machine learning

Import model

Select external model

Import model file * ?

Choose File

File name

ChurnPMML.pmml

Context

Customer [Change](#)

In the **Outcome definition** dialog box, you define which probability you would like to predict and the expected performance of the model, which is used as a benchmark when monitoring the model.

New predictive model

Outcome definition [Set labels](#)

The objective of the model is to predict
Segment

Predict the probability of

churned

loyal

Modeling technique

Tree model

Expected performance (AUC) ?

80

Import the model and, on the **Mapping** tab, make sure that all predictors are mapped to fields in the data model. Missing fields can be created, but this should be discussed with the system architect beforehand.

The model predictors are automatically mapped to fields in the data model.

Create missing fields

Refresh mapping

After the model is saved, you can test it for a single customer or run it for a batch of customers.

Run predictive model

Troy Data Transform

Field name	Type	Input
Age	Double	26
Gender	string	M
NetPromoterScore	Double	9
MaritalStatus	string	Married
AverageBalance	Double	1500.67
AverageSpent	Double	3200.53
EmailOptIn	string	Y
DebtToIncomeRatio	Double	45
SMSOptIn	string	Y
MonthlyPremium	Double	0.0
HasMortgage	string	Y
DMOptIn	string	Y

Run

When you test the model for a single customer, you can use a data transform as input data. When customer Troy is used as the data source, the model predicts that he is likely to churn. The model also outputs his propensity to churn, which is, in this case, 93.42%.

Run predictive model

Single run Batch run

> Inputs

< Outputs

Results

Result	Monitoring performance
churned	0
Propensity	Monitoring evidence
0.9342621091861922	0.0

Output	Value
Segment	churned

In contrast, the model predicts that customer Barbara is likely to remain loyal, with a low propensity to churn of 35.83%.

Run predictive model

Single run **Batch run**

> Inputs

∨ Outputs

Results

Result
loyal
Monitoring performance
0

Propensity
0.3583554398897344
Monitoring evidence
0.0

Output	Value
Segment	loyal

You can also run the model on a batch of customers. When the model is run for a larger input data set, the output shows the number of customers that are classified as either likely to remain loyal or likely to churn in the near term.

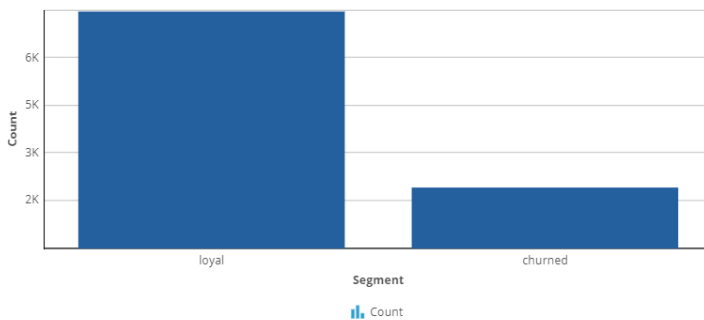
Run predictive model

Single run **Batch run**

Data source*
CustomerBatch
Source type
Data set
This data source contains approximately 10,000 records.

Total records executed: 10000 Total failed: 0

Output
Segment



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

- How to import third-party predictive models into Prediction Studio.
- How to test the model for a single customer.
- How to run the model for a batch of customers.

Using machine learning services

Introduction

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

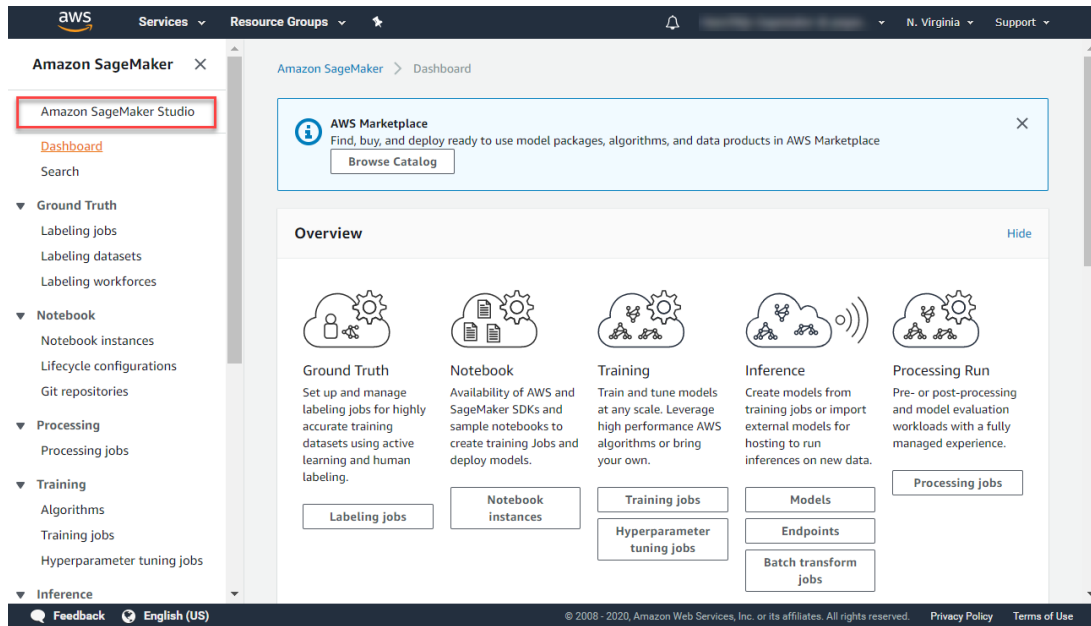
Transcript

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. The steps are similar to using other machine learning services such as Google AI Platform. 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.



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

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

S3 object key prefix

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.

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

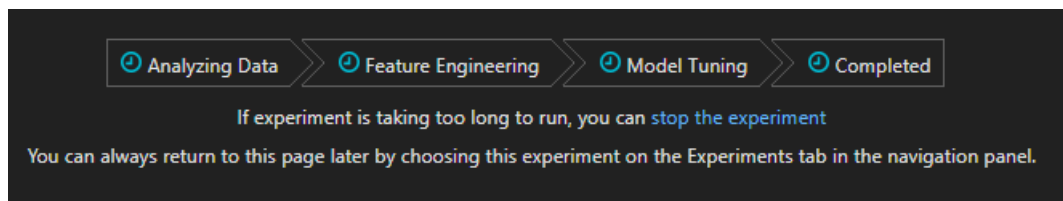
Output data location (S3 bucket)
 Enter the location in S3 where you want to store the output.

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

S3 object key prefix

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



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

EXPERIMENT: CHURNAWS Open candidate generation notebook Open data exploration notebook

Trials Job profile

TRIALS

1 row selected Deploy model

Trial name	Status	Start time	Objective
★ Best tuning-job-1-1a89f03cd5343889f-205-33590e8	Completed	2 hours ago	0.9393600225448608
tuning-job-1-1a89f03cd5343889f-184-10566b3	Completed	2 hours ago	0.934220016002655
tuning-job-1-1a89f03cd5343889f-171-ce9ec6a4	Completed	3 hours ago	0.934220016002655
tuning-job-1-1a89f03cd5343889f-238-e8521619	Completed	2 hours ago	0.934220016002655
tuning-job-1-1a89f03cd5343889f-211-e0321194	Completed	2 hours ago	0.934220016002655
tuning-job-1-1a89f03cd5343889f-144-27960f16	Completed	3 hours ago	0.9340400099754333
tuning-job-1-1a89f03cd5343889f-148-Satfdaf7	Completed	3 hours ago	0.9340400099754333
tuning-job-1-1a89f03cd5343889f-248-caa2bca6	Completed	2 hours ago	0.9340400099754333
tuning-job-1-1a89f03cd5343889f-166-c13901fa	Completed	3 hours ago	0.9340400099754333
tuning-job-1-1a89f03cd5343889f-168-8446584	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

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

Instance type Instance count

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.

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)

Cancel

Save

Test machine learning service



Successfully connected to the machine learning service.

Close

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

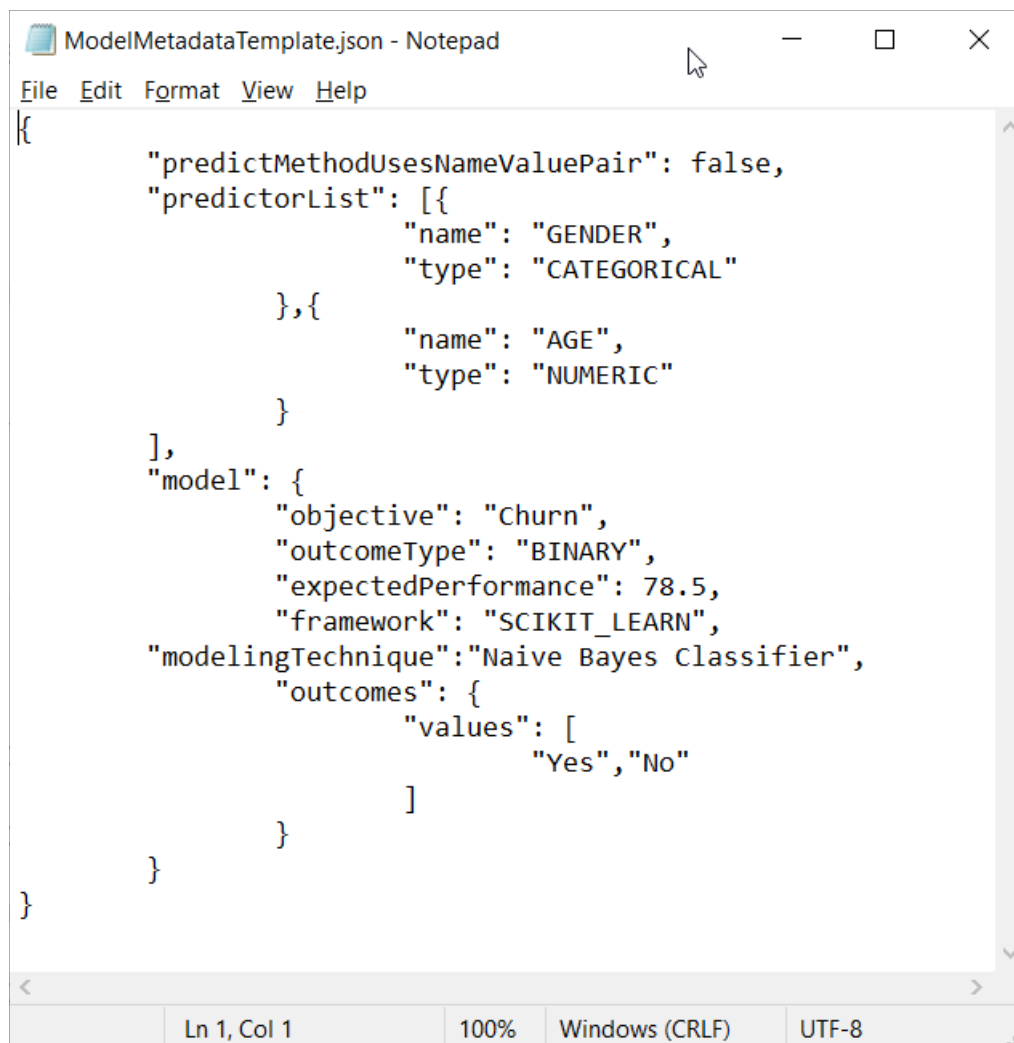
Name ★
ChurnSageMaker

Create model ?

Use Pega machine learning Import model **Select external model**

Machine learning service ★ Model
Amazon Machine Learning ▼ SageMakerChurn-model ▼

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



```
{
  "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. In Google AI Platform, this property is automatically fetched.

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) [?](#)

Back Cancel Create

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	Monitoring performance
0.9071381688117981	0
Propensity	Monitoring evidence
—	0.0

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

∨ Outputs

Results

Result	Monitoring performance
0.0010575958294793963	0
Propensity	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.

Creating a prediction using an ML model

Introduction

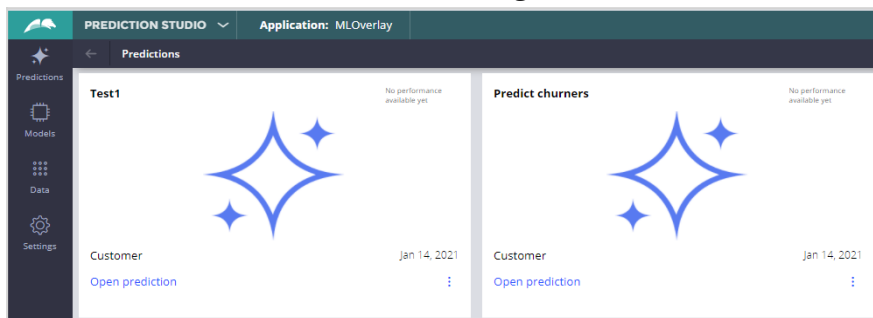
Acquiring new customers can be more costly than retaining current customers. U+ Bank implements Pega Customer Decision Hub™ for their customer engagement and wants to reduce the churn rate. Learn how to create a new prediction in Prediction Studio that calculates the likelihood that a customer might churn in the near future.

Transcript

This demo shows you how to create a new prediction in Prediction Studio in the development environment. Predictions combine predictive models and best practices in data science.

U+ Bank uses Pega Customer Decision Hub™ to personalize the credit card offered to customers on their website. If a customer is eligible for multiple offers, artificial intelligence (AI) decides which offer to show. For customers that are likely to leave the bank soon, the bank wants to make a proactive retention offer instead of a credit card offer.

The bank has recorded historical churn data for its customer base, and a data scientist used this data to create a predictive churn model. With this model, you create a prediction to use in Customer Decision Hub to display a retention offer to customers with a high churn risk on the website. Predictions are managed in Prediction Studio.



You can create three types of predictions. To improve customer engagement with retention offers, choose Customer Decision Hub. Predictions for case automation and text analytics are also available.

To create a prediction that aims to calculate the likelihood that a customer might churn, 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 this prediction. This placeholder is useful in case you do not have a predictive model yet, as it allows the Next-Best-Action

specialist to continue work while a predictive model is built.

Churn			
Name	Type	Performance	Status
Test2	Scorecard	---	ACTIVE

As you already have a predictive model, the next step is to replace the scorecard that drives this prediction with the predictive churn model. When the replacement is ready for review, approve the candidate model, and save the configuration.

Churn			
Name	Type	Performance	Status
Churn	Predictive model	---	ACTIVE

Once the prediction is created, test your work. Select a persona as the data source and run the prediction. Troy is predicted to leave the bank in the near future; therefore, the outcome is churn. Barbara has a low propensity to churn; therefore, the outcome is loyal.

The prediction is created in a branch in the development environment. A system architect needs to merge the branch to the application to ensure that the prediction is part of the CDH-Artifacts ruleset. Only then can the changes be deployed to the other environments using the enterprise 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 in a prediction

MLOps

Description

In the ever-evolving world of data-driven decisions, organizations aim to refine predictive models for optimal outcomes. This involves challenging existing models with candidates, tracking performance, and promoting effective models within the Customer Decision Hub framework.

Change management through 1:1 Operations Manager facilitates seamless adjustments, aided by validation datasets for effective model challenges. The process includes selecting sources, comparing models, and configuring challengers for review. Flexibility exists in shadowing, challenging, or replacing active models with contenders, all while keeping business outcomes in focus. Upon deployment, changes are enacted, and validated challenger models can be promoted, empowering adaptive decision-making in a dynamic landscape.

Learning objectives

- Describe the change management process.
- Shadow, challenge, or replace the predictive model that drives a prediction.

Change management process

Introduction

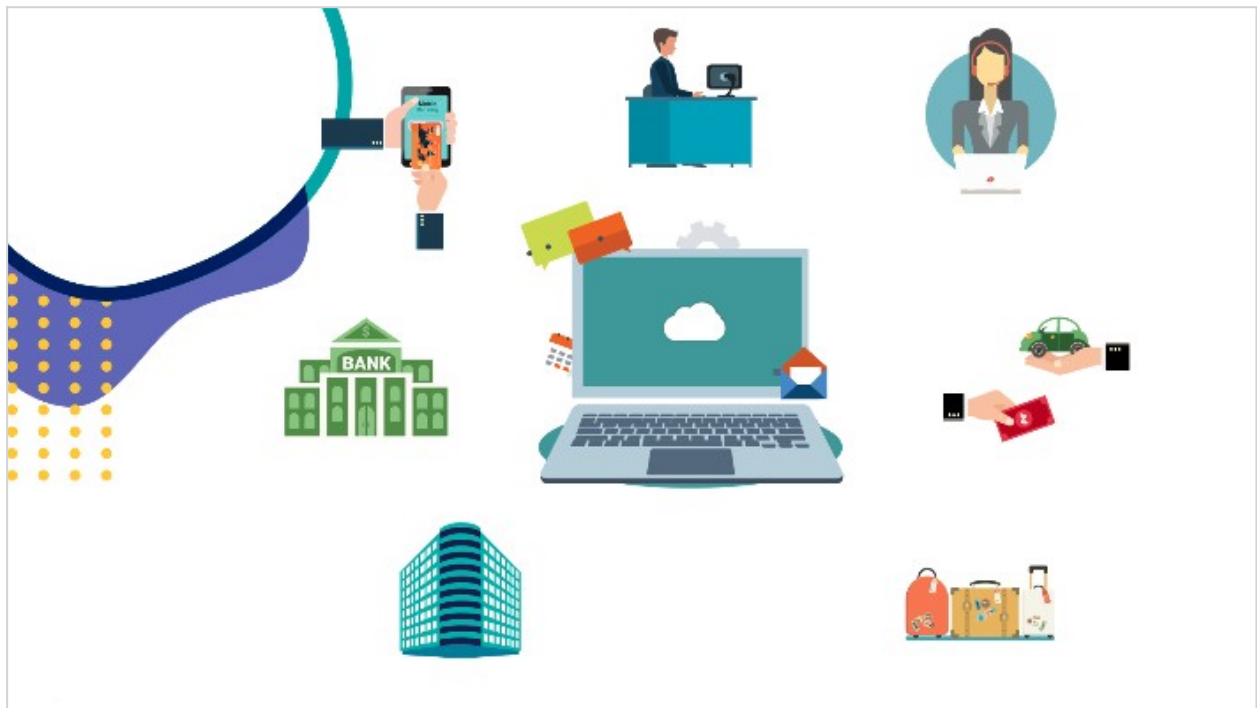
When business requirements change rapidly, the software development process needs to be more agile, while still producing high-quality, reliable software. This topic covers the importance of the change management process, and how it works in a one-to-one customer engagement project for Pega Cloud® customers.

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.

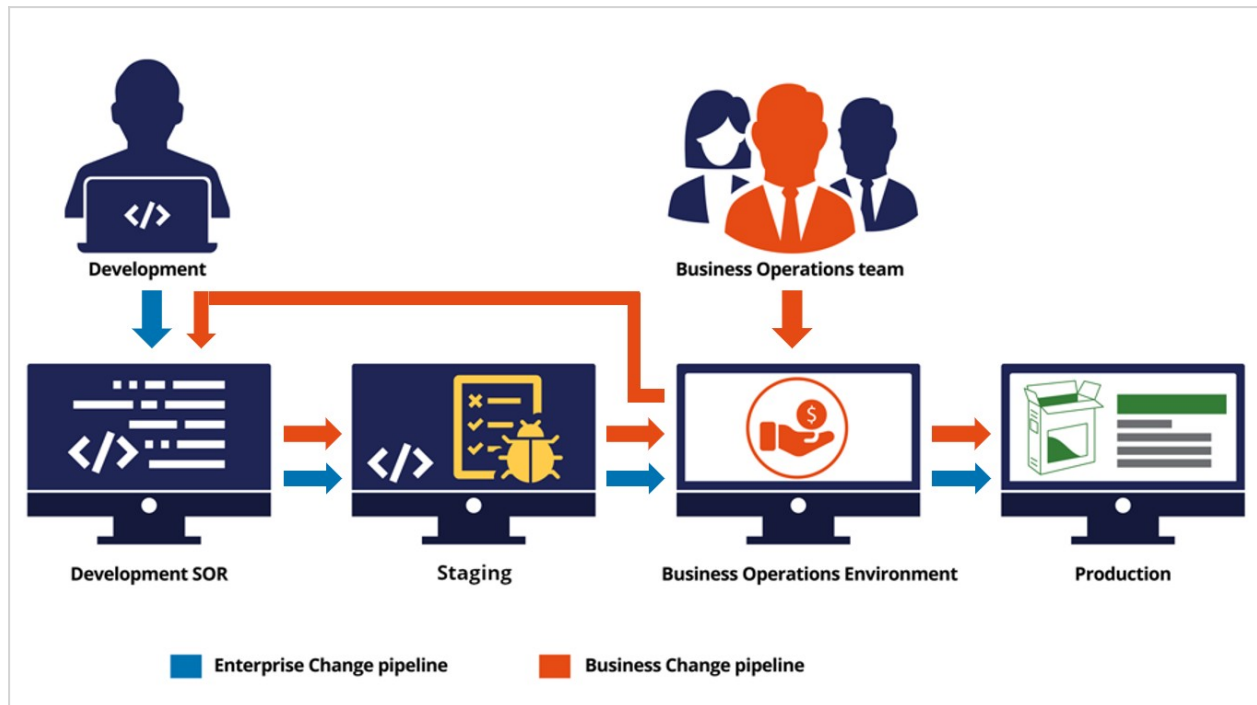
Developers make these changes in the development environment. Changes to the application are pushed to other environments through the enterprise-change pipeline managed by the Pega Deployment Manager™.

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.



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.

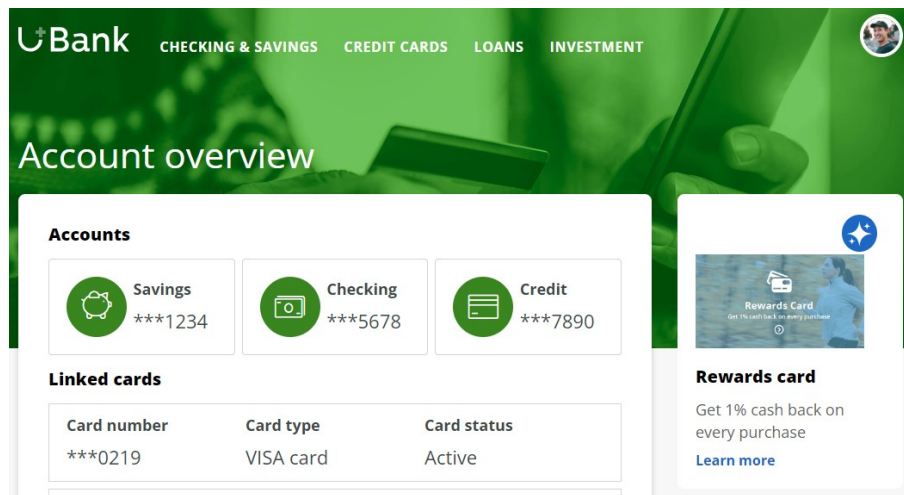
Challenging a predictive model

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.



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.

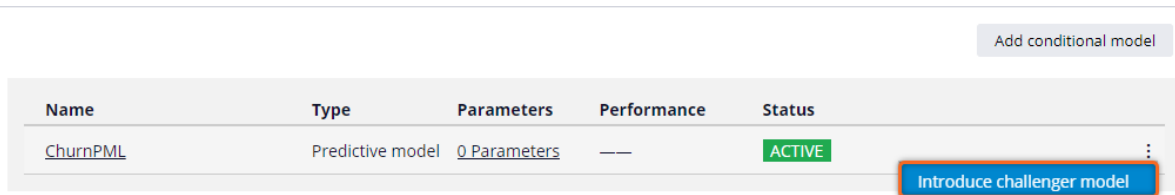
The bank personalizes interactions with customers that have a high churn risk, like Troy, and prioritizes retention offers. In this case, Troy receives a 'free extra miles' offer in the hope of retaining him. In contrast, customers with a low churn risk, like Barbara, receive a credit card offer. The 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 challenger 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)

No file chosen

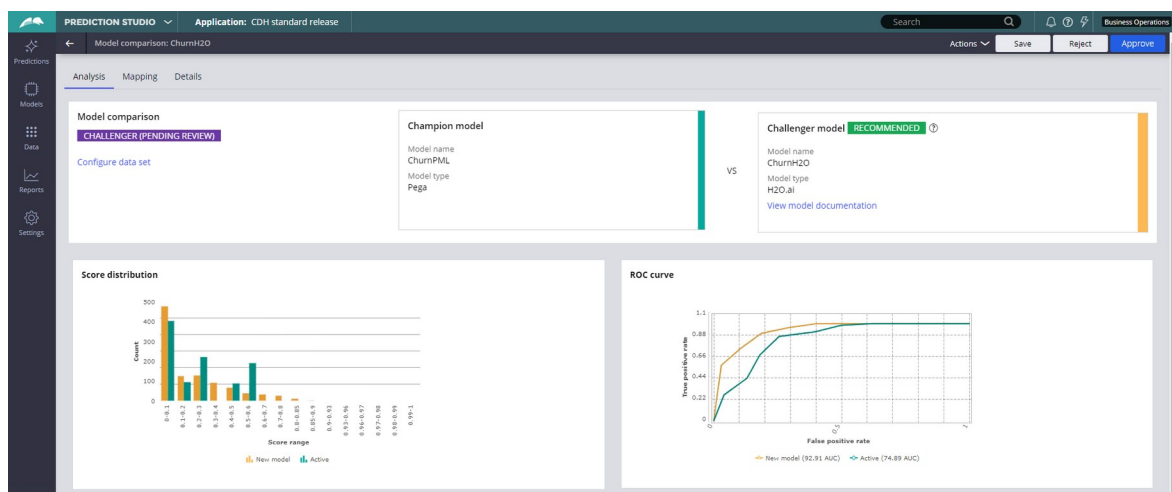
ModelDocumentation.txt 

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.



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

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.

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.

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 Financial Services Clickstream, a summary used for aggregating clickstreams on financial services-based web pages.

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

So, 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 the 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!

Creating and understanding decision strategies

Description

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

Learning objectives

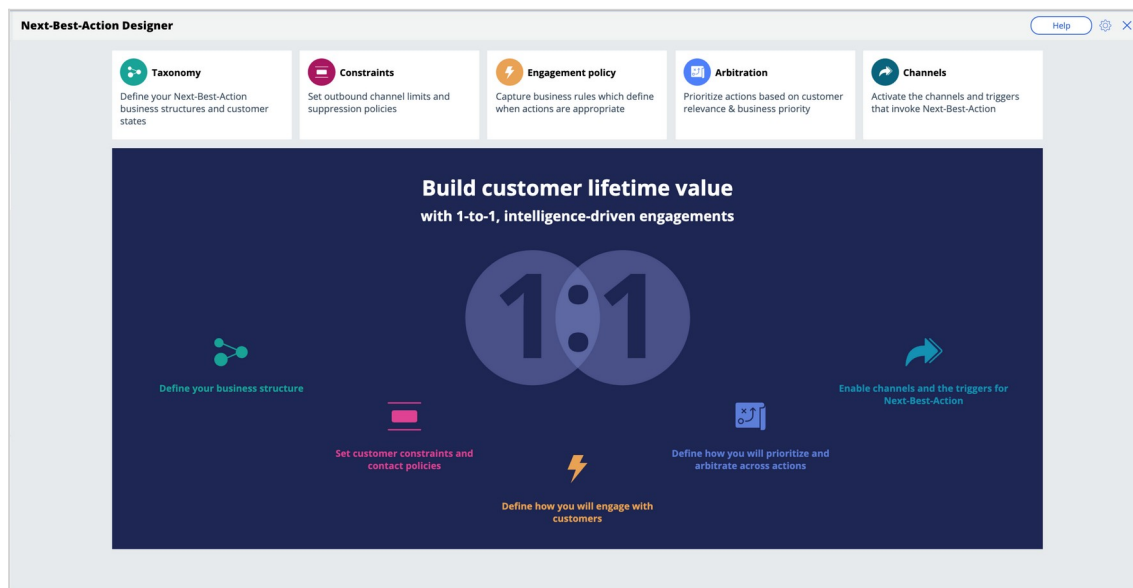
- Describe how decision strategies are used in the Next-Best-Action strategy framework
- Explain the decision strategy canvas and its building blocks
- Create decision strategies from scratch
- Explain what's going on inside each component when a decision strategy is executed

Decision strategies

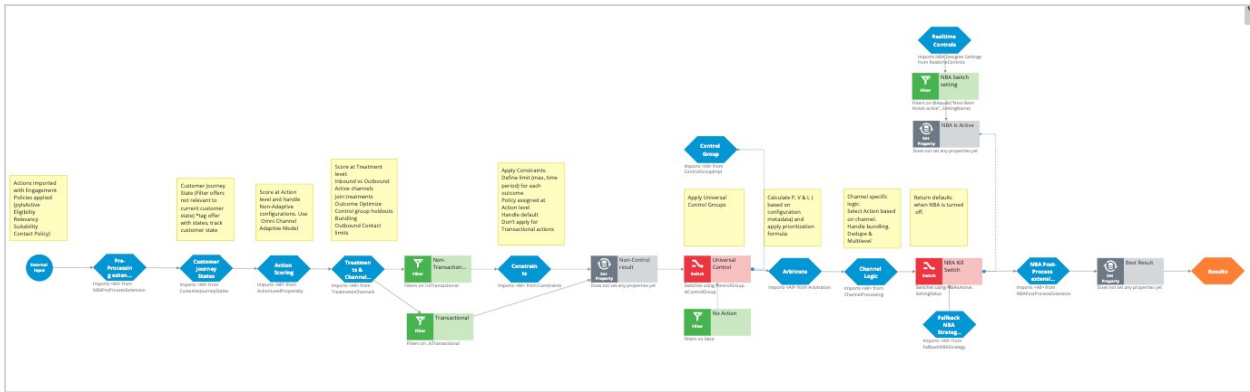
Transcript

Next-Best-Action Designer guides you through the creation of a Next-Best-Action strategy for your business. Its intuitive interface, proven best practices and sophisticated underlying decisioning technology enable you to automatically deliver personalized customer experiences across inbound, outbound, and paid channels.

The Next-Best-Action Designer user interface allows you to easily define, manage and monitor Next-Best-Actions.



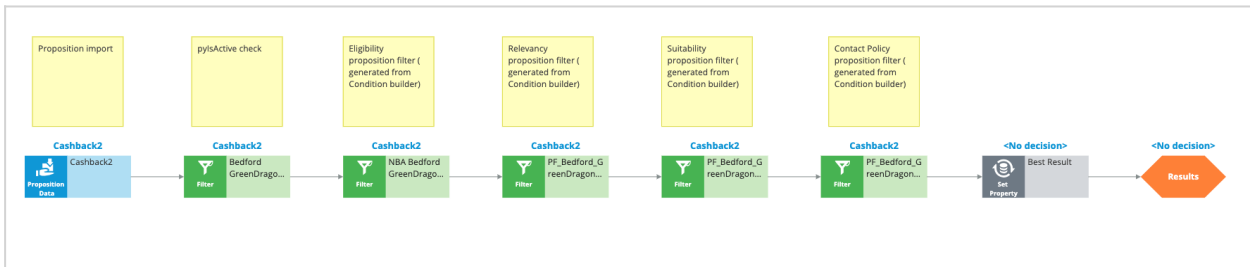
As you use the Next-Best-Action Designer user interface to define strategy criteria, the system uses these criteria to create the Next-Best-Action Strategy framework. This framework leverages best practices to generate Next-Best-Action decision strategies at the enterprise level. These decision strategies are a combination of the business rules and AI models that form the core of the Pega Centralized Decision Hub, which determines the personalized set of Next-Best-Actions for each customer.



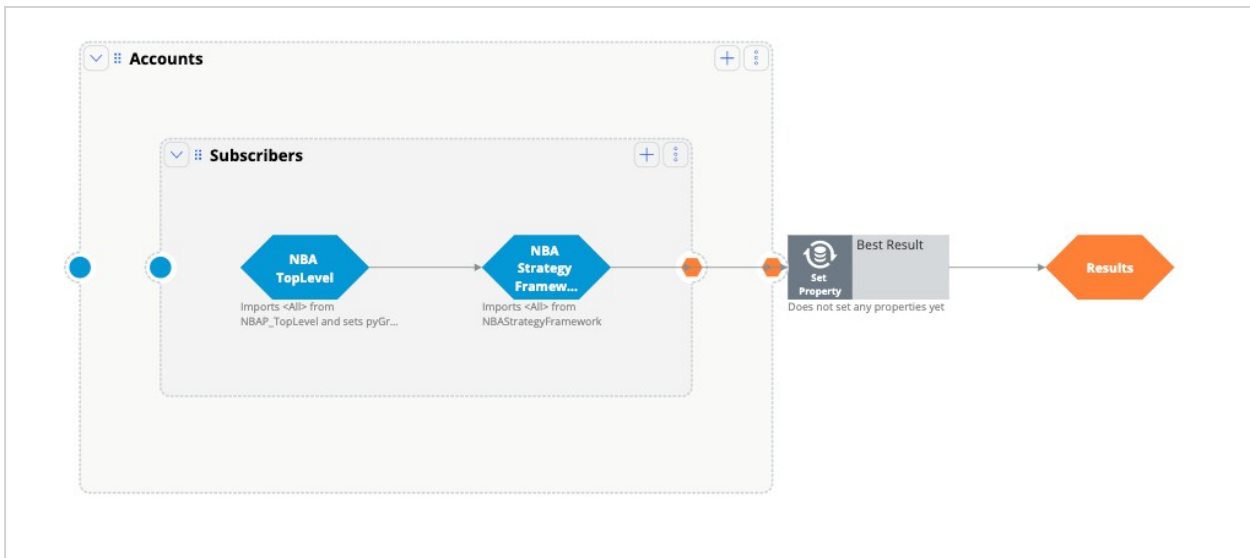
If you want to modify the strategy later, you can do that from Next-Best-Action Designer's simple and transparent interface.

The strategy framework is applied to all relevant Actions and Treatments after you define a Trigger in the Next-Best-Action Designer **Channels** tab.

Each Trigger generates a strategy that first imports the Actions from the appropriate level of the business structure and then applies the Eligibility, Applicability and Suitability rules.

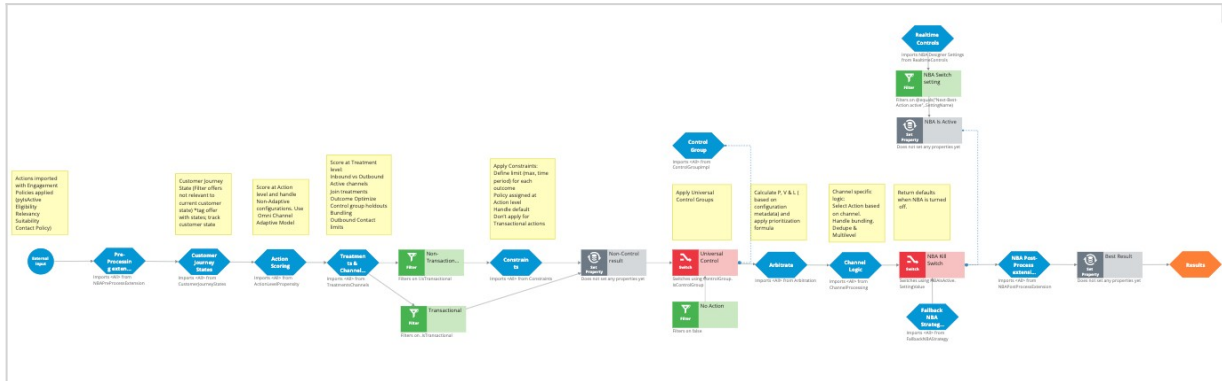


The strategy then passes these results to the strategy framework for processing.

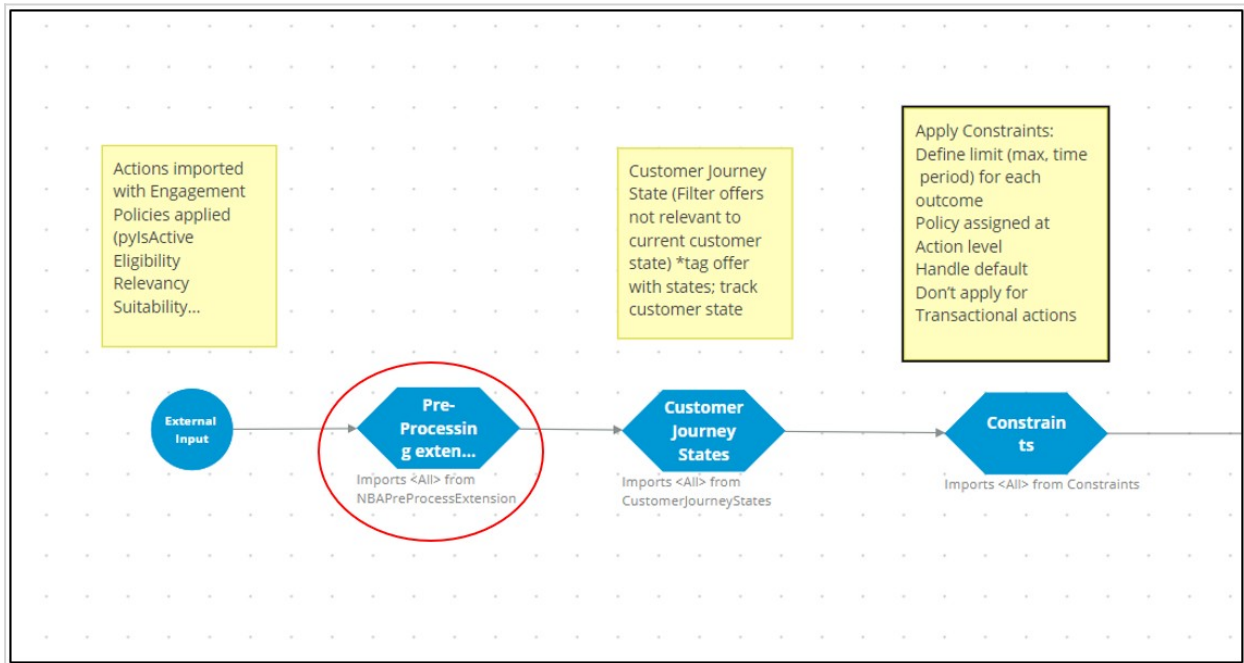


There are several extension points within the framework. An extension point is an empty rule or activity that is intended to be overridden to meet the specific needs of the application. When building an implementation of the current framework, the decision strategy designers must override the empty activity with a functioning interface to their customer master file.

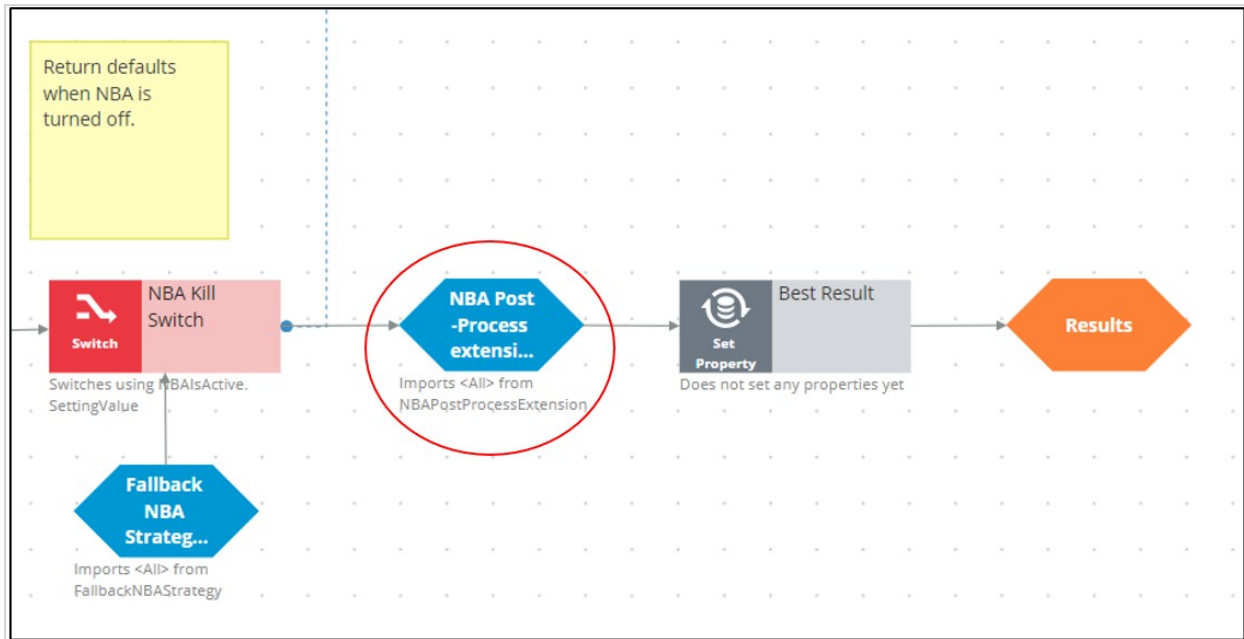
This is the NBA framework strategy when applied to each of the Actions.



The first component within the strategy framework is an extension point for any Action pre-processing you might need to perform.



The last functional component within the strategy framework is policy another extension point for any post-processing that must be performed.



Similarly, there are many other extension points such as the outbound limits extension points and business value extension points.

To ensure upgradeability, avoid overriding any part of the framework that is not a designated extension point.

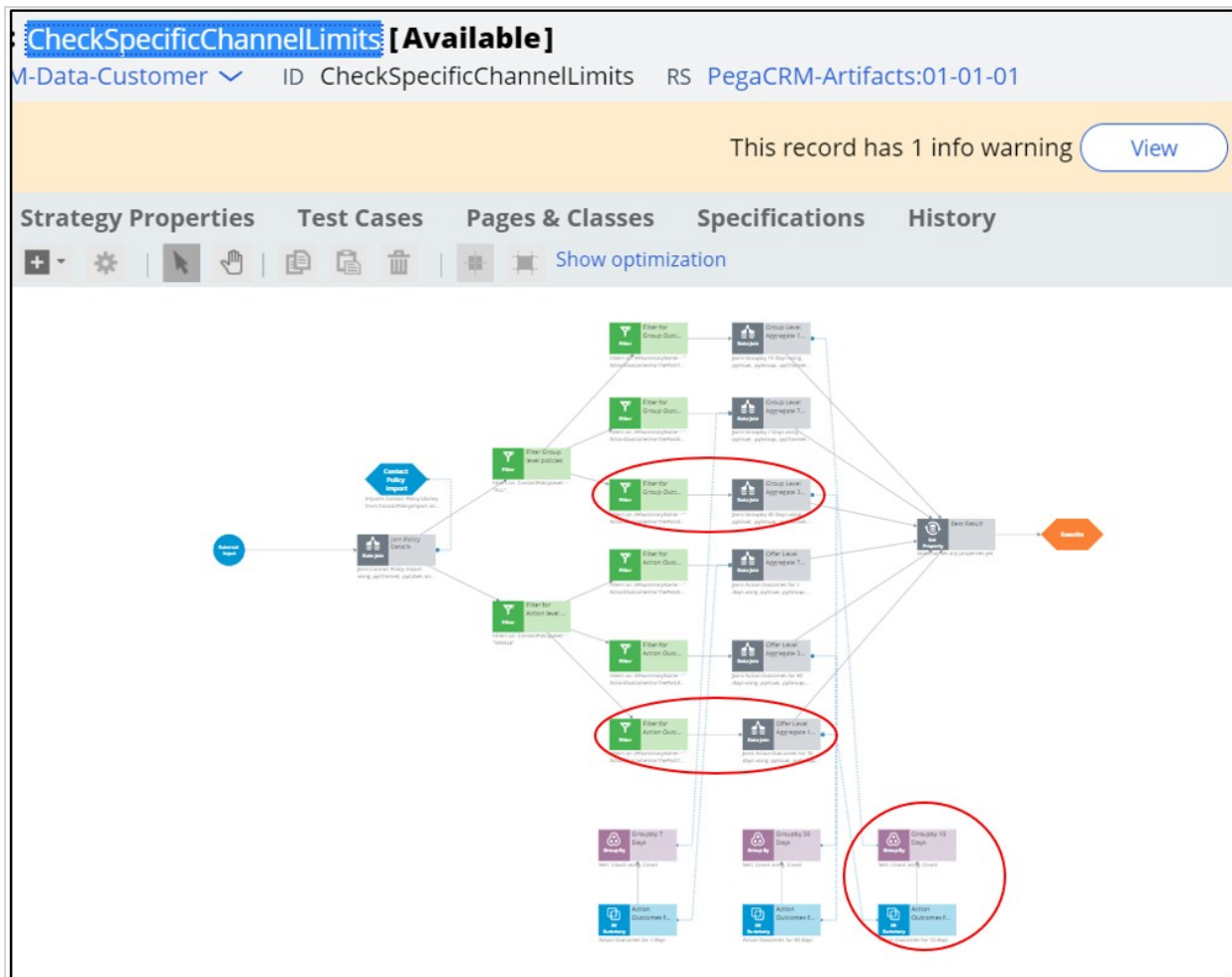
Also, the generated framework has some extension points where you can create strategies.

For example, while configuring values for Arbitration, you can specify a business value for an Action, or you can use a strategy to calculate the value. This can be done by adding a strategy to the existing framework.

Similarly, in defining the engagement rules, you can use a new strategy as a definition instead of an existing condition. Strategy designers can create such strategies from scratch using the decision strategy canvas.

Or, while defining the suppression rules, you can add a strategy to define new suppression rule limits instead of the existing 7 or 30 days.

For example, in the screenshot below, the CheckSpecificChannelLimits rule has been extended to have a 15-day limit:



In conclusion, the NBA Designer provides a guided and intuitive UI to bootstrap your application development with proven best practices. NBA designer generates the underlying strategies for you, which can be extended using existing values in the designated extension points or by building decision strategies from scratch, depending on the business requirement.

Content in 3 different ppts in artifacts folder

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.

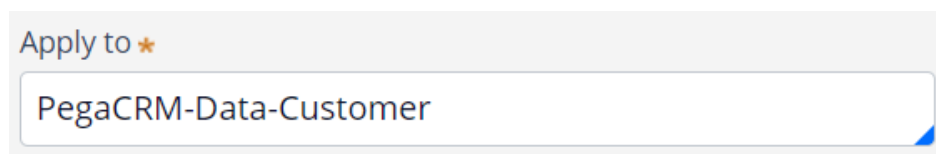
The Strategy-Results class limits the output of the strategy to the actions contained in the Business Issue and Group.

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.

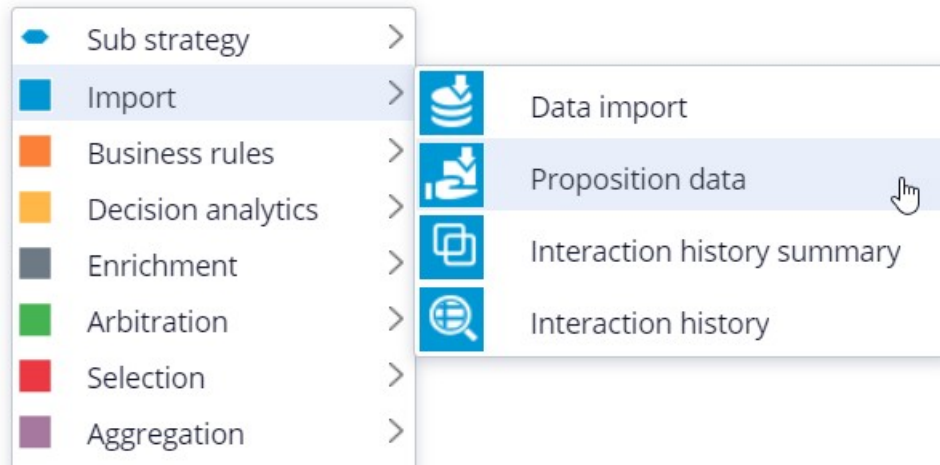


You can now start building the strategy. Right-click on the canvas to get the Context menu, which shows all component categories.

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.

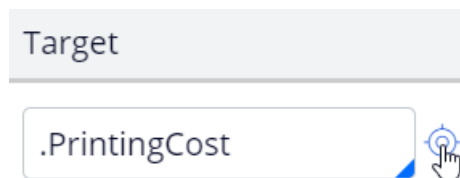
Click 'Add Item' to create the equation that will calculate the printing cost for each of the components.

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.



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 $\text{BaseCost} + 5 * \text{LetterCount}$.

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.

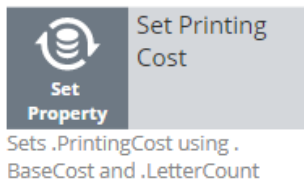
Finalize the Expression.



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.




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* 

Order by

Highest first (9 to 1)

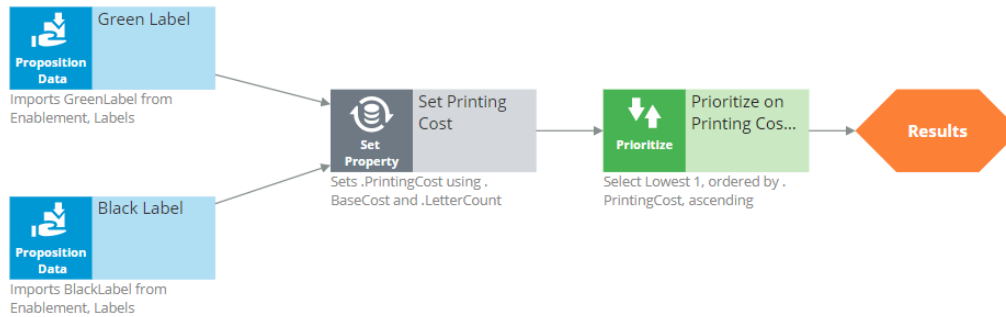
Lowest first (1 to 9)

Output

Top

All

Now you can connect the components and save the strategy.



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

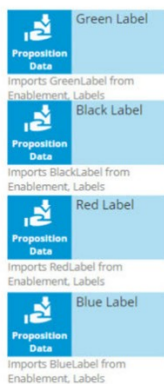
Using Pega Decision Management, you do not need to be an expert in programming, math or data science to design and execute sophisticated decision strategies that engage your customers throughout the customer journey. With its highly intuitive graphical canvas, Pega Decision Management enables you to 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.



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.

 Green Label

Imports GreenLabel from Enablement, Labels

 Black Label

Imports BlackLabel from Enablement, Labels

 Red Label

Imports RedLabel from Enablement, Labels

 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.

 Green Label

Imports GreenLabel from Enablement, Labels

 Black Label

Imports BlackLabel from Enablement, Labels

 Red Label

Imports RedLabel from Enablement, Labels

 Blue Label

Imports BlueLabel from Enablement, Labels

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10

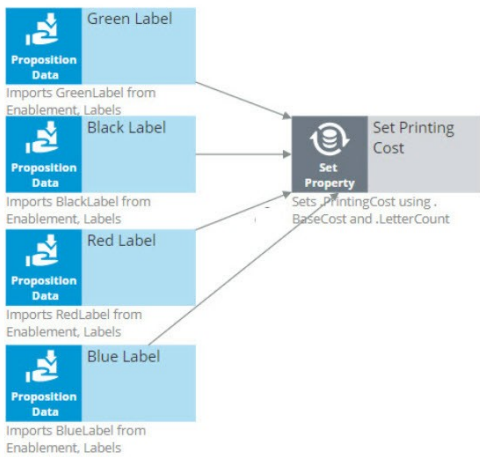
Rank	Name	BaseCost	LetterCount
1	Black Label	20	10

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.

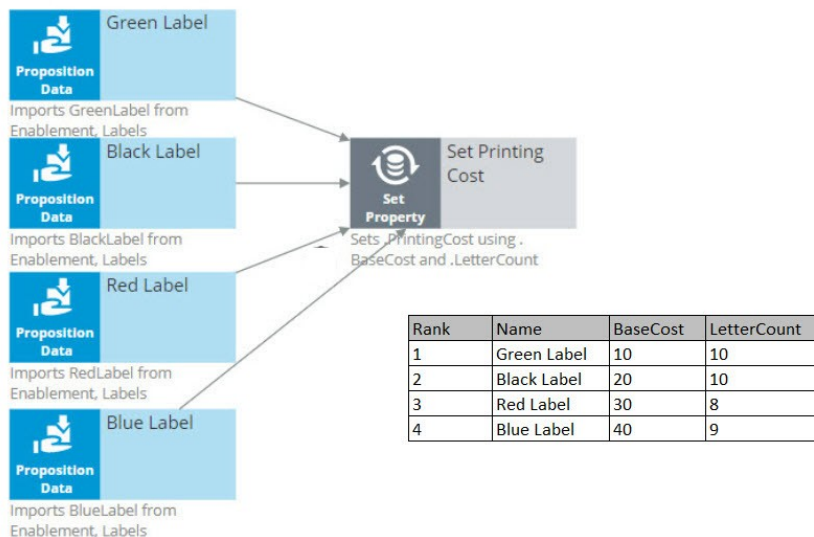
On the strategy canvas, components are connected by drawing arrows from component to component. So, what do these arrows mean exactly?



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.



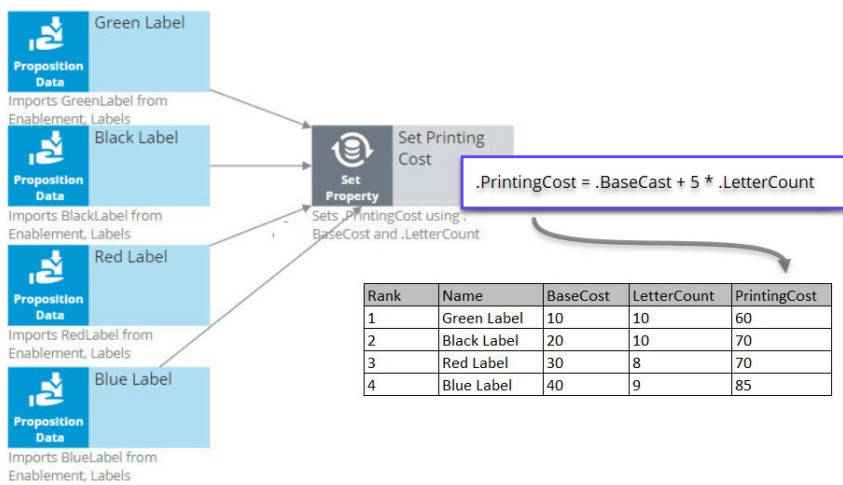
In the Set Property component, the Rank is determined by the order in which the actions are received by the component.

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.



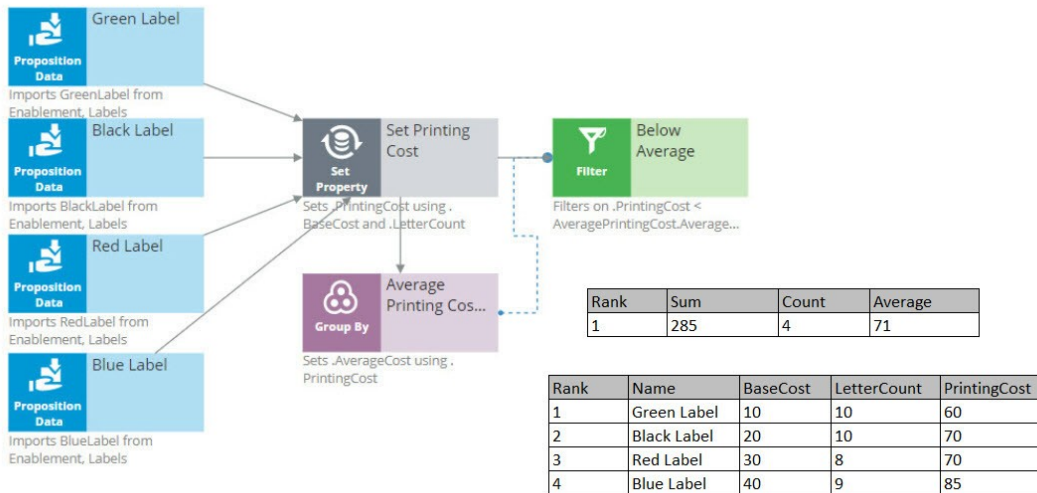
Because we are combining the data in our four Proposition Data components into one Set Property component, we only need to add one PrintingCost property to the new component, and it automatically computes the printing cost for all four actions.

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.

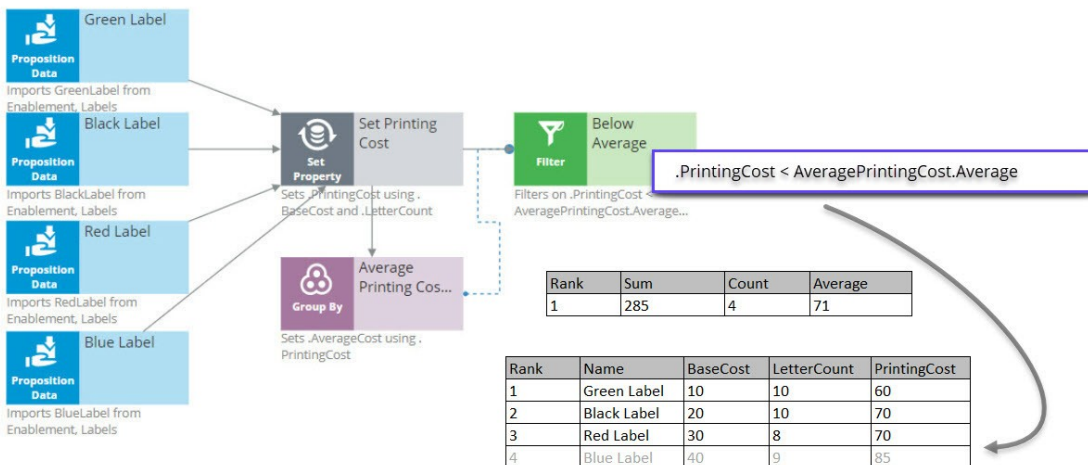
A 'Group by' component offers essential aggregation capabilities, like Sum and Count, that are used in many decision strategies. We will use it to calculate the average printing cost.



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.



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.

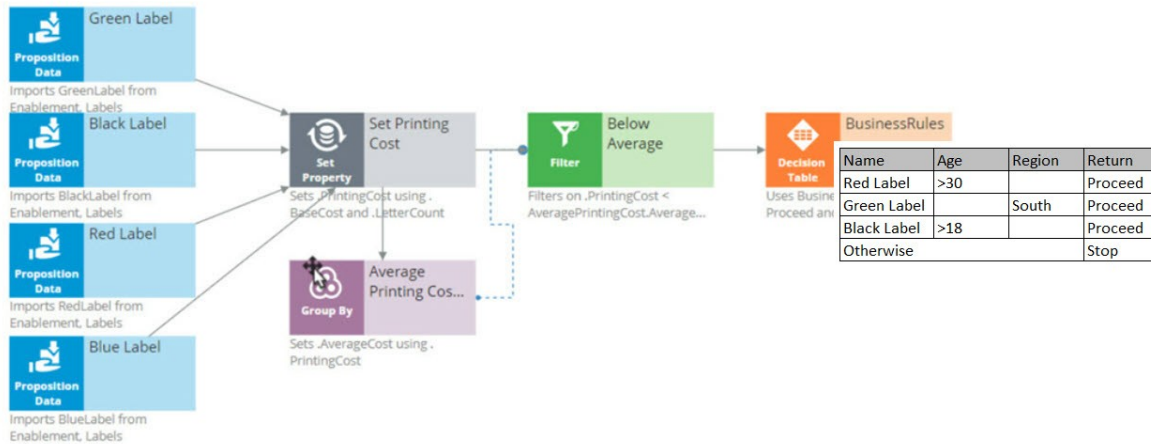
The Rank 1 average equals 71 in the 'Group by' component. This means that the filter will allow Label actions through that have a printing cost lower than 71.

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.



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

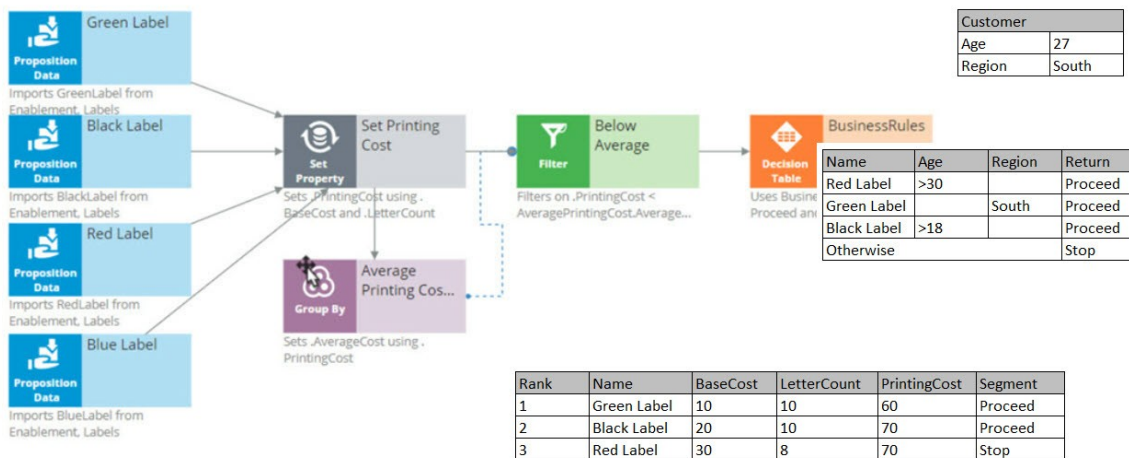
The business criteria say that the Red Label action can be offered if the customer's age is over 30 and they are from any region. If these criteria are met, the Return value is 'Proceed'.

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.



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.

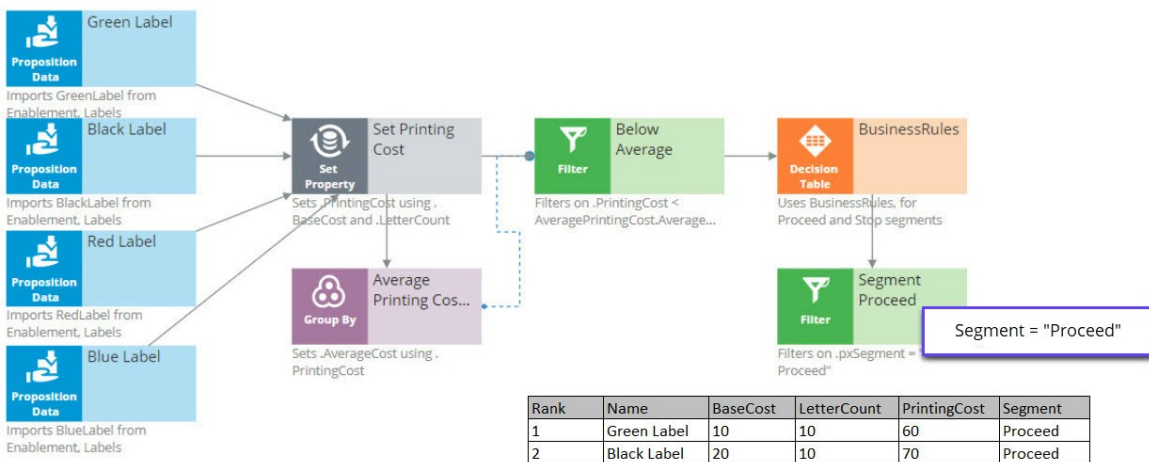
Next, it looks at the Black action and sees that the criteria for Black is that the customer's age is greater than 18. This customer is 27.

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.



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.

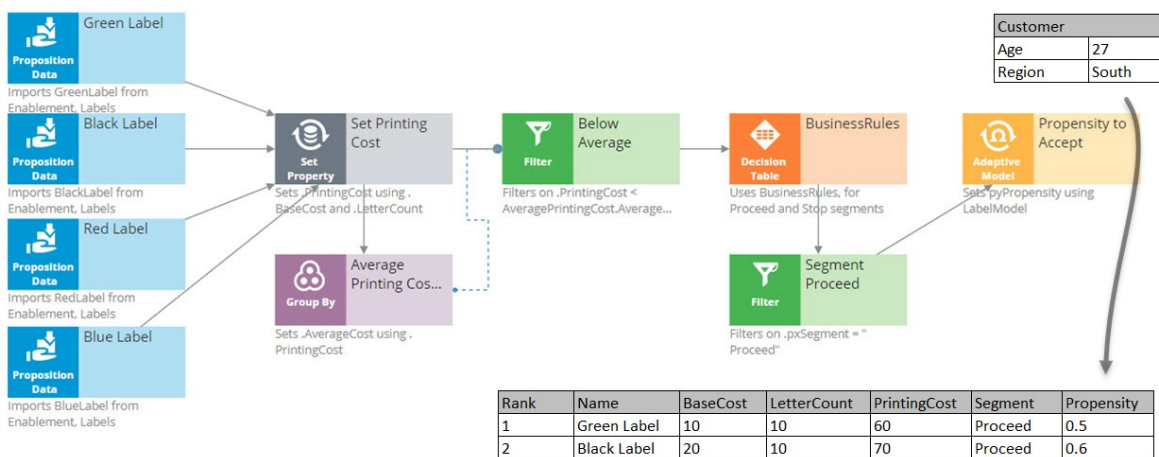
First is the Green Label. Green is allowed through, which means its properties will be available in the new component.

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.



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.

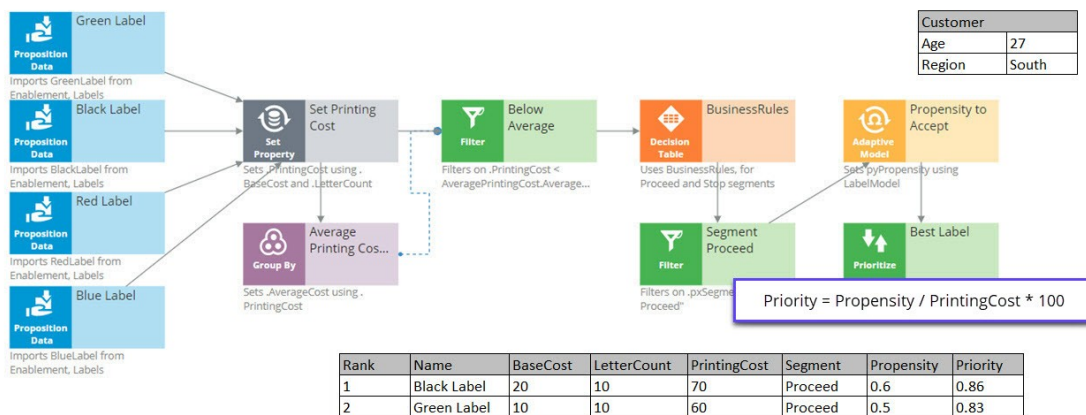
For the Black Label action, the likelihood turns out to be 0.6.

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.



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.

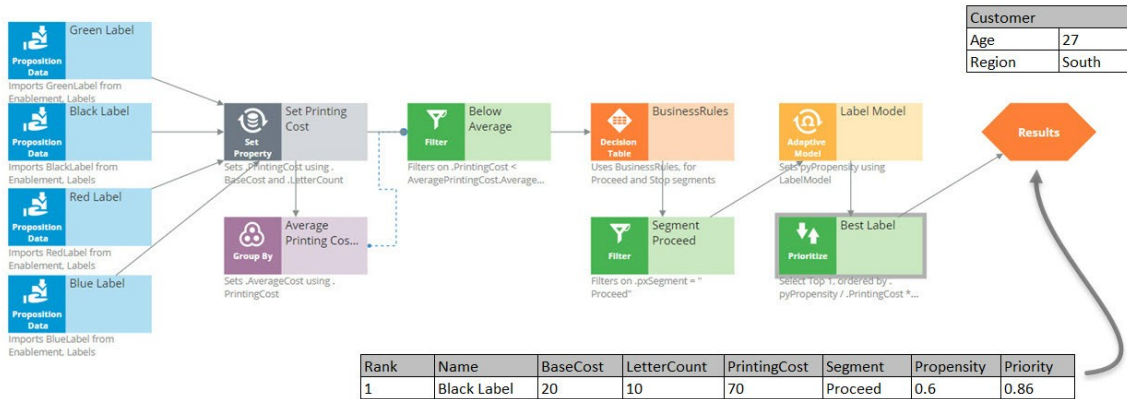
Because 0.86 is higher than 0.83, the Black Label action is now ranked number one.

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.



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. Input fields that are not directly available in the customer data model can be made 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.

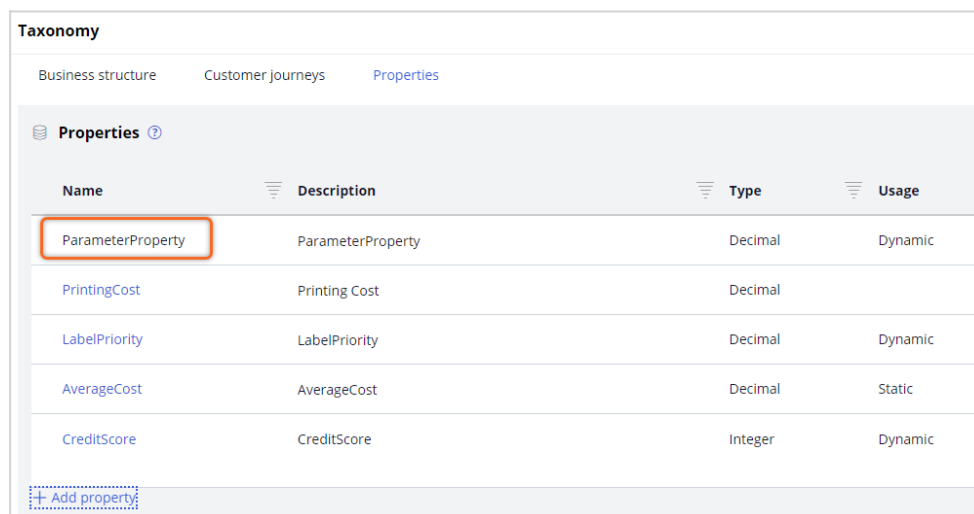
Video

@@@

Property to store a parameter

This demo shows you how to create parameterized predictors for expressions, offline model scores, and online model scores in Customer Decision Hub. In all three cases, the Data Scientist creates an SR property to store the parameter value, adds the property as a parameterized predictor to the prediction, and configures the pre-processing extension strategy.

In NBA Designer, you create a new property as part of the taxonomy. In this demo, we repurpose a generic *ParameterProperty*. The system automatically maps the property to the *CDH-Strategy Result* (or *CDH-SR*) class.



Name	Description	Type	Usage
ParameterProperty	ParameterProperty	Decimal	Dynamic
PrintingCost	Printing Cost	Decimal	
LabelPriority	LabelPriority	Decimal	Dynamic
AverageCost	AverageCost	Decimal	Static
CreditScore	CreditScore	Integer	Dynamic

Mapping of the parameter

Next, you map a new parameterized predictor in the prediction to *ParameterProperty* in Prediction Studio. The Predict Web Propensity prediction calculates the likelihood that a customer clicks on a web banner. Like *ParameterProperty*, the prediction results are saved to the *CDH-SR* class.

Prediction properties ✕

Prediction name

Outcome

Prediction output
 Predicting Propensity to Click for each combination of

	<input type="text" value=".pyIssue"/>	
and	<input type="text" value=".pyGroup"/>	
and	<input type="text" value=".pyName"/>	
and	<input type="text" value=".pyDirection"/>	
and	<input type="text" value=".pyChannel"/>	
and	<input type="text" value=".pyTreatment"/>	

[+ Add](#)

Save results to

CDH-SR

Data-pxStrategyResult

You then create a new parameterized predictor and map it to *ParameterProperty*.

Edit Parameters ✕

Parameterized predictors (5) Pages (1)

Map fields to parameterized predictors to supply this data to this model. < 1 2

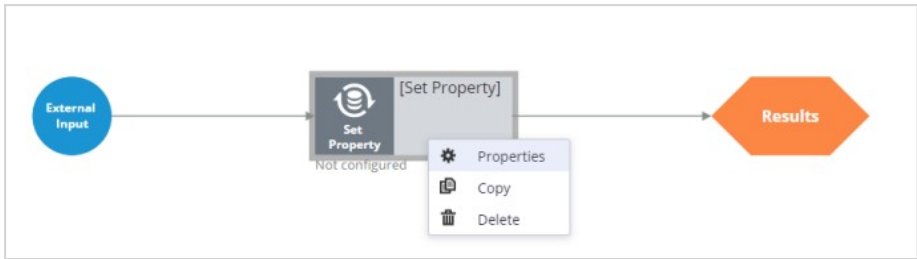
Predictor	Data type	Predictor type	Field
NewParameter	Decimal	numeric	.ParameterProperty

[+ Add parameterized predictor](#)

Configuring the extension point

To populate the parameter, you configure the NBA pre-processing extension point strategy. The pre-processing strategy runs before the prediction strategy so that the output of the extension strategy is available as input to the prediction. You save the strategy in the correct *Apply TO* class and then set the strategy result class to CDH-SR.

The first example of a parameterized predictor is an expression. You add a Set Property component to set the parameter propensity to be equal to 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.

Set property properties

Name*

Component ID Calculation

Description Use generated Use custom

Source components Target

Define action, target, and source

+ Add item X Delete

Action	Target	Source
Se	.ParameterProperty	@equal to @if(DecisionRequestCDH.Ct

Expression builder Browse Test

```
1 @if(DecisionRequestCDH.Customer.FSClickstream.I
nvestmentPageVisitsLast90Days=0,0, divide(Decis
ionRequestCDH.Customer.FSClickstream.Investment
PageVisitsLast30Days, DecisionRequestCDH.Custome
r.FSClickstream.InvestmentPageVisitsLast90Day
s))
```

Otherwise, it returns the ratio of the Investment page visits in the last 30 days to the last 90 days. A high value may indicate the increasing interest of the customer in the content of this page.

The second example of a parameterized predictor is offline model scores. The data scientist team might produce many product group scores for each customer and each channel. A Decisioning Architect creates a database table to accommodate the data. A data set, associated to the customer context, makes the data available as a data source. Notice that 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

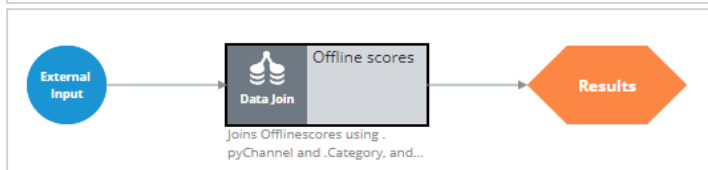
Exclude source component results that do not match join condition.

You then configure the component to populate the Parameter Property with the model scores.

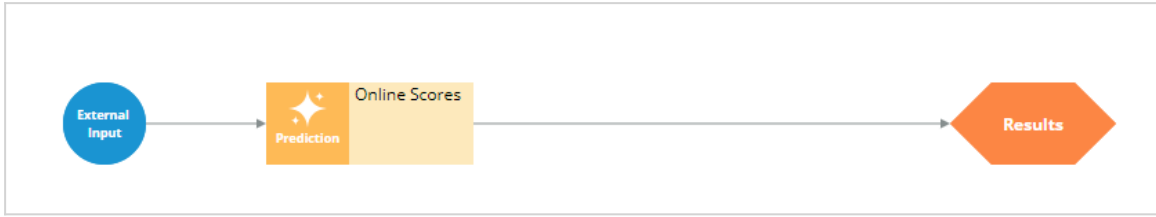
Define mapping

+ Add item × Delete

Target		Source (Offlinescores)
:: Set	.ParameterProperty	equal to Offlinescores.Score



The third example of a parameterized predictor is the score of a churn prediction running 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 that references the churn prediction to the pre-processing extension strategy.



Set the Parameter Property to equal the propensity calculated by the churn prediction.

Define action, target, and source

+ Add item × Delete

Action	Target	Source
Se	.ParameterProperty	equal to .pyPropensity

Run the strategy to verify that Parameter Property is populated.

Outcome weight	0.0
Parameter Property	0.72017587 60057334
Placement type	Tile
Previous component	Online Scores
Priority	25.8181156 1903224
Product category	---
ProductID	---
Propensity	0.72017587 60057334

To summarize, creating parameterized predictors for adaptive models involves:

- Creating an SR property to store the parameter value
- Adding the property as a parameterized predictor to the prediction
- Configuring the pre-processing extension strategy

Using predictions in engagement strategies

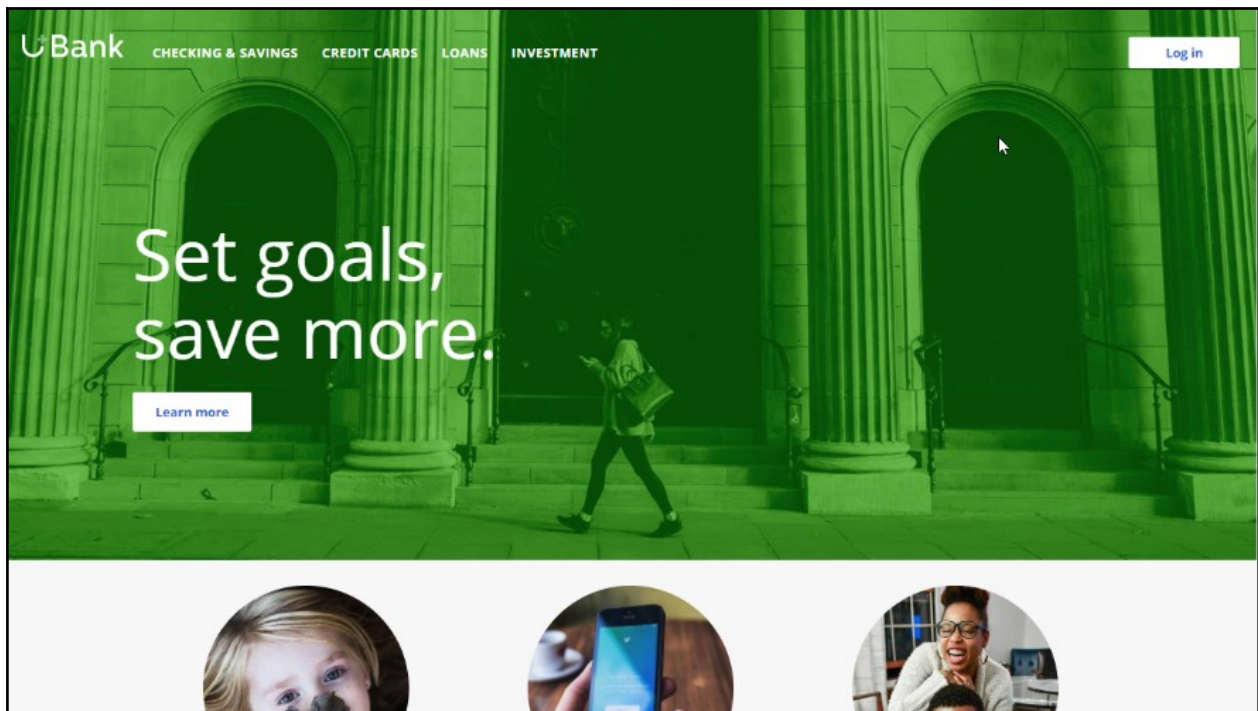
Introduction

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

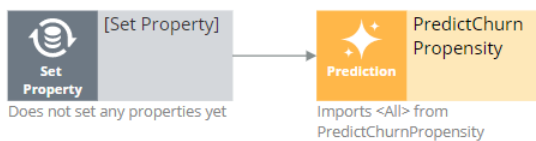
Labels for the possible values of the responses.

Churn

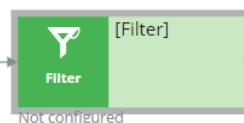
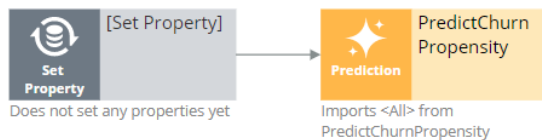
Target label Alternative label

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.



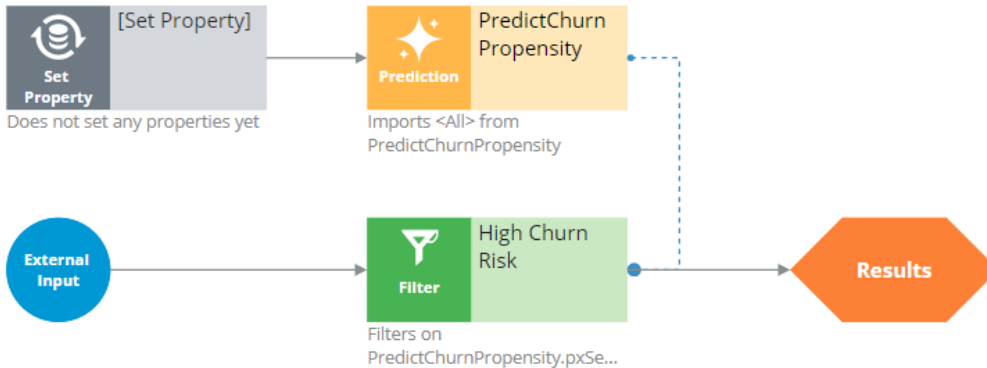
The filter condition is defined to output a retention offer when the pxSegment property of the prediction is equal to **Churn**.

Expression builder

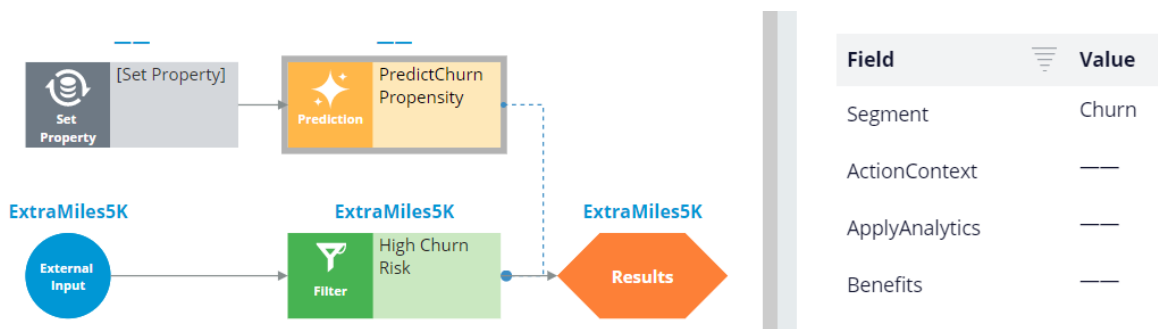
Browse

Test

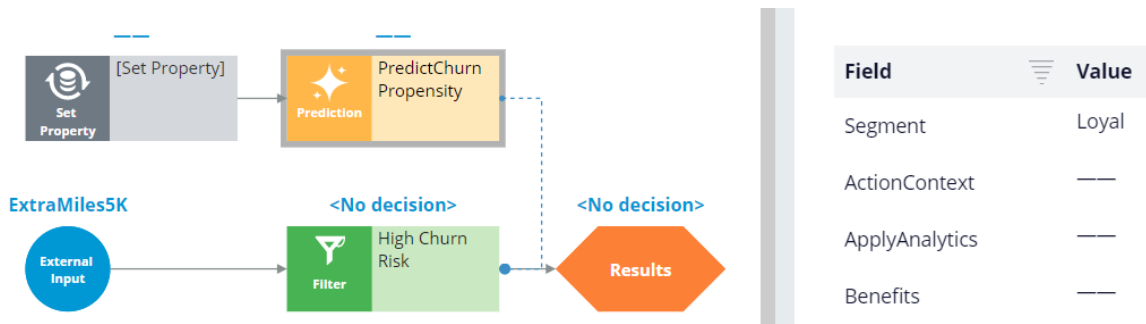
```
1 PredictChurnPropensity.pxSegment="Churn"
```



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**.



The strategy does not have a result for **Barbara**, because the Segment value is **Loyal**.



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.

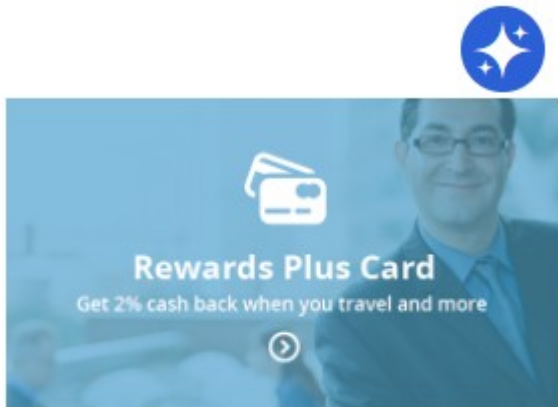


Extra miles 5K

5,000 extra miles

[Learn more](#)

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

[Learn 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.

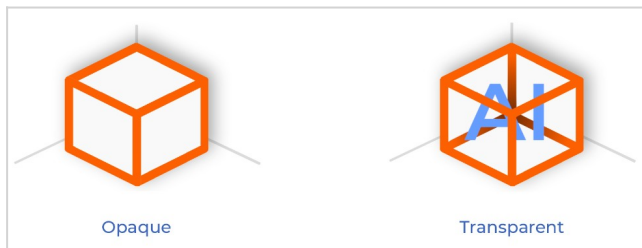
Transcript

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 4 All Business Issues	Adaptive model - Gradient Boosting Pega 1 Compliance Nurture, Retain, Service	Bivariate model Pega Compliance 3 All Business Issues	Genetic algorithm Pega Compliance 2 All Business Issues	Regression Compliance 4 All Business Issues
Clustering model Compliance 3 All Business Issues	Ensemble model Compliance 1 Nurture, Retain, Service	General regression Compliance 4 All Business Issues	k-Nearest neighbors model Compliance 1 Nurture, Retain, Service	Naive Bayes model Compliance 2 All Business Issues
Random forest Compliance 1 Nurture, Retain, Service	Scorecard Compliance 5 All Business Issues	Support Vector Machine Compliance 1 Nurture, Retain, Service	GBM and XGBoost Compliance 1 Nurture, Retain, Service	Unknown model type Compliance 1 Nurture, Retain, Service

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.

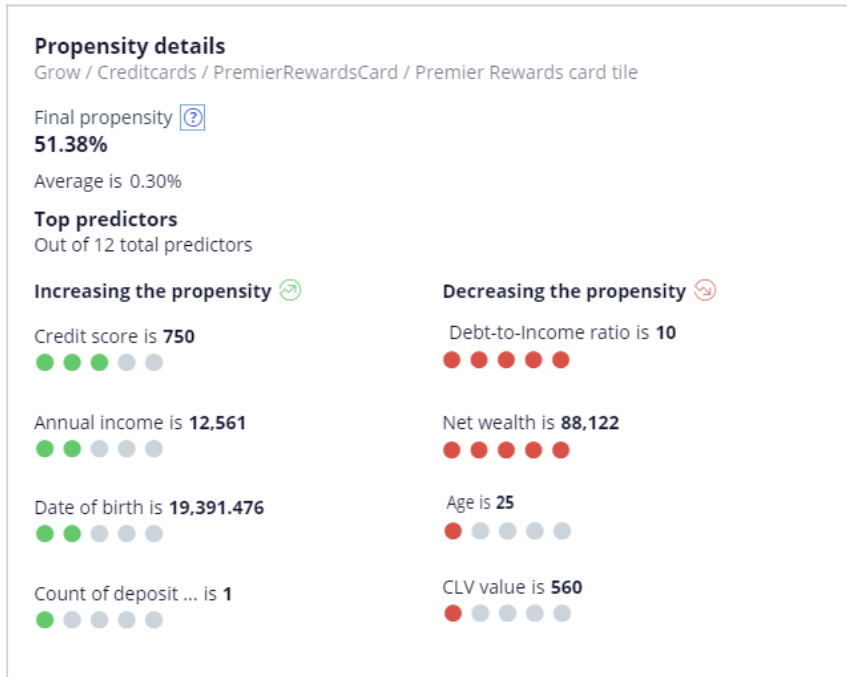
Transcript

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.



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.

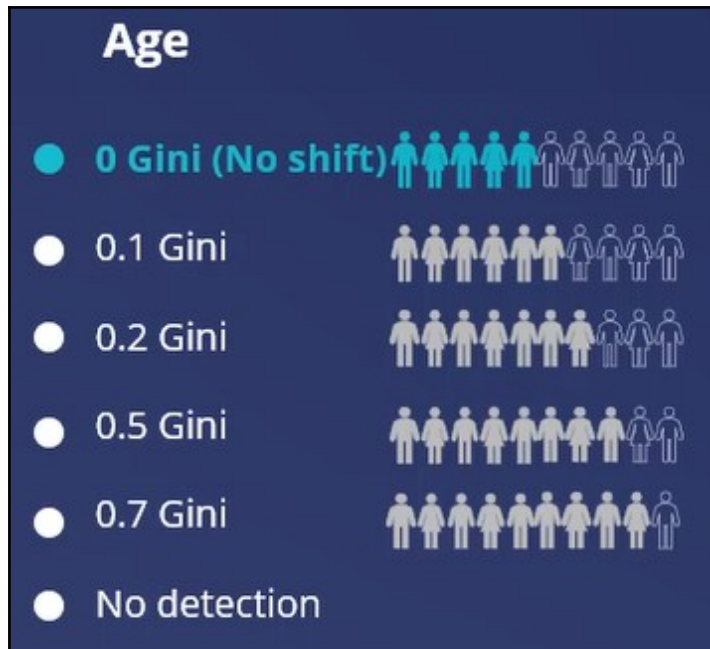
Transcript

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.

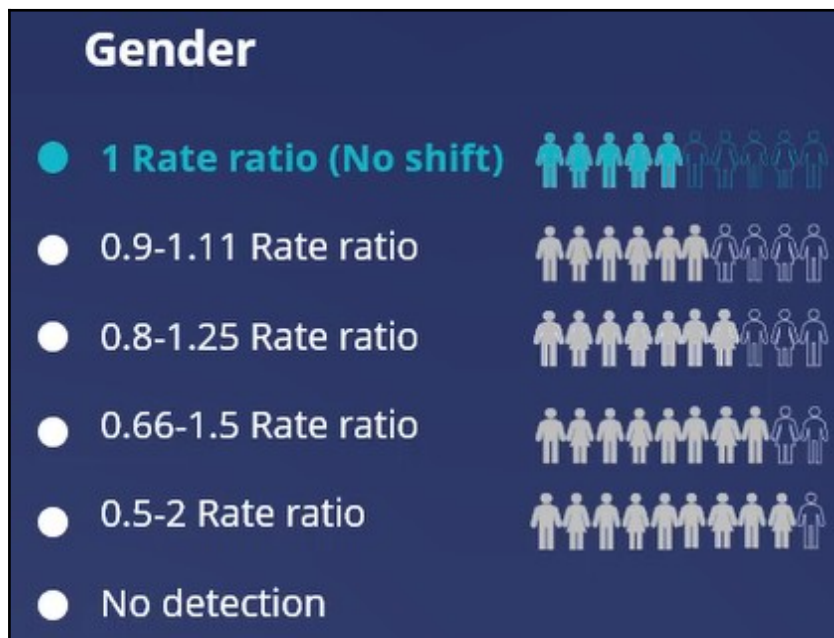
An ethical bias policy forms the base of an ethical bias simulation. This policy will include the bias fields and the thresholds for each field. You can include any property from your customer class. For instance, age and gender properties are typical properties you might want to include in bias testing.

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.



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.



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 **Bias threshold**

Bias fields

Define fields for which bias will be checked.

Context

Data-Decision-Request-Customer

[+ Add bias field](#)

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 Configure

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 Edit report Actions X

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

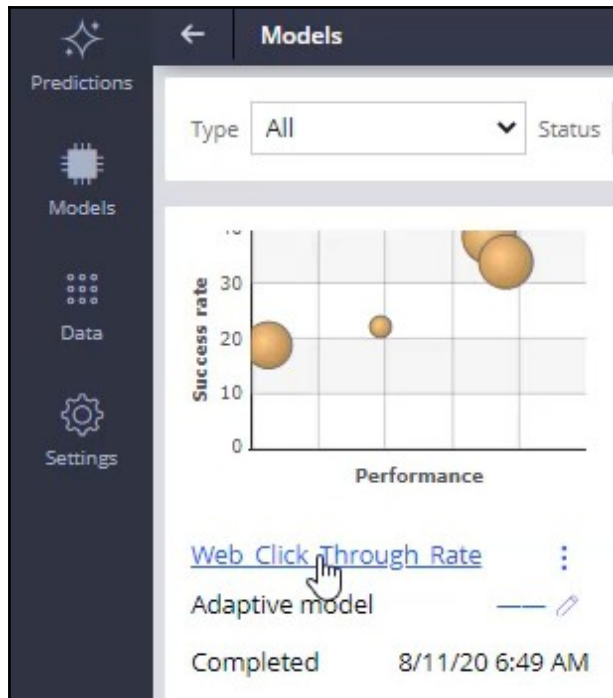
Sampled Customers

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.

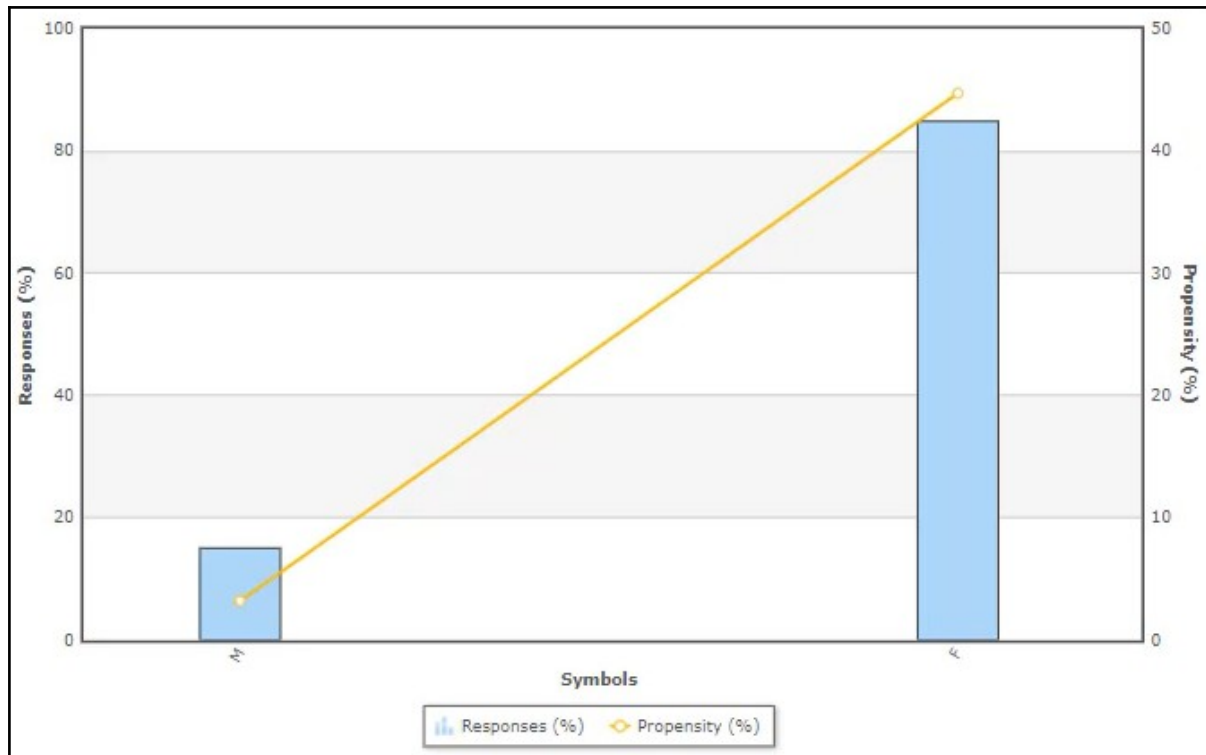


Now, open the Premier rewards card model report to view the predictors.

Predictors	Score distribution	Trend			
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.