Pega Data Scientist

STUDENT GUIDE

PEGA Academy

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

Mission: Data Scientist
Products: Pega Platform™ 8.7, Pega Customer Decision Hub™, Pega Customer Service™
URL: https://academy.pega.com/mission/data-scientist/v4
Date: 09 May 2022

Contents

Pega AI overview....5
- Predictive models drive predictions....6
- Prediction Studio....10

Customer Decision Hub overview....19
- Next-Best-Action paradigm....20
- One-to-one customer engagement paradigm....25
- Action arbitration....31
- Next-Best-Action Designer....37

Creating and understanding decision strategies....44
- Decision strategies....45
- Decision strategy canvas....50
- Creating a decision strategy....58
- Decision strategy execution....64

Creating predictions....75
- Creating a prediction....76
- Using predictions in engagement strategies....78

Creating predictive models....84
- Predictive models....85
- Building models with Pega machine learning....89
- Importing predictive models....97
- Using machine learning services....102
- Model transparency....109

MLOps....111
- MLOps process....112
- Placing a predictive model in shadow mode....116

Adaptive analytics overview....119
- Adaptive analytics....120
- Predictors and outcomes of an adaptive model....126

Advanced settings of an adaptive model....130
Adaptive model outputs....132
Optimizing AI in the NBA framework....133
Configuring an adaptive model....134
Creating parameterized predictors....141
Monitoring adaptive models....149
Regular monitoring of adaptive models....150
Inspecting adaptive models....152
Exporting historical data....159
The impact of machine learning....164
Measuring lift using a control group....165
Pega Process AI overview....171
Pega Process AI overview....172
Predicting fraud....176
Predicting fraud....177
Predicting case completion....183
Predicting case completion....184
Text analytics for email routing....189
Text analytics....190
Email routing....195
Training a topic model to improve email routing....201

Pega AI overview

Description

Prediction Studio is the dedicated workspace for data scientists to control the life cycles of predictions and the predictive models that drive them. Prediction Studio offers prediction and model reports that allow users to monitor and spot predictions and models that underperform.

Learning objectives

- Describe how predictive models drive case management predictions, Pega Customer Decision Hub™ predictions, and text analytics predictions
- Describe the purpose of the control group in Customer Decision Hub predictions
- Describe the purpose of the work areas in Prediction Studio
- Recognize the transparency settings for predictive models

Predictive models drive predictions

With the decision management capability of Pega Platform™, you can enhance applications to help optimize business processes, predict customer behavior, analyze natural language, and make informed decisions to better meet customers' needs and to achieve positive business outcomes.

Transcript

This video introduces you to Pega AI, a feature of the decision management capability of Pega Platform™.

Other decisioning features of the Pega Platform include:

- Decision strategies to improve customer experience and deploy intelligent processes based on behavioral and operational data and data sets to read and write the data used in the decision strategies.
- You can use event strategies to detect patterns in data streams and react to them.
- And to ingest, process, and move data from one or more sources to one or more destinations, you can configure data flows as scalable and resilient data pipelines.

Decision management uses Pega AI to make predictions about customer behavior, successful case completion, the topic of an incoming message, or other subjects to make the decisions more relevant.

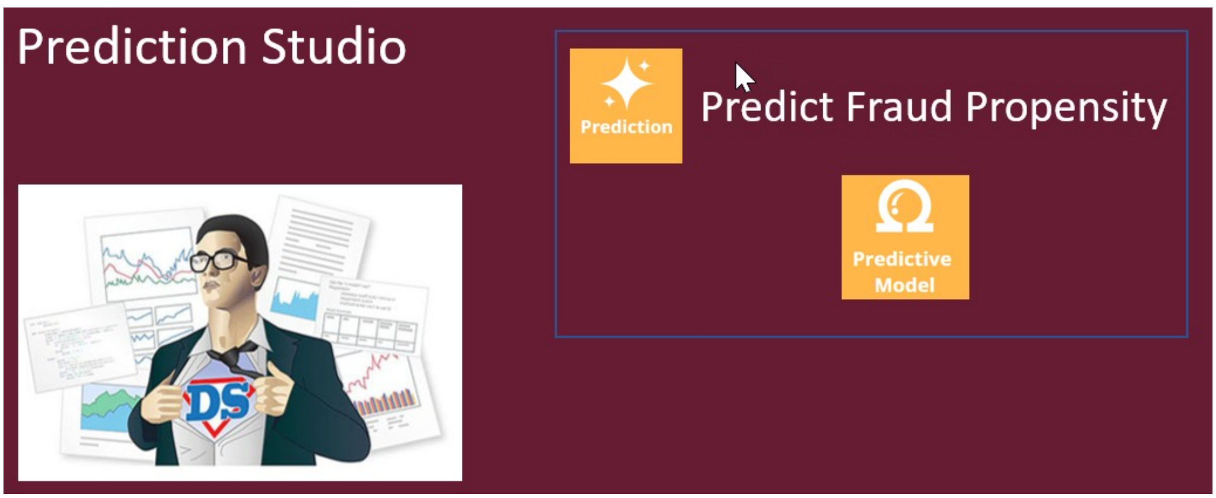
Decision management is a Pega Platform capability. You can apply decision management to any application that is built on Pega Platform.

Predictions differ to suit the domain they are used in, but one or more predictive models drive them all.

A data scientist can create a predictive model in Pega Platform or an external environment that can export the model as a PMML or H2O file. Another option is to connect to a machine learning service such as Google ML or AWS SageMaker.

If an insurance company wants to use Pega Process AI™ to route incoming claims that might be fraudulent to an expert based on the outcome of a predictive model …

… the data scientist creates a fraud model to drive a new case management prediction in Prediction Studio.



Prediction Studio is the dedicated workspace where you manage the life cycle of predictive models and the predictions they drive.

A prediction is a hand-off to an application developer, who can then use the prediction in a decision step in the case type to route cases more accurately. This strengthens the separation of concerns.

You can use Pega Customer Decision Hub™ to make next-best-action decisions for your customers.

Customer Decision Hub predictions can predict customer behavior, such as which customer is about to churn …

… or predict the likelihood that a customer clicks on a web banner to support the decision on which banner to show to a customer.

Pega Adaptive Decision Manager (ADM), a key component of the decision management capability …

… allows a data scientist to configure self-learning, adaptive models that continuously improve predictions about business processes and customer behavior.

An adaptive model rule typically represents many adaptive model instances because each unique combination of the model context generates a model.

In Customer Decision Hub, adaptive models drive many predictions that come with the product out of the box, such as the Predict Web Propensity prediction that predicts the likelihood that a customer clicks a web banner.

Customer Decision Hub predictions have several features specific for the domain …

… such as a control group for which the prediction outputs a random propensity instead of the propensity that is generated by the adaptive model instance.

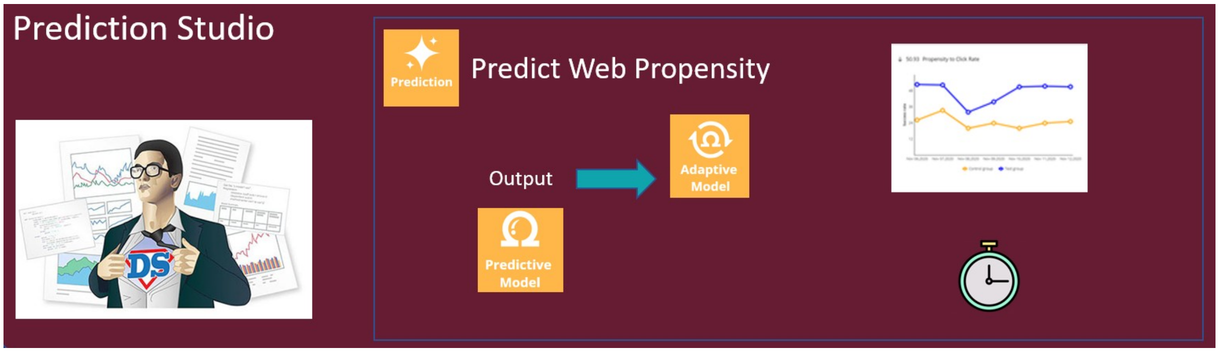
Comparison of the control group and the model propensity-based group allows you to measure the lift in a success rate that the AI generates, an important business metric.

Also, Customer Decision Hub predictions feature a response timeout setting. After the timeout expires, a negative response is recorded.

The response timeout setting depends on the use case. For example, in a web use case, several minutes suffice …

… while in an outbound email campaign, the response timeout is set to several days to allow customers enough time to respond.

You can further enhance the prediction by using the output of a predictive model as a predictor in the adaptive model.



The Pega Customer Service™ application uses the natural language processing capability of decision management to analyze incoming text and route the messages based on the topics and entities detected.

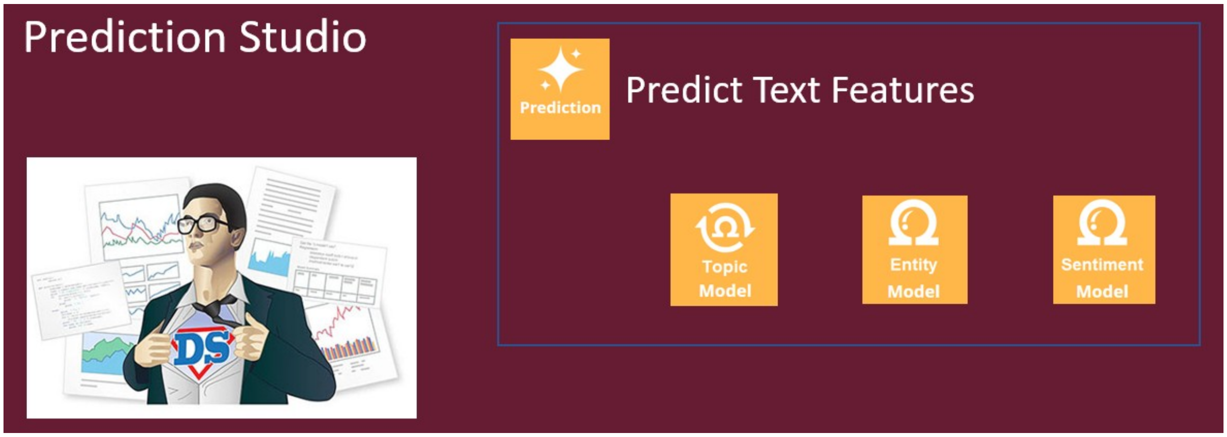
Pega Customer Service uses Text analytics predictions that are distinctly different from both case management predictions and Customer Decision Hub predictions.

Text analytics predictions use predictive models to detect the topic of an incoming message that the application can use to optimize the routing of the message to the relevant department.

Secondly, text analytics predictions use entity extraction models that qualify text as, for example, an account number, a postal ZIP code, or an address.

The application can use this information to fill relevant fields in a case automatically.

Finally, the text analytics predictions come with a sentiment model that can route or prioritize negative messages to improve the customer experience.



Feedback on the detected topics, entities, and sentiment by CSRs improves the performance of the text analytics prediction over time.

This video has concluded. What did it show you?

- Pega AI allows you to improve business processes and customer engagement by using predictions.
- Predictive models drive the predictions.
- The predictive models can be static or adaptive.
- Predictions are managed in Prediction Studio.

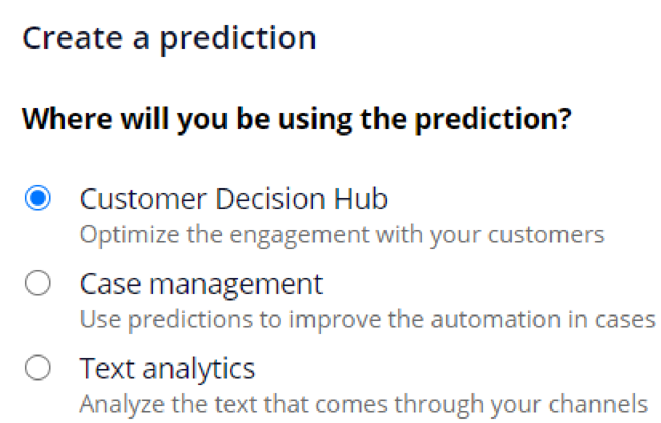
Prediction Studio

Predictions and the predictive models that drive them are created, monitored and updated in Prediction Studio, the dedicated workspace for data scientists.

Transcript

This video gives you an overview of the features of Prediction Studio. The workspace provides data scientists with everything they need to author, deploy, govern, monitor, and change predictions. Prediction Studio has four work areas: Predictions, Models, Data, and Settings.

The Predictions landing page is used to create and manage predictions. Predictions can be one of three types.



Customer Decision Hub predictions are used in the Pega Customer Decision Hub™ application to optimize 1:1 customer engagement. **Case management predictions** are used in case types to support decisions in business processes and **Text analytics**

predictions are used in the Pega Customer Service™ application to predict the topic of incoming messages. The three types of predictions differ to suit the domain they are used in, but one or more predictive models drive them all.

The Model landing page is used to create and manage the predictive models. There are four types of predictive models.

Static **predictive** models are built on historical data. A data scientist can create a predictive model in an external tool and import the model file. Another option is to connect to a machine learning service, such as Google ML or Amazon SageMaker.

Predictive model

Adaptive models continuously learn from responses and adapt to changes over time. You can configure an adaptive model rule that typically represents many adaptive model instances, because each unique combination of the model context will generate a model.

Adaptive model

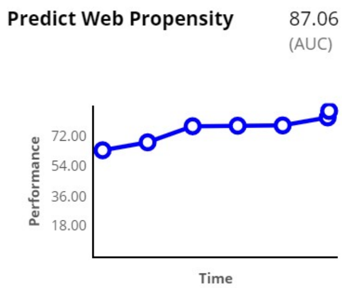
Text categorization models can detect the topic of a message and the sentiment of the author. **Text extraction** models identify entities such as an email address, an account number, or a city.

Text categorization

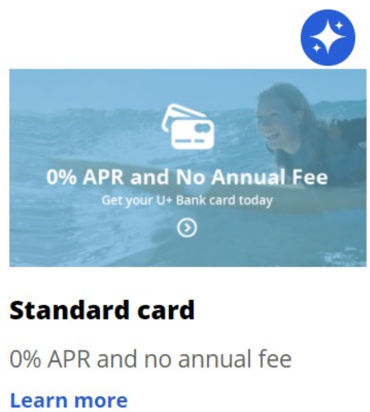
Text extraction

Many widely used Customer Decision Hub predictions ship with the product.

One of these is the **Predict web propensity** prediction, which predicts the likelihood that a customer clicks a web banner.



Consider, for example, the cross-sell in a web scenario for U+ Bank. The bank shows a personalized credit card offer to eligible customers when they log in to the bank's website.



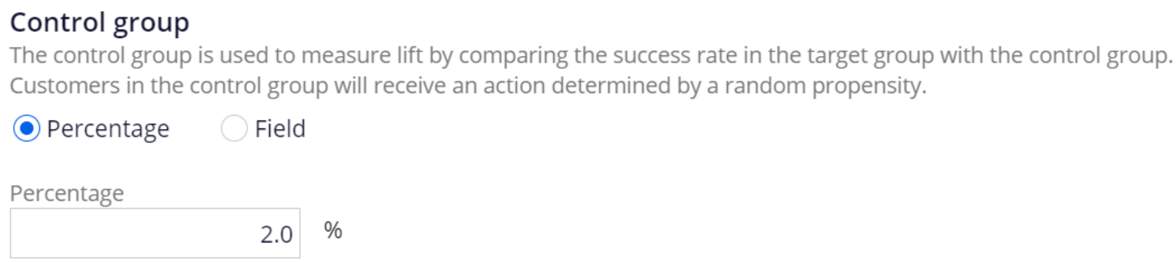
When a customer is eligible for multiple credit cards, the prediction calculates the propensity of receiving a positive response from the customer for each card. Customer Decision Hub decides which credit card to offer based on business rules, interaction context, and predictions.

The adaptive model that drives the **Predict Web Propensity** prediction is the **Web Click Through Rate** model.

Supporting models

Name	Component name	Type	Performance	Status
Web_Click_Through_Rate	Web_Click_Through_Rate_Customers	Adaptive model	68.55 AUC	ACTIVE
Web_Click_Through_Rate	Web_Click_Through_Rate_Accounts	Adaptive model	68.55 AUC	ACTIVE

You can configure several aspects of a Customer Decision Hub prediction. A control group is a small group of customers who receive random offers, as opposed to the target group.



Customers in the target group receive the offers that they are most likely to accept, based on the propensity that the prediction calculates for each customer.

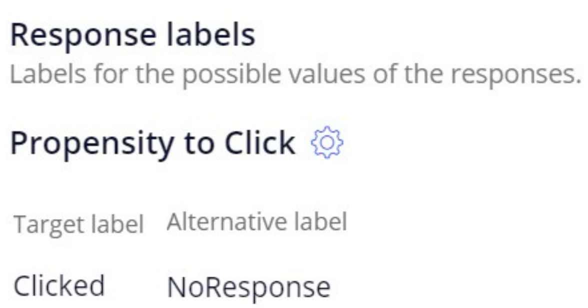
The purpose of the control group is to calculate lift by comparing the success rate in the control group with the success rate in the target group.

The random offers also allow predictive models to continuously explore all actions.

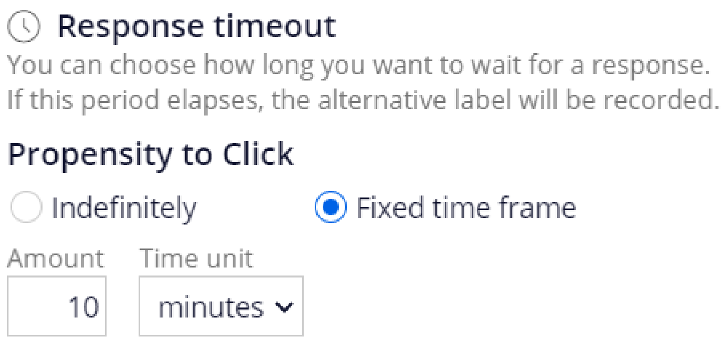
Based on the lift, you can determine the effectiveness of your prediction, for example, in increasing conversion rates. The control group is typically defined as 2% of all customers, but this can be changed.

The response labels represent the possible outcomes of a prediction. The propensity is computed based on the number of outcomes registered under the target label versus the alternative label.

For example, in **Predict Web Propensity**, because you want to predict the likelihood of a customer clicking a banner, the Target label (which in this case represents the positive outcome) is **Clicked**. The alternative label that represents a negative outcome in this case, is **NoResponse**.

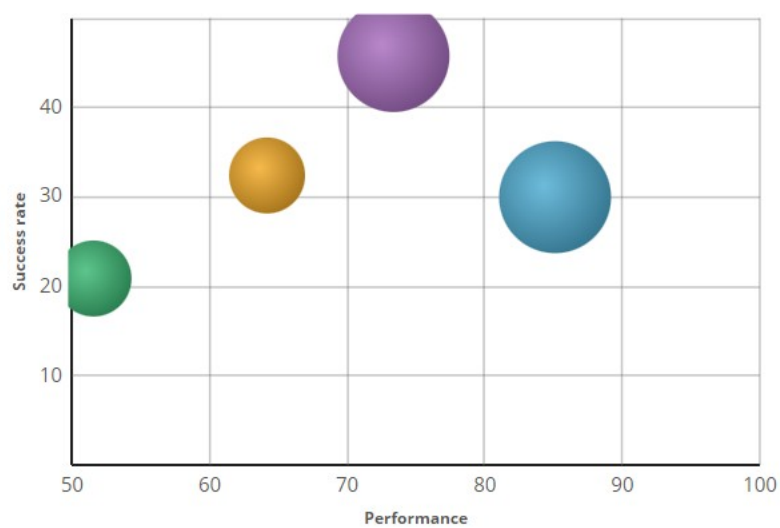


The **NoResponse** response can be captured on request or automatically depending on the response time-out setting. The response time-out defines how long to wait for the customer to respond to your offer. After the specified amount of time elapses, the system automatically records the alternative outcome for the interaction.



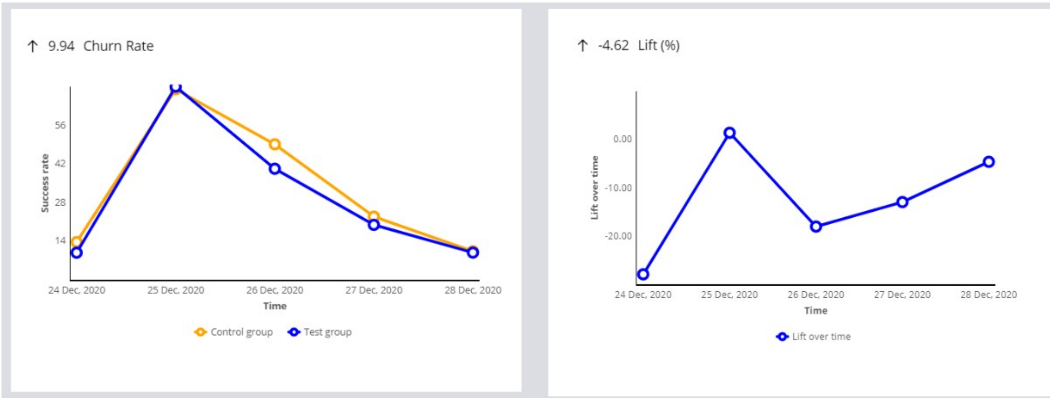
This setting depends on the use case. When predicting a click on a web banner, you typically set it to 30 minutes or less, but in an outbound email offer, a waiting time of several days is more appropriate.

The **Web Click Through Rate** model rule is the model template for each of the credit card offers. You can monitor the performance of the models in a diagram that shows the success rate versus model performance.



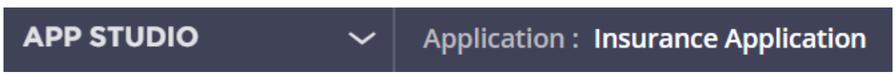
The models are represented by colored circles. The size of the circles indicates the number of responses captured by the model.

Monitor the prediction over time to analyze how successful it is. The available metrics are the success rate, the lift calculated using the control group, the prediction performance, and the total number of cases.

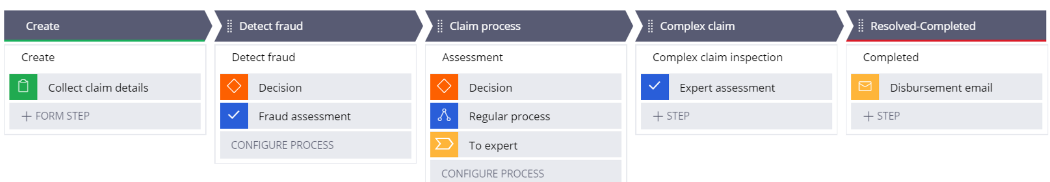


Prediction Studio generates actionable insights and notifies the user when predictions and predictive models show unexpected behavior (for example, a significant drop in success rate).

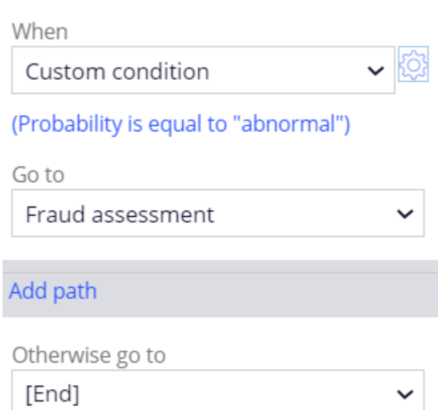
Case management predictions support decisions in a case type.



Consider this case type, which handles incoming car insurance claims.



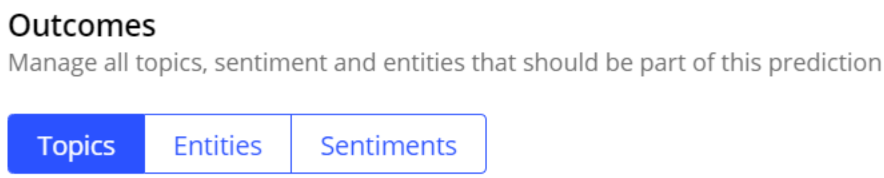
An application developer can use the outcome of the prediction in the condition of a decision step, instead of a business rule. Based on the condition, a case is routed to a fraud expert when the prediction flags the claim as abnormal.



The Pega Customer Service application can use text predictions to analyze messages that come in through various channels, such as email and chat channels.

A text prediction is automatically generated for each new channel.

A text prediction detects topics, entities, and sentiment, to improve the routing of messages to the appropriate department.



Topics can be detected based on keywords or machine learning. In this example, topic detection is keyword based. It is highly recommended to include machine learning in topic detection.

Consider the following message, about an address change:

Hi U+ Bank,

I have noticed, in my last account statement, you have used a wrong address. Please change my mailing address to read: 222 West Las Colinas Blvd., Irving, TX 75039, USA, effective immediately. And I'm happy to have a fresh email address: sara@gmail.com.

Cheers, Sara Connors

Based on keywords, two topics are detected. Both topics have a confidence score of 1, so it is not possible to determine the correct topic.

Topic
Action > Complaint
Action > Account Address Change

To train the topic model, use a data set with records that contain a message and the associated topic. When the trained model is tested with the same message, the model correctly generates the highest confidence score for the address change topic.

	Sentiment	Sentiment score	Model name	Model type	Confidence
Account Address Change	Positive	0.41	U+ Bank customer support	Pega NLP	0.71
Complaint	Positive	0.41	U+ Bank customer support	Pega NLP	0.29

The sentiment model is shipped with the product and predicts an overall positive sentiment.

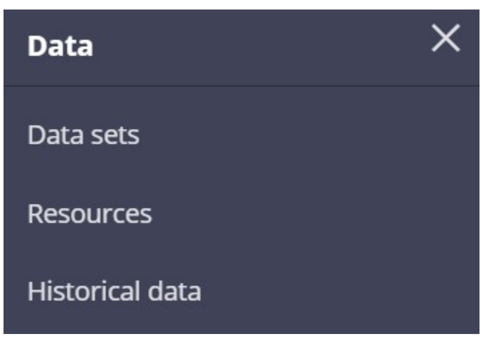
Output

Hi U+ Bank, I have noticed, in my last account statement, you have used a wrong address. Please change my mailing address to read: 222 West Las Colinas Blvd., Irving, TX 75039, USA, effective immediately. And I'm happy to have a fresh email address: sara@gmail.com. Cheers, Sara Connors

Entity extraction can be based on keywords, machine learning, or RUTA scripts.

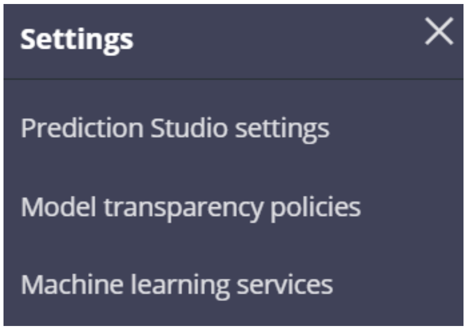
Value	Name	Type	Method	Resolved value
Irving,	City	City	Machine learning	Irving,
sara@gmail.com	Email	Email	RUTA	sara@gmail.com
TX	State	State	Machine learning	TX
75039	Zipcode	ZipCode	Machine learning	75039

The **Data** work area is used to define data sets, resources, and historical data.



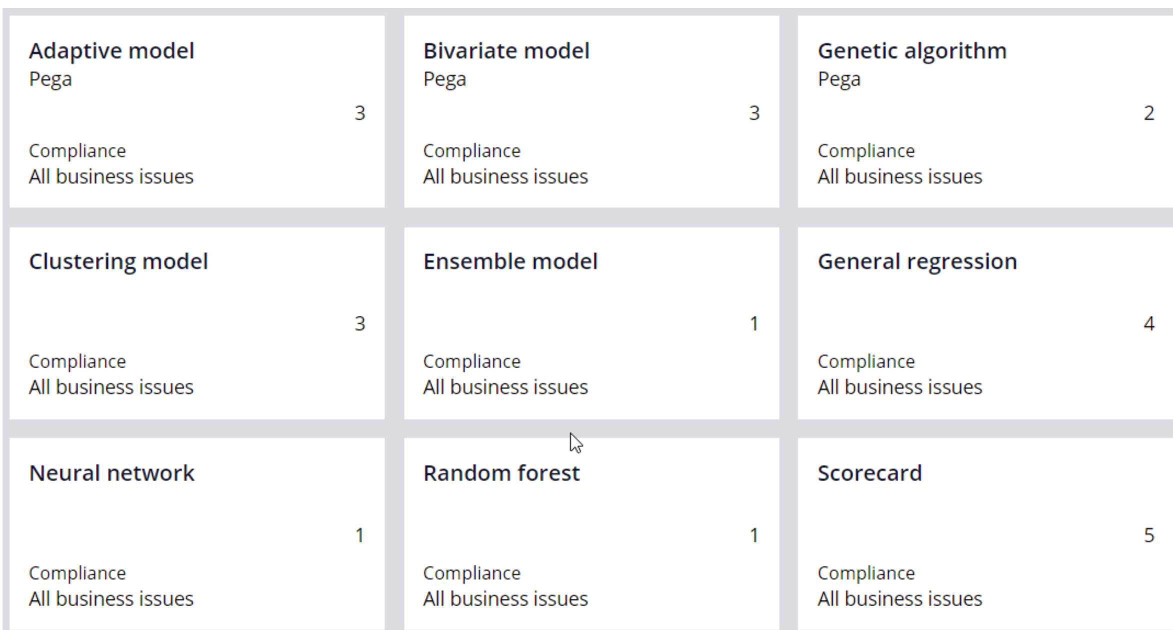
A data set instance can be sourced from a database table, from stream services, or even social media, such as Twitter and YouTube. Resources include taxonomies and the default sentiment lexicon to use in building machine learning models. When enabled, historical data used for the training of adaptive models and monitoring of predictive models is recorded for offline analysis.

In the **Settings** work area, you can manage general Prediction Studio settings and connect to third-party machine learning platforms.



Also, you can review company policies regarding the transparency thresholds for different business issues. In risk management, decisions must be explainable. In marketing, more accurate models may be allowed at the expense of transparency.

Each model type is assigned a transparency score ranging from 1 to 5, where 1 means that the model is opaque, and 5 means that the model is transparent. Depending on the threshold setting, some types of models can be non-compliant for a specific business issue.



This demo has concluded. What did it show you?

- How to create and manage Customer Decision Hub predictions, case management predictions and text analytics prediction.
- How to create and manage the predictive models that drive the predictions.
- How to inspect the model transparency settings of the business.

Customer Decision Hub overview

Description

Familiarize yourself with the 1:1 customer engagement paradigm and discover how Pega's omni-channel AI delivers the right action during every customer interaction.

Prediction Studio is the dedicated workspace for data scientists to control the life cycles of predictions and the predictive models that drive them. Prediction Studio offers prediction and model reports that allow the user to monitor and spot predictions and models that underperform.

Learning objectives

- Explain the basics of the Next-Best-Action approach
- Describe the purpose of Next-Best-Action Designer and the user interface
- Explain the types of predictions that are available in Prediction Studio
- Describe the purpose of the control group
- Describe the bubble chart that visualizes the adaptive model performance
- Recognize the transparency settings for predictive models

Next-Best-Action paradigm

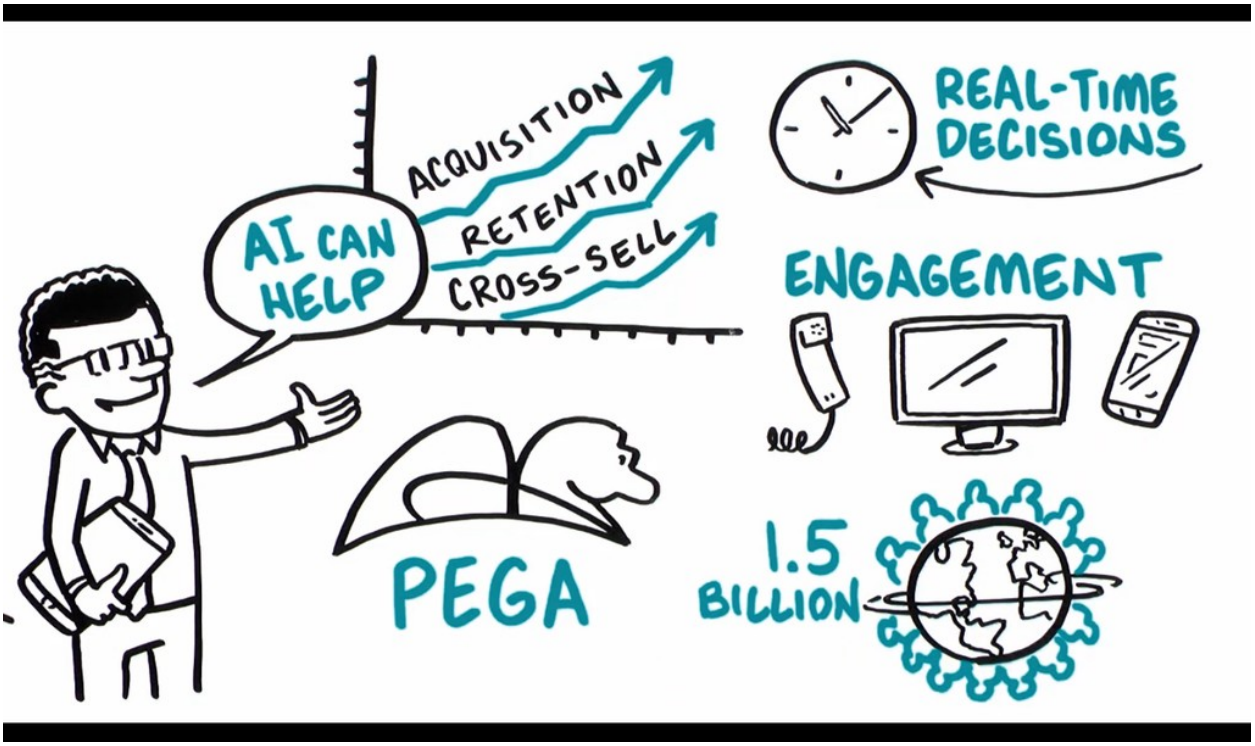
Introduction

The value of big data and analytics is fully realized when every customer conversation delivers exactly the right message, the right offer, or the right level of service to provide a great experience while maximizing the customer's value to the organization. With Pega Next-Best-Action, business experts develop decision strategies that combine predictive and adaptive analytics with traditional business rules to maximize this value.

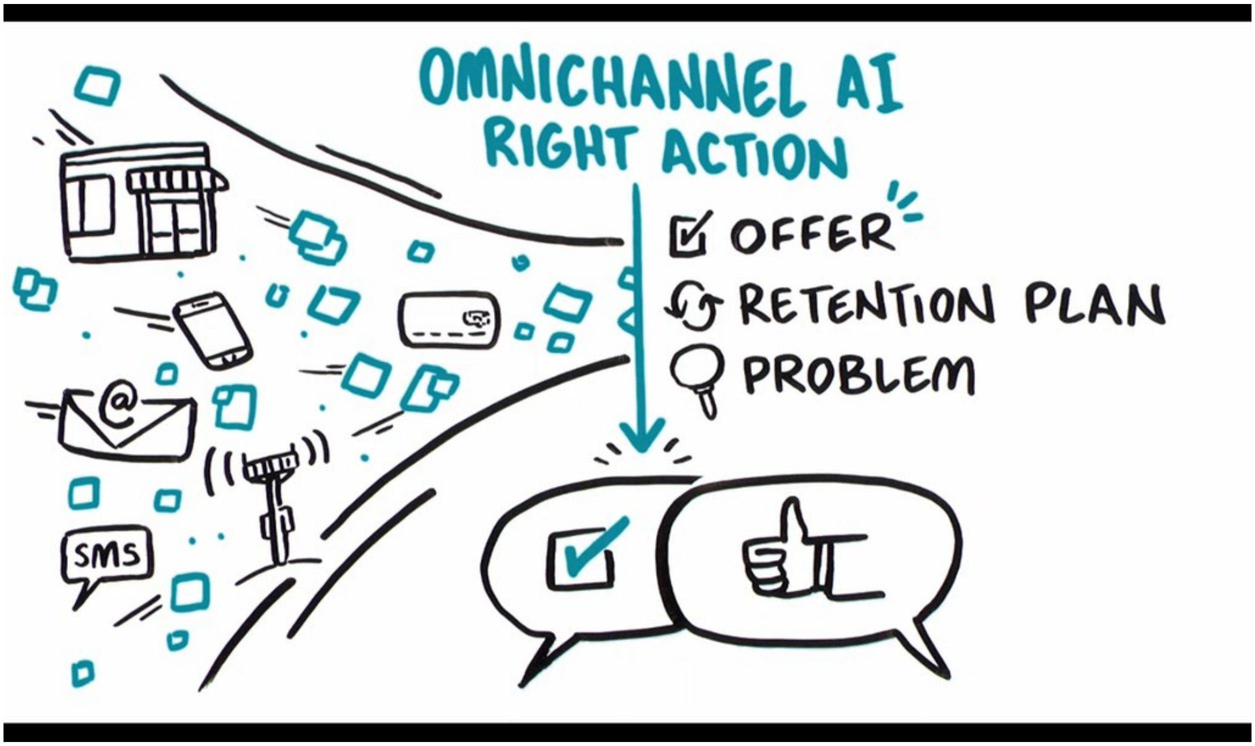
Transcript

This is your customer. You want him to buy your products, use your services and have a great experience. And your competitors want the same thing. To compete, you have to take the right action at every customer touch, ensuring that each conversation delivers exactly the right message, offer and level of service. You want to provide a great experience, while maximizing the customer's value to your organization.

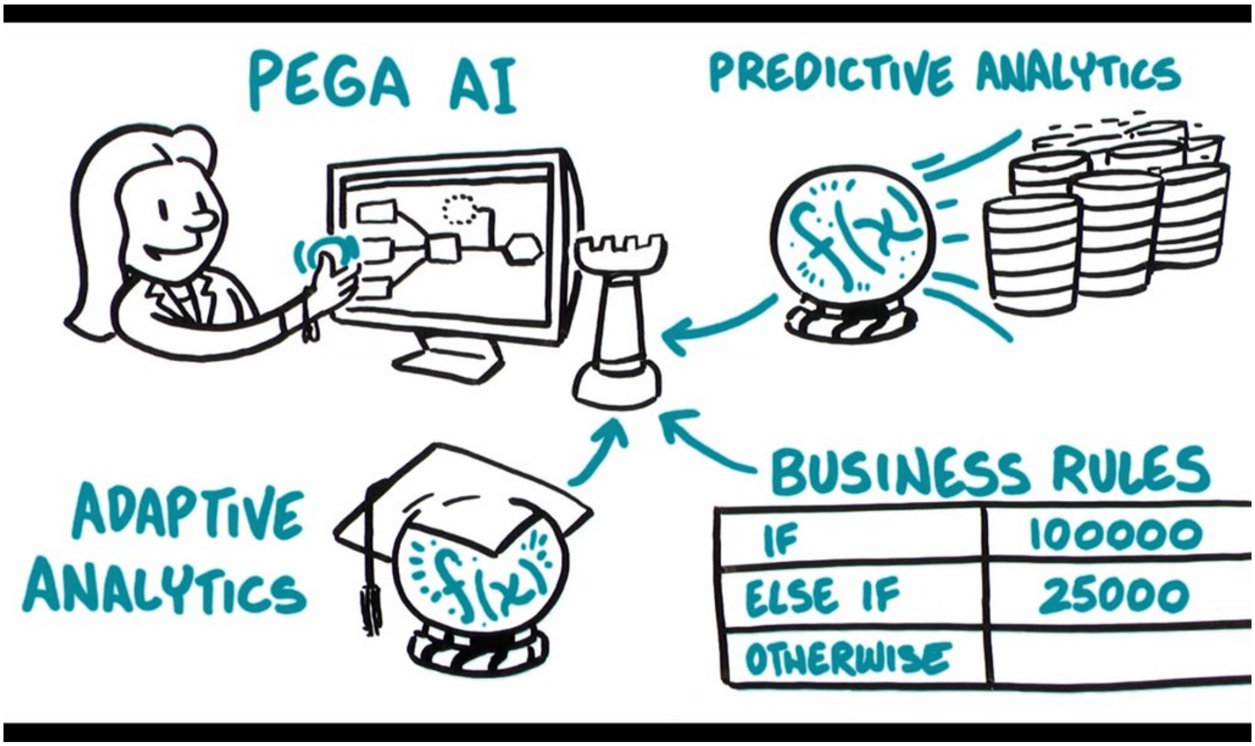
Artificial Intelligence, or AI, can help—if you can get past the hype. Pega has been using AI to create real business value for years, driving real-time decisions that deliver awesome engagement on any channel and improving experiences for over 1.5 billion customers across the globe.



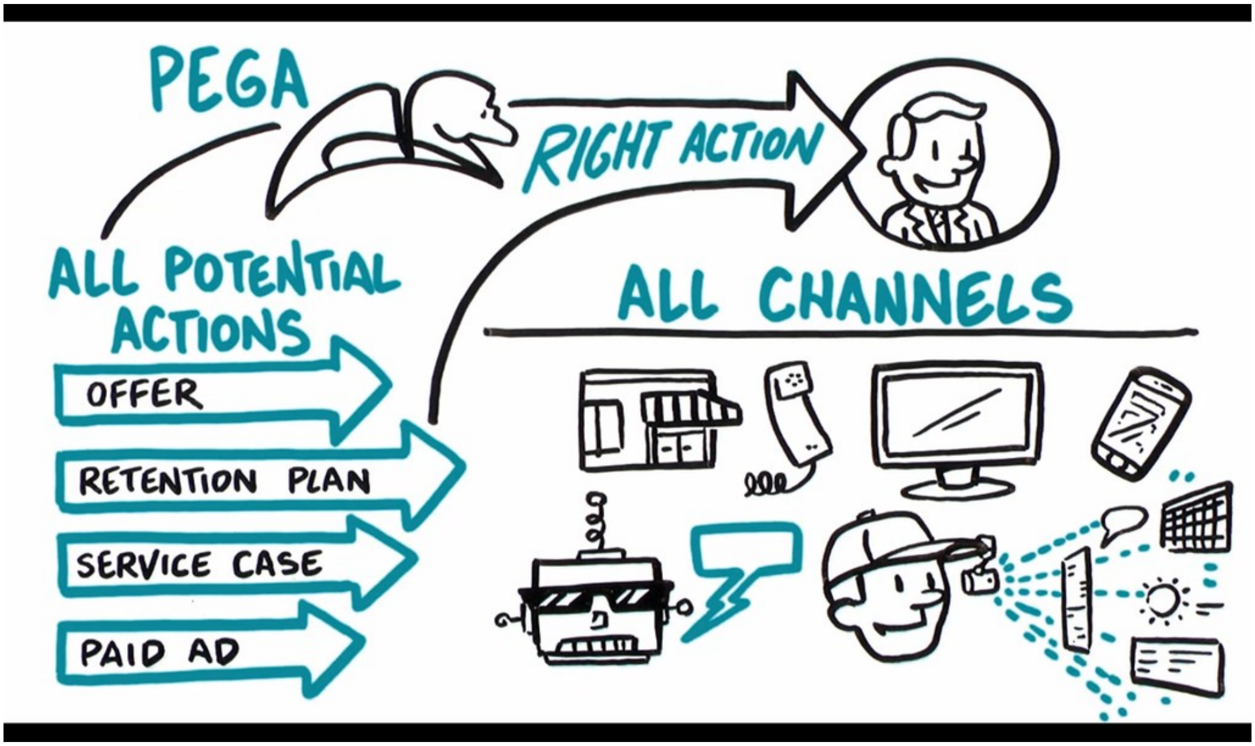
Pega's omni-channel AI delivers the right action at every customer touch by crunching millions of data points in real-time. Make an offer, initiate a retention plan, predict a problem before it happens. Every decision generates the next-best-action for your customer, and your business.



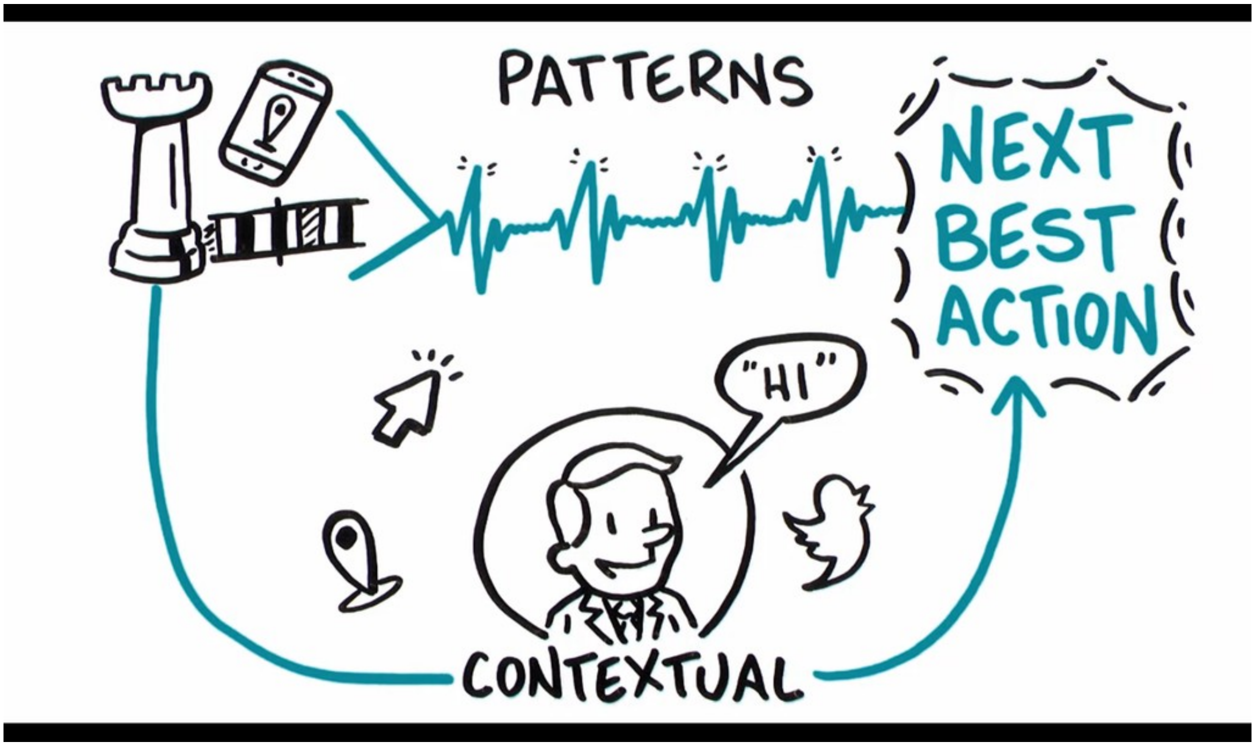
Pega's AI is built for business people, not scientists or developers. They design visual decision strategies that combine predictive analytics, algorithms developed through mining large sets of data, adaptive analytics, machine-learning algorithms that improve with each interaction, and traditional business rules that allow users to prioritize and arbitrate between decisions.



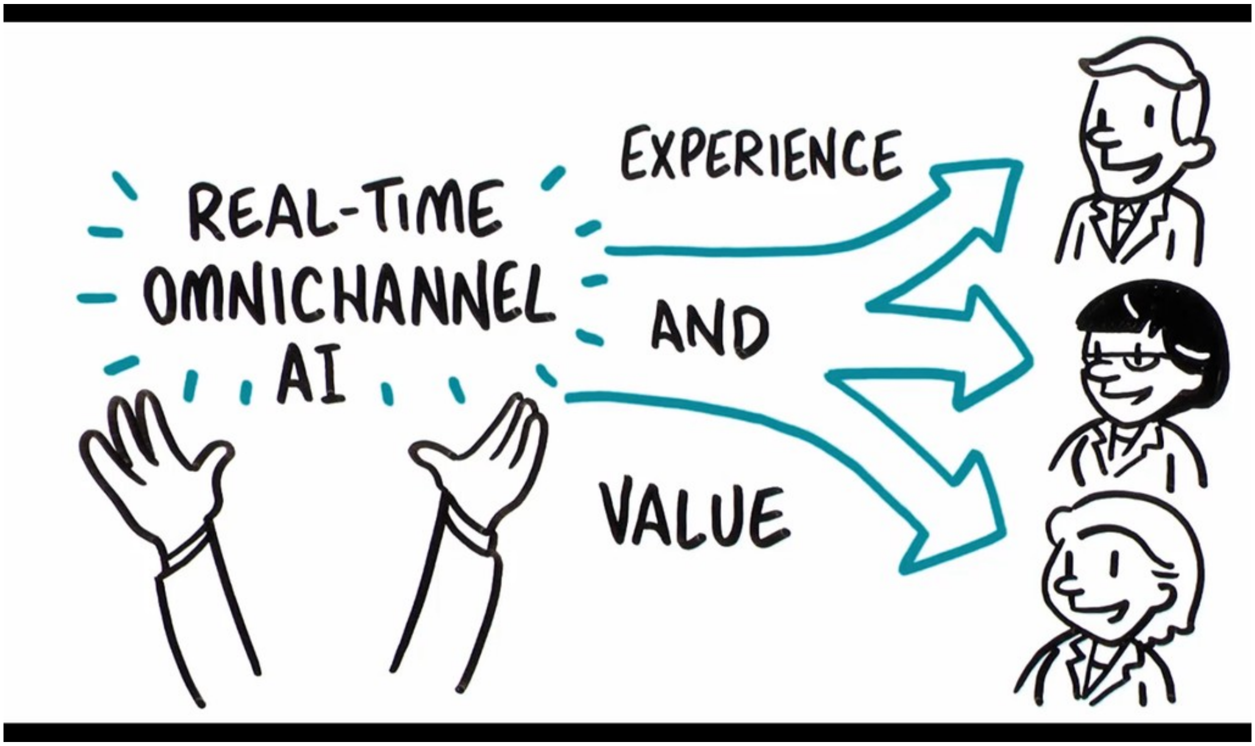
Pega uses the strategy to look across all the potential actions you may take with a customer, make an offer, initiate a retention plan, open a service case, place an ad, and ensure exactly the right action is taken at every interaction and it works across all channels to provide a consistent experience in a store, on the phone, on the web, mobile, with the chat bot, or just some crazy tech that hasn't even been invented yet.



And Pega connects to streams like mobile locations or network events to detect patterns and drive the Next Best Action proactively. And strategies are completely contextual. Any change in the customer's context — a click, a reply, a location change, a Tweet — will trigger the Next Best Action. So, you can really listen to your customers and act accordingly.



Pega's real-time, omni-channel AI puts the power in your hands, so you can optimize every customer interaction for experience, and value.



One-to-one customer engagement paradigm

Introduction

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 have to be able to perform the right action in the right channel at the right moment for each customer. We call this capability, “1-to-1 Customer Engagement”.

Transcript

In this video, learn about the 1-to-1 Customer Engagement paradigm and how the principles of Next-Best-Action are implemented using the Pega Customer Decision Hub™.

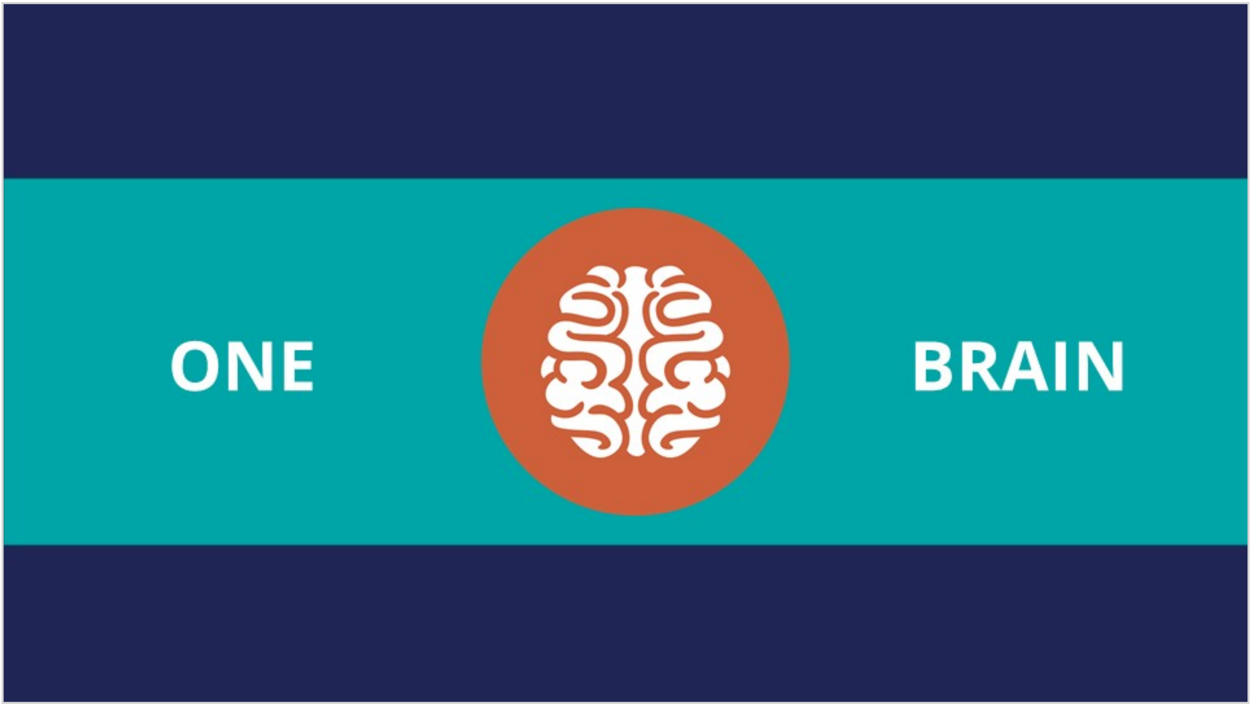
Customers are more empowered than ever before. As a result, they have very high expectations of the experiences they receive from their service providers. Their experiences must make sense within the context of their lives. This means they must be meaningful, consistent, and personalized across every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

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 have to be able to perform the right action in the right channel at the right moment for each customer.

We call this capability, "1-to-1 Customer Engagement".

1-to-1 Customer Engagement enables companies to 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 1-to-1 Customer Engagement is an idea that’s simple to conceive, but very difficult to execute: one centralized brain.

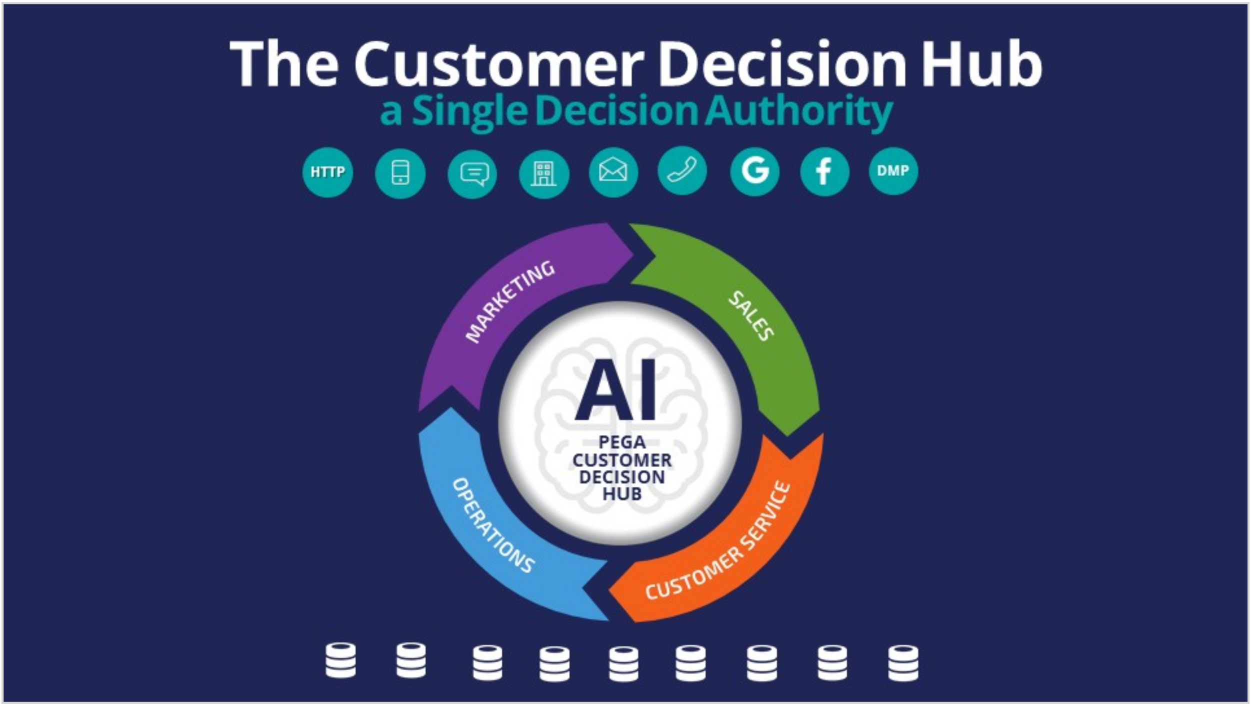


In other words, one piece of intelligence that acts as a single decision authority across your application ecosystem.

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

In Pega Customer Decision Hub, this centralized brain is the core capability that leverages AI to enable 1-to-1 Customer Engagement.

In Pega Infinity™, the 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.



Data from every customer engagement across the enterprise is collected by the Brain and used to make predictions and decisions about every interaction in every channel.

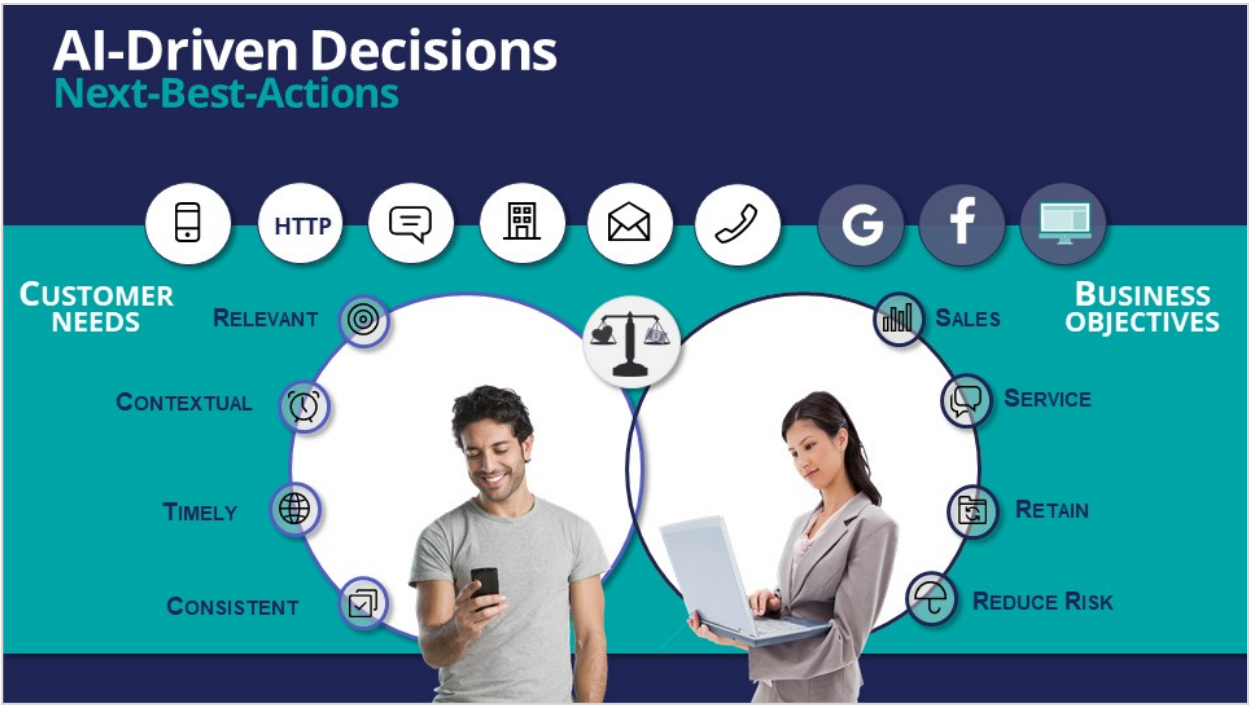
Continuous learning and decision-making are the foundation of a 1-to-1 Customer Engagement solution.

The 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-Action.

Every Next-Best-Action weighs customer needs against business objectives to optimize decisions based on priorities set by the business manager.

In the milliseconds before interacting with a customer, the Customer Decision Hub processes thousands of predictive and adaptive models to determine customer needs, considering the customer's immediate context to ensure the Next-Best-Action is relevant, timely, and contextual. These models can be propensity, risk, or churn models.

Next, the decision strategy considers business rules and matches those with the customer's context and higher-level business goals.



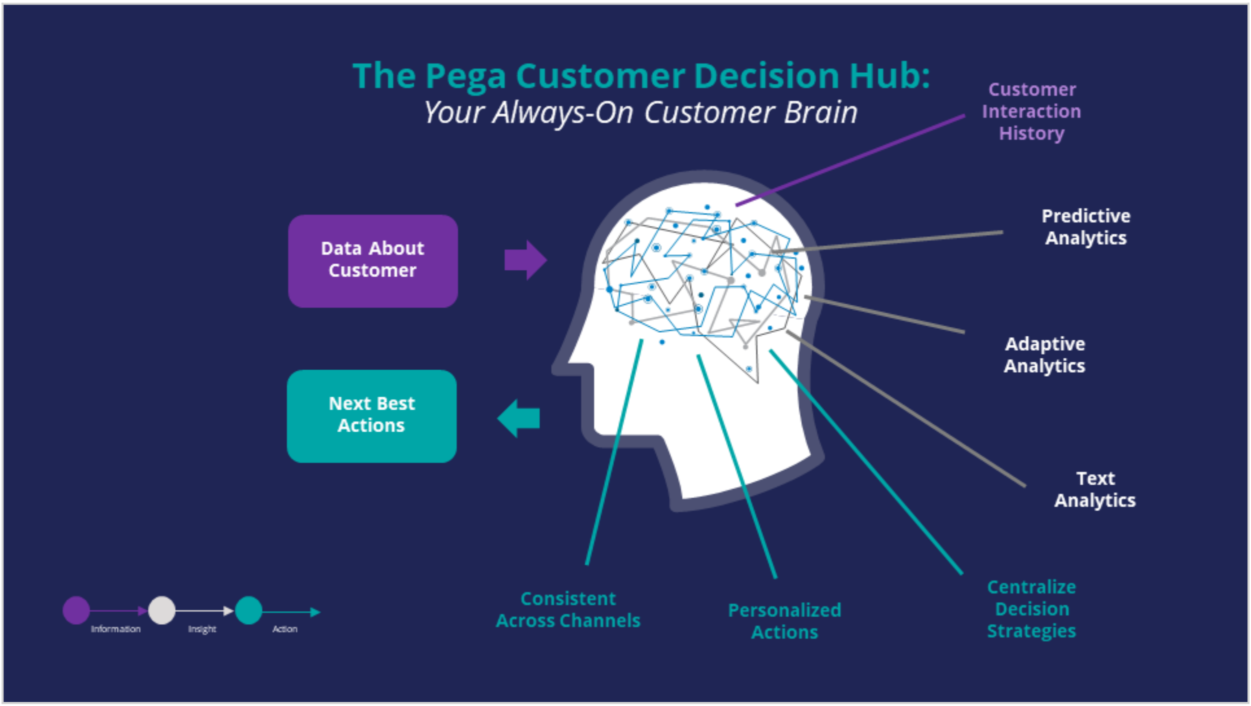
All of this information is used by the Next-Best-Action decision strategy to evaluate every potential action that could be taken with a particular customer in a given situation. The decision strategy then recommends the best way to interact with the customer to achieve the optimal result.

Using the Next-Best-Action approach, the Customer Decision Hub is able to identify the best moments for making a sale, providing a service, making a retention offer, or doing nothing at all (e.g. if nothing is relevant enough to warrant the customer's attention). Next-Best-Action is even able to select which offers are most likely to be accepted by the customer in a sales or retention situation. Next-Best-Action decisions are distributed, in real-time, to each of your real-time owned channels, such as web, mobile, and contact center. Through Pega Customer Decision Hub, Next-Best-Actions can also be distributed 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.

Once the Next-Best-Actions are distributed and customer responses have been received by the Brain, the whole process begins again, and new Next-Best-

Actions are distributed within milliseconds. Every outbound channel, including a data management platform, is dynamically updated with the Next-Best-Action to ensure consistency and an optimized customer experience no matter which channel the customer interacts with.

In summary, the Pega 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 leverages advanced AI techniques to make predictions.

And it uses decision strategies (which combine traditional business rules with predictive, adaptive and text analytics), to deliver consistent and personalized Next-Best-Actions across all channels.

Action arbitration

Introduction

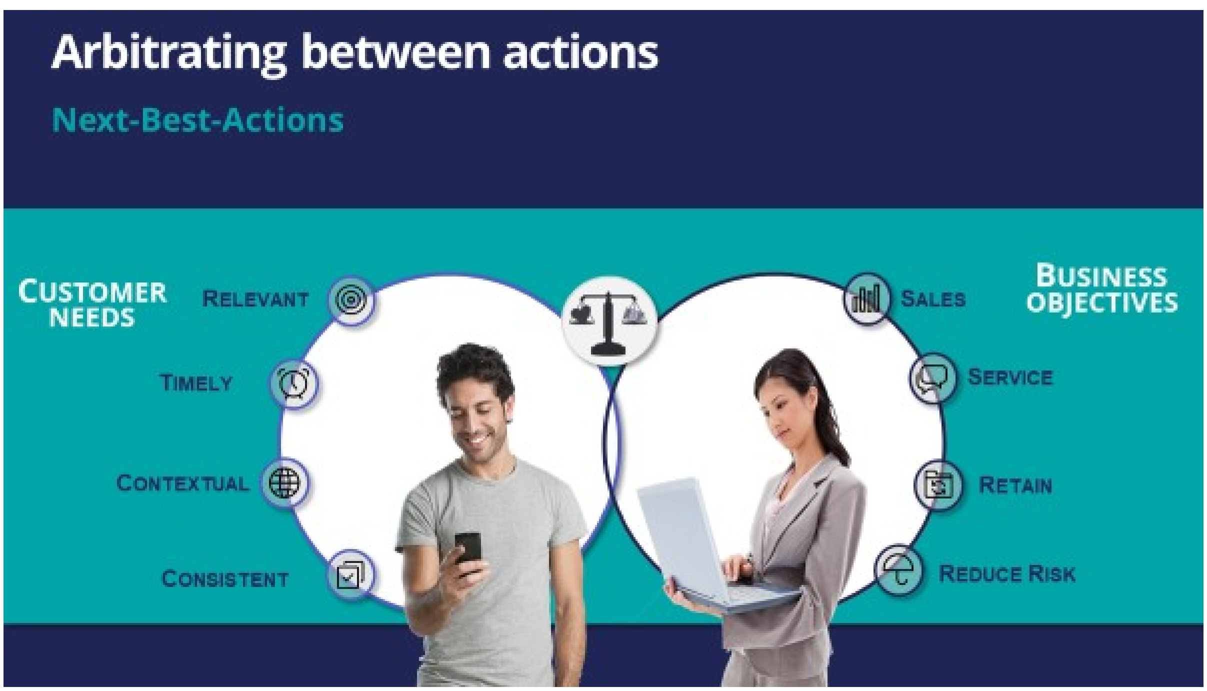
Pega Customer Decision Hub combines analytics, business rules, customer data, and data collected during each customer interaction to create a set of actionable insights that it uses to make intelligent decisions. Arbitration aims to balance customer relevance with business priorities by weighing numerical values for the following factors: propensity, context weighting, action value, and business levers. Learn to create a simple formula for arriving at a prioritization value, which is used to select the top actions.

Transcript

This video explains the concept of action arbitration.

Pega 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 Next-Best-Action.

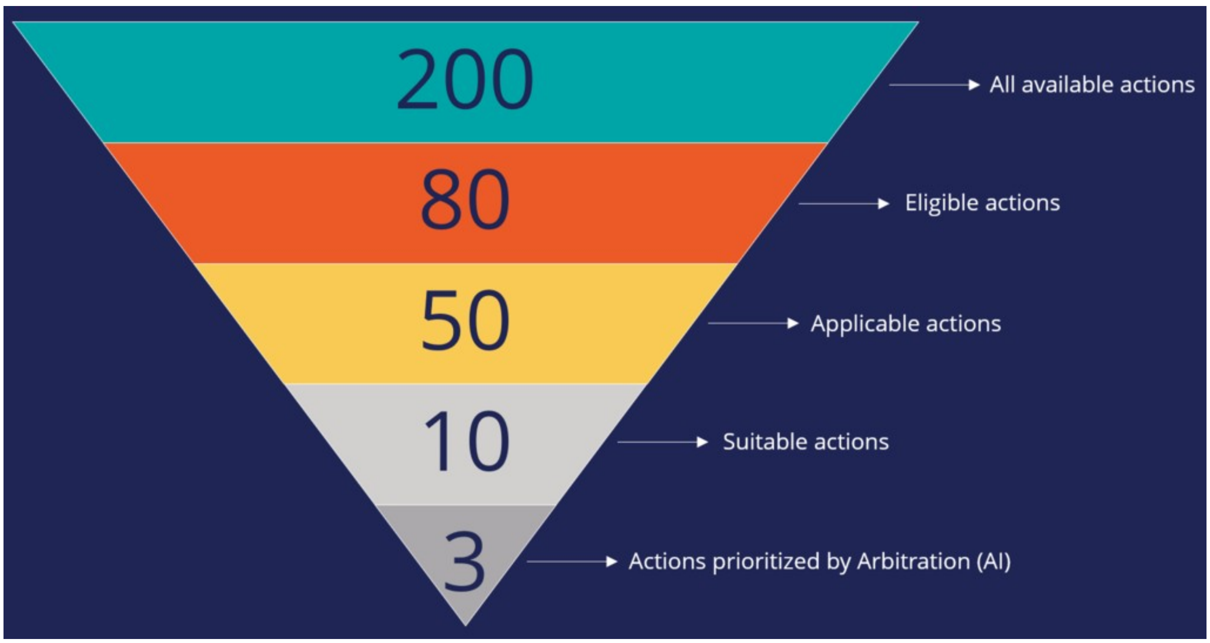
Every Next-Best-Action weighs customer needs against business objectives to optimize decisions based on priorities set by the business manager.



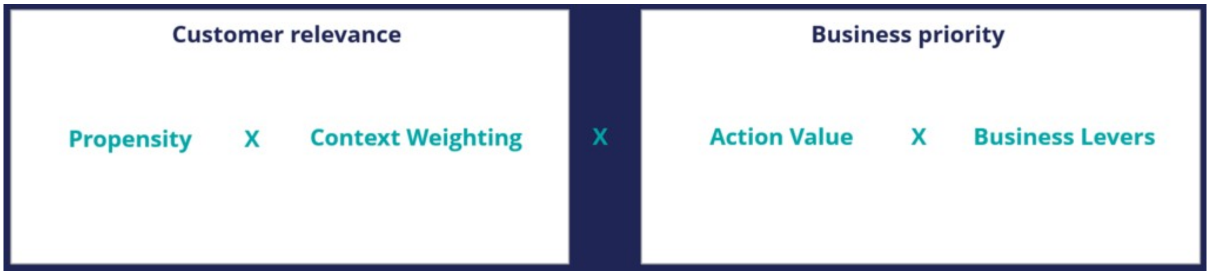
U+ Bank, a retail bank, has several actions for its customers and has configured engagement policies to suit both customer needs and business objectives.

In this scenario, a marketer for U+ has designed 200 actions that can be presented to customers. To select the Next-Best-Actions from these, Pega Customer Decision Hub first checks the eligibility conditions and filters the actions. Then, the applicability conditions are run to filter it further. Next, Customer Decision Hub checks the suitability conditions to derive the final set of available actions.

These actions move through one final stage before being presented to customers: the arbitration stage. Arbitration is used to prioritize and choose the best actions based on what is relevant for the customer right now.



Arbitration aims to balance customer relevance with business priorities. The factors weighed are **Propensity**, **Context Weighting**, **Action Value**, and **Business Levers**, each represented by numerical values. A simple formula is used to arrive at a prioritization value, which is used to select the top actions. The number of top actions selected depends on the channel of interaction. For example, the top three actions, plus two tiles and one hero treatment, can be selected for display on a bank's website.



Propensity is the likelihood of a customer responding positively to an action; this is calculated by AI. For example, the higher the likelihood of a customer accepting an offer, the higher the Propensity value for that offer.

Context Weighting allows Pega Customer Decision Hub to consider the situational context for each action. For example, if a customer contacts the bank to close their account, the highest-priority action is to ensure that the customer is retained. The priority of an action is increased by a specified value when the context is detected.

Action Value enables you to assign a financial value to an action and prioritize high-value actions over low-value ones. This value is typically normalized across Issues and Groups. For example, an unlimited data plan is more profitable than a limited data plan. So, in a situation where a customer is eligible for both plans, the unlimited plan has higher priority.

Business Levers allow the business to assert some level of control over the prioritization of actions defined within the system. Levers are used to manually nudge Customer Decision Hub toward Next-Best-Actions based on external factors. For example, the recommended Next- Best-Action might be to offer a credit card to a customer when they visit the home page. But to meet a business goal, the Mortgage Line of Business favors a mortgage offer even if that offer is ranked a little lower on the list of possible actions.

Consider an example where three actions are selected for arbitration. At the moment, only the Propensity is used for prioritization.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
	Sales	Credit cards	Gold Card	0.5	1	1	1	0.5
	Retention	Proactive	10% discount	0.55	1	1	1	0.55
	Service	Administrative	Address change	0.4	1	1	1	0.4

Action arbitration with propensity before prioritization

The result of the arbitration is that the top action is the one with the highest Propensity.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Retention	Proactive	10% discount	0.55	1	1	1	0.55
2	Sales	Credit cards	Gold Card	0.5	1	1	1	0.5
3	Service	Administrative	Address change	0.4	1	1	1	0.4

Action arbitration with propensity after prioritization

Examine what happens when Context Weighting together with Propensity are considered for arbitration. For example, if the intent of a customer calling customer service is to change their address, the Context Weight of a Service action increases.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Retention	Proactive	10% discount	0.55	1	1	1	
2	Sales	Credit cards	Gold Card	0.5	1	1	1	
3	Service	Administrative	Address change	0.4	2	1	1	

Action arbitration with context weight before prioritization

As a result, the Arbitration caters to the current need of the customer and presents a Service action as the top action for the customer. Thus, the Arbitration caters to the current need of the customer and presents a Service action as the top action for the customer.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Service	Administrative	Address change	0.4	2	1	1	0.8
2	Retention	Proactive	10% discount	0.55	1	1	1	0.55
3	Sales	Credit cards	Gold Card	0.5	1	1	1	0.5

Action arbitration with context weight after prioritization

Consider another scenario in which a customer is eligible for two credit cards and two other actions. Now, consider that the Action Value is also used in arbitration when prioritizing. In this case, the Platinum Card is assigned a higher value by the business than the Gold Card.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Sales	Credit cards	Gold Card	0.6	1	1	1	
2	Sales	Credit cards	Platinum Card	0.55	1	2	1	
3	Retention	Proactive	10% discount	0.2	1	1	1	
4	Service	Administrative	Address change	0.1	1	1	1	

Action arbitration with action value before prioritization

Thus, the arbitration selects the Platinum Card as the top action.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Sales	Credit cards	Platinum Card	0.55	1	2	1	1.1
2	Sales	Credit cards	Gold Card	0.6	1	1	1	0.6
3	Retention	Proactive	10% discount	0.2	1	1	1	0.2
4	Service	Administrative	Address change	0.1	1	1	1	0.1

Action arbitration with action value after prioritization

Finally, consider an example in which all four parameters are used for arbitration. In this case, U+ Bank wants to promote two new checking account offers under the Sales issue. The bank sets a higher Business Lever value for the Checking Accounts actions.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Sales	Credit cards	Gold Card	0.6	1	1	1	
2	Sales	Credit cards	Platinum Card	0.55	1	1	1	
3	Sales	Checking Accounts	Premium Checking	0.55	1	1	2	
4	Sales	Checking Accounts	Student Checking	0.5	1	1	2	
5	Retention	Proactive	10% discount	0.2	1	1	1	
6	Service	Administrative	Address change	0.1	1	1	1	

Action arbitration with business levers before prioritization

Although the Propensity of the Checking Accounts actions is low, they are selected as the top actions due to their high Lever values.

Rank	Issues	Groups	Actions	Propensity	Context weighting	Action value	Business levers	Priority
1	Sales	Checking Accounts	Premium Checking	0.55	1	1	2	1.1
2	Sales	Checking Accounts	Student Checking	0.5	1	1	2	1
3	Sales	Credit cards	Gold Card	0.6	1	1	1	0.6
4	Sales	Credit cards	Platinum Card	0.55	1	1	1	0.55
5	Retention	Proactive	10% discount	0.2	1	1	1	0.2
6	Service	Administrative	Address change	0.1	1	1	1	0.1

Action arbitration with business levers after prioritization

Next-Best-Action Designer

Introduction

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. Next-Best-Action Designer is organized according to the high-level sequence of steps needed to configure the Next-Best-Action strategy for your organization.

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.

The tabs across the top of the user interface represent the steps that need to be completed to define Next-Best-Actions.

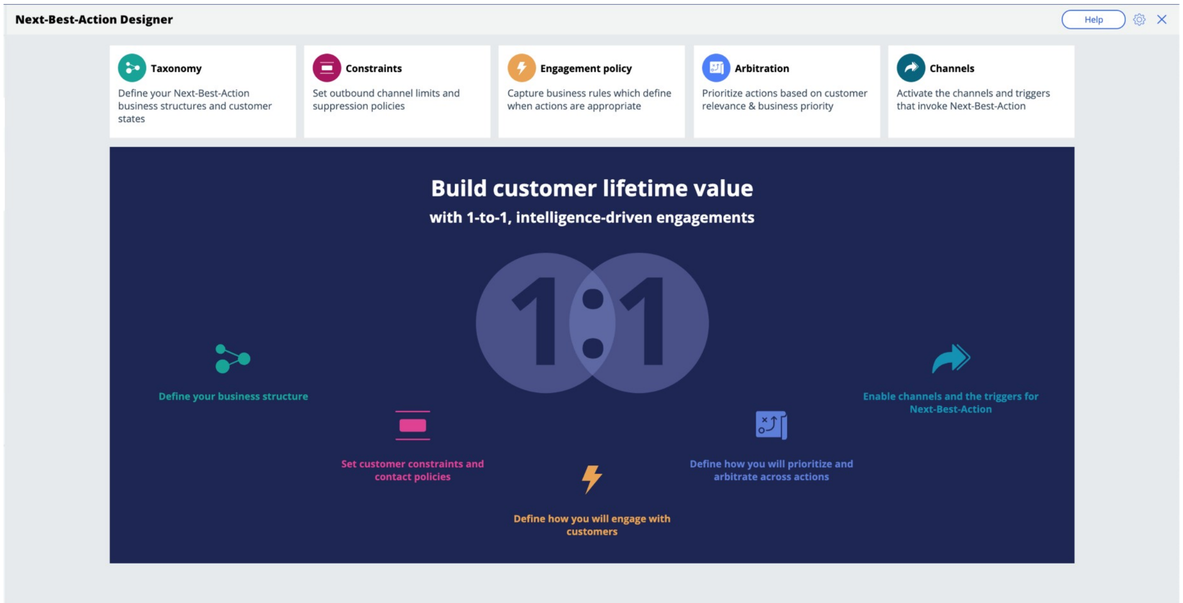
Use the **Taxonomy** component to define the business structure for your organization.

Use the **Constraints** component to implement channel limits and constraints.

Use the **Engagement policy** component to define the rules that control which actions are offered to which customers.

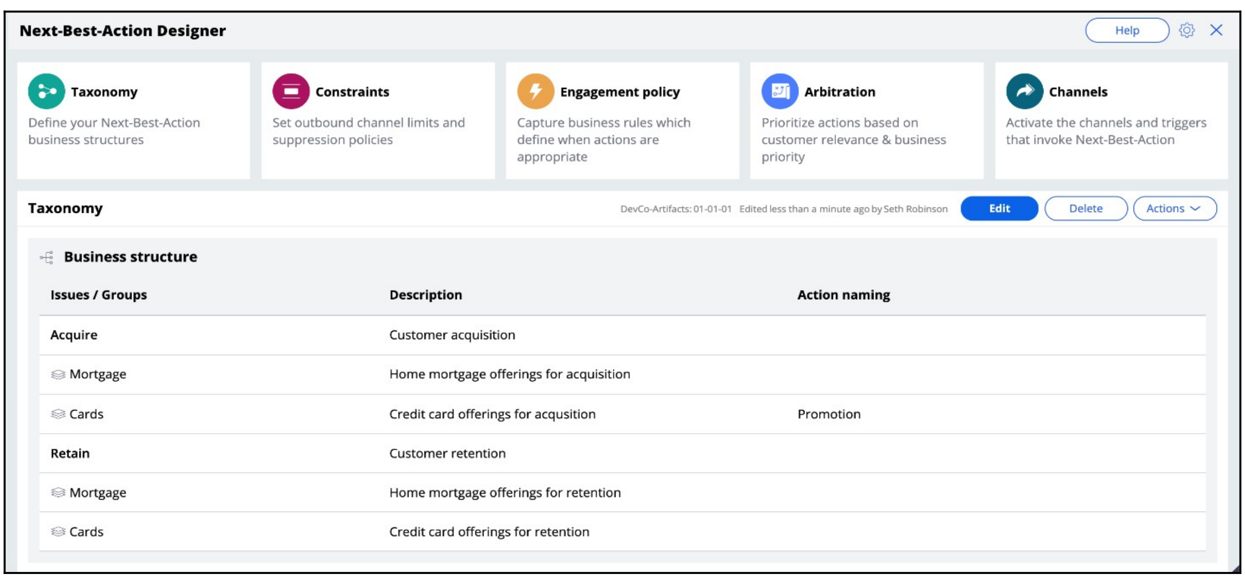
Use the **Arbitration** component to configure how actions are prioritized.

Use the **Channels** component to configure when and where Next-Best-Action is triggered.



The system uses these definitions to create an underlying 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.

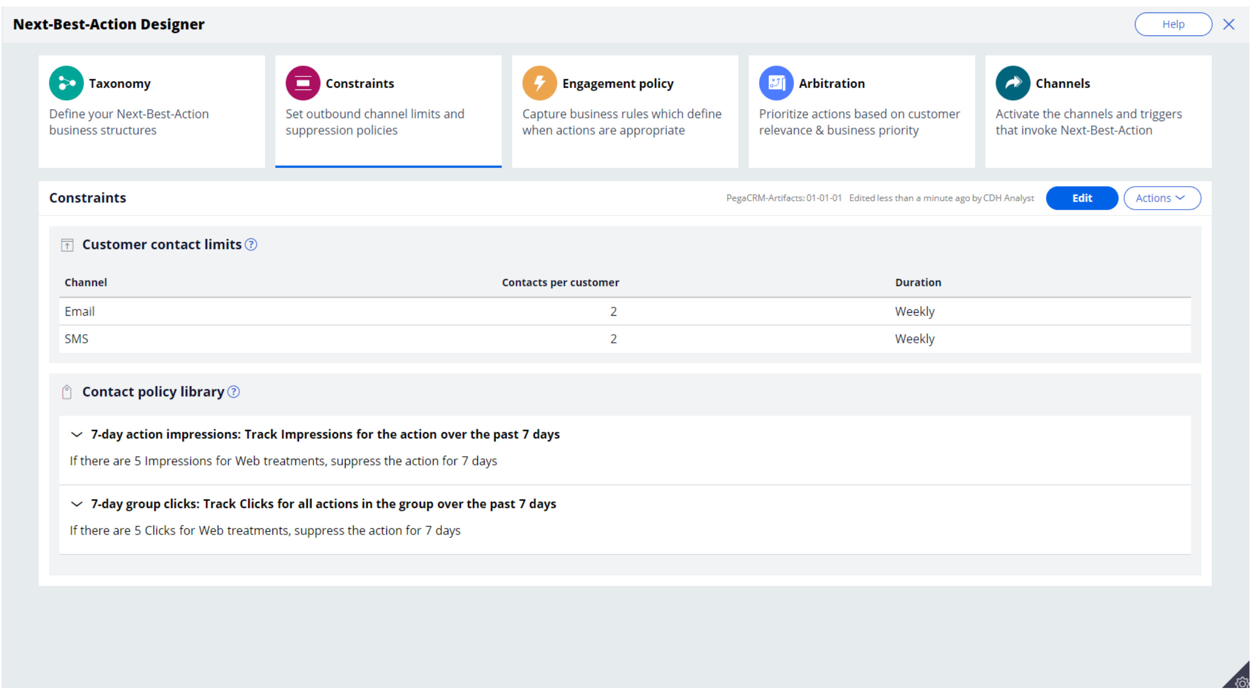
Use the **Taxonomy** component to define the hierarchy of Business Issues and Groups to which an action belongs.



A Business Issue is the purpose behind the actions you offer to customers. For example, actions with the purpose of retaining existing customers should be grouped under the business Issue of Retention. Actions with the purpose of acquiring new customers belong to the business Issue of Acquisition.

Business Groups are used to organize customer actions into categories. For example, as part of the business Issue of Acquisition, you can create Groups for products like Credit Cards, Mortgages, or Personal Loans, with the intention of offering these to potential customers.

Use **Constraints** to specify outbound contact limits as well as to limit overexposure to a specific action or group of actions.

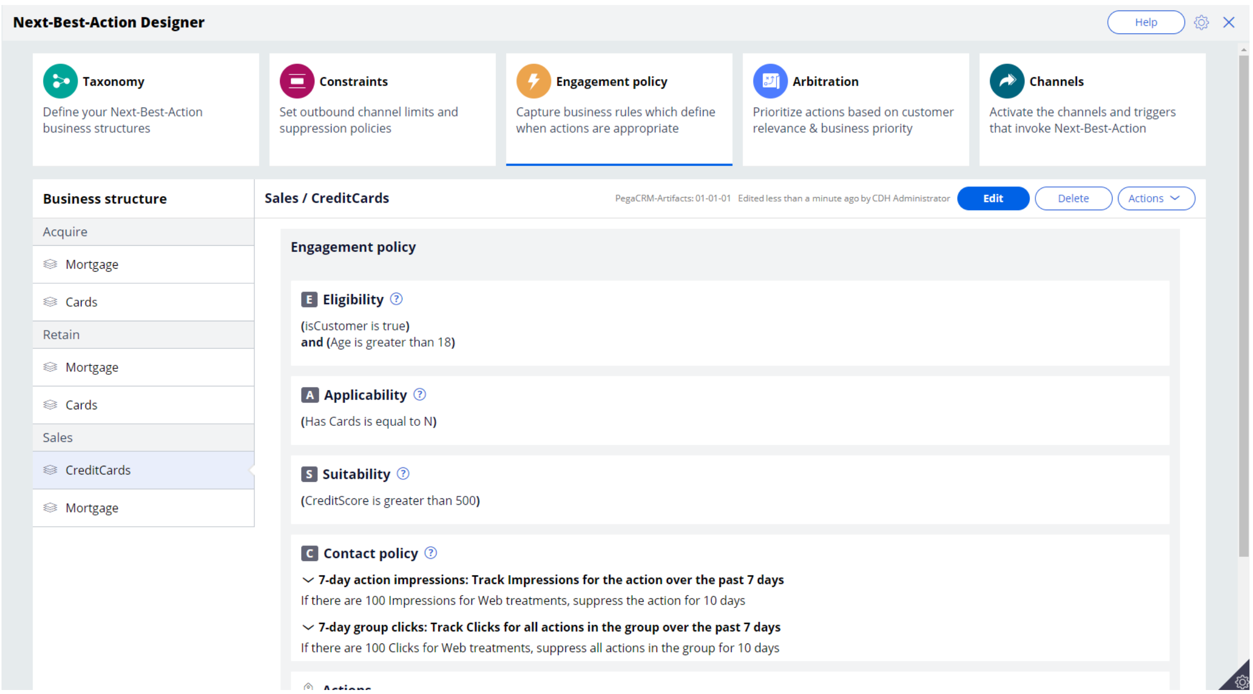


Customer contact limits allow you to limit the number of interactions that a customer can receive over a given period of time on a specific channel. For example, you can decide that you do not want your customers to receive more than two emails per week or two SMS messages per week.

On the Constraints tab of Next-Best-Action Designer, 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.

In the Contact Policy library, you define suppression rules that automatically put an action on hold after a specific number of outcomes are recorded for some or all channels. For example, an action can be suppressed for a customer for seven days after the customer has seen an ad for that action five times. Suppressing or pausing an action prevents over-exposure by limiting the number of times a customer is exposed to the same action.

Use **Engagement policies** to define when specific actions or groups of actions are appropriate for customers.



There are four types of engagement policies:

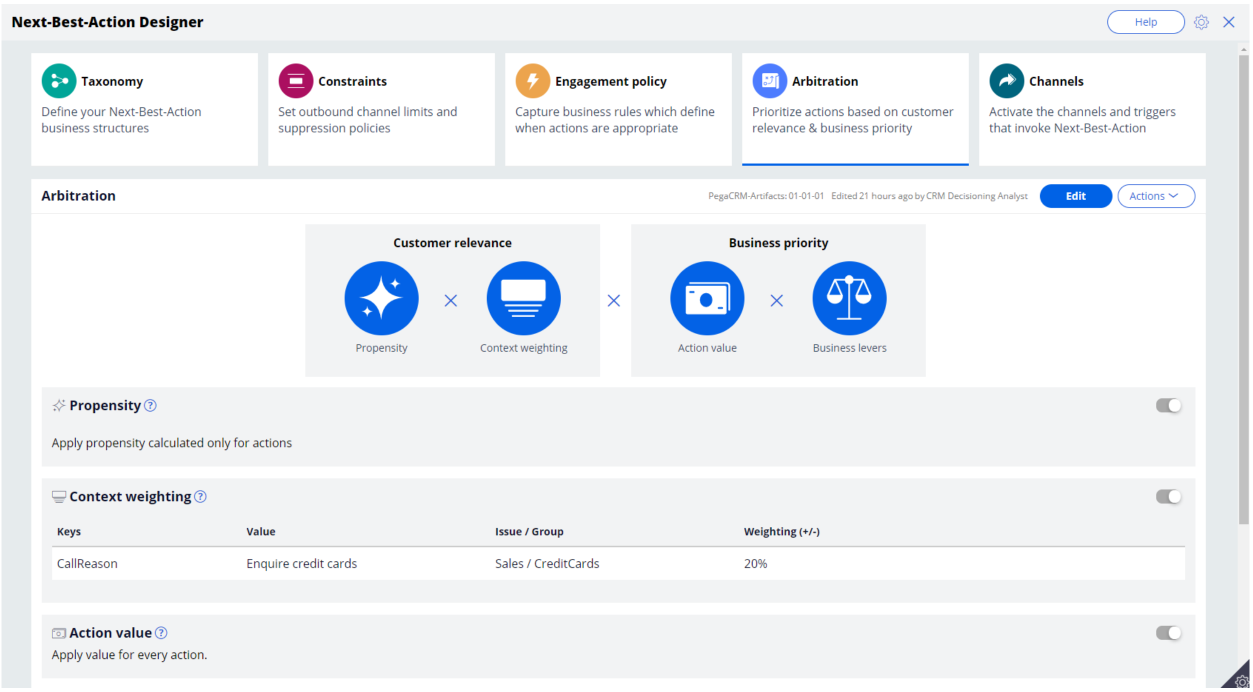
Eligibility determines whether or not a customer qualifies for an action or group of actions. For example, an action may only be available for customers over a specific age or living in a specific geographic location.

Applicability determines if 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 may not be relevant for a customer who already owns a card.

Suitability determines if an action or group of actions is appropriate for a customer for ethical or empathetic reasons. For example, a new loan offer may not be appropriate for a customer whose credit score is low, even though it might be profitable for the bank.

Contact Policies determine when an action or group of actions should be suppressed and for how long. For example, you can suppress an action after a specific number of promotional messages has been sent to customers. To activate Contact Policy rules created in the library on the Constraints tab, add them to the Engagement Policy tab.

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



The factors weighed in arbitration are: Propensity, Context weighting, Action value, and Business levers, each represented by numerical values. A simple formula is used to arrive at a prioritization value, which is used to select the top actions.

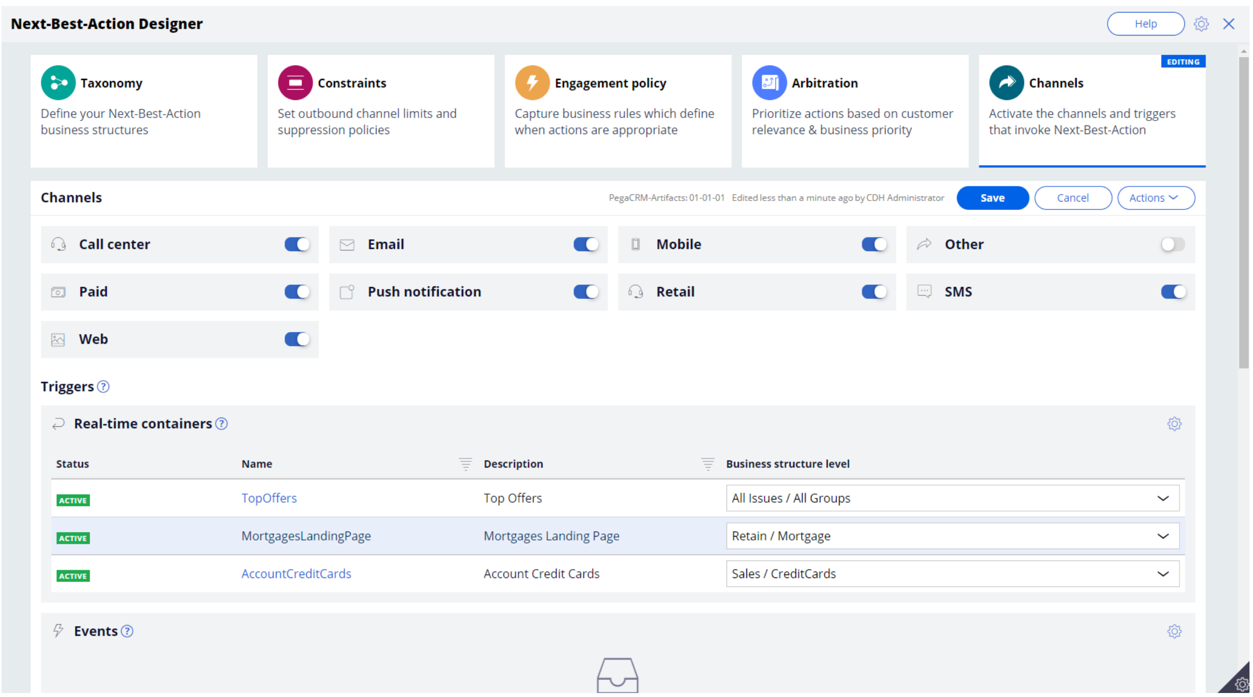
Propensity is the likelihood of a customer responding positively to an action. This is calculated by Artificial Intelligence (AI). For example, a click on an offer banner or an accept of an offer in the contact center are considered positive behaviors.

Real-time contextual data is an important part of making highly relevant recommendations. **Context weighting** allows you to assign weighting to a specific context value for all actions within an Issue or Group. For example, if a customer contacts the bank to change their address, the weight of the Service context will increase, and the highest priority action will be to ensure that the relevant service is delivered to the customer.

Action 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. Action values are typically normalized across Issues and Groups.

Business levers enable you to accommodate ad hoc business priorities by specifying a weight for an action or Group of actions and/or its associated Business Issue.

Next-Best-Action Designer enables Next-Best-Actions to be delivered via inbound, outbound and paid channels.



These channels can be toggled on or off. If a channel is toggled off, the Next-Best-Actions will not be delivered to that channel.

An external real-time channel is any channel that presents actions selected by the Customer Decision Hub to a customer. These channels can include a website, or a call-center or mobile application. A real-time container is a placeholder for content in an external real-time channel.

A trigger is a mechanism whereby an external channel invokes the execution of a Next-Best-Action decisioning process for specific Issues and Groups. The result will be delivered back to the invoking channel. For example, when a real-time container called “Mortgages Landing Page” is configured, the website invokes this real-time container before loading the mortgage page.

As you have seen in this video, Next-Best-Action Designer is organized according to the high-level sequence of steps needed to configure the Next-Best-Action strategy for your organization. These steps involve:

- Defining the business structure for your organization
- Implementing the channel limits and constraints
- Defining the rules that control which actions are offered to which customers
- Configuring how actions are prioritized
- Configuring when and where Next-Best-Action is triggered

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

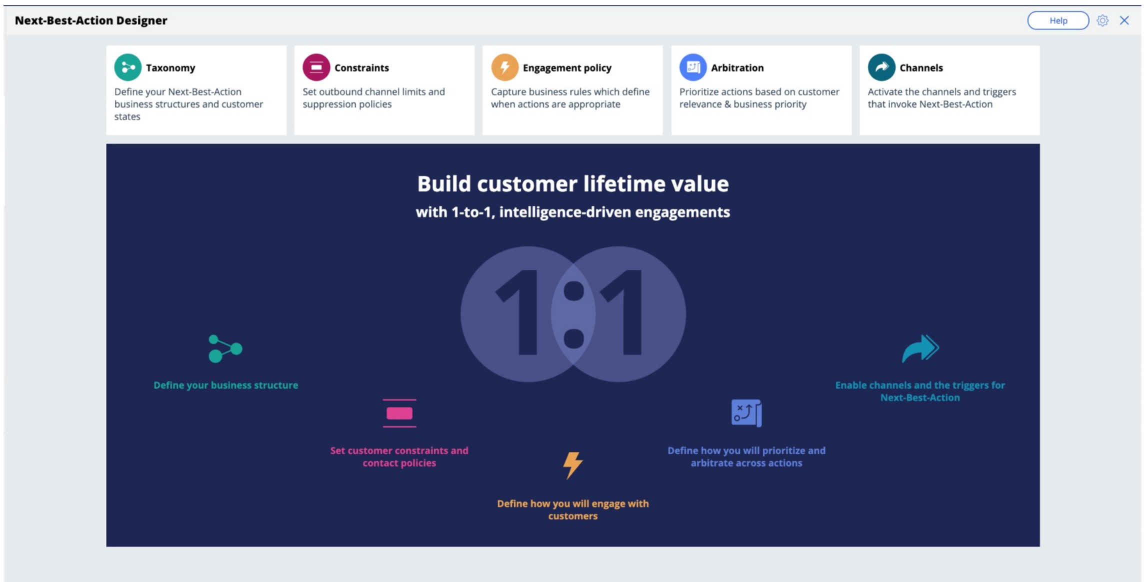
- Describe how decision strategies are used in the Next-Best-Action strategy framework
- Explain the decision strategy canvas and its building blocks
- Create decision strategies from scratch
- Explain what’s going on inside each component when a decision strategy is executed

Decision strategies

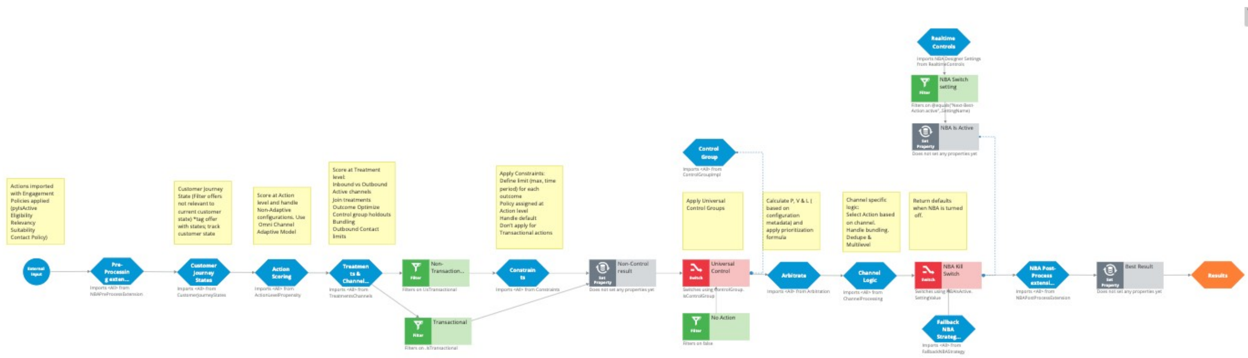
Transcript

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.



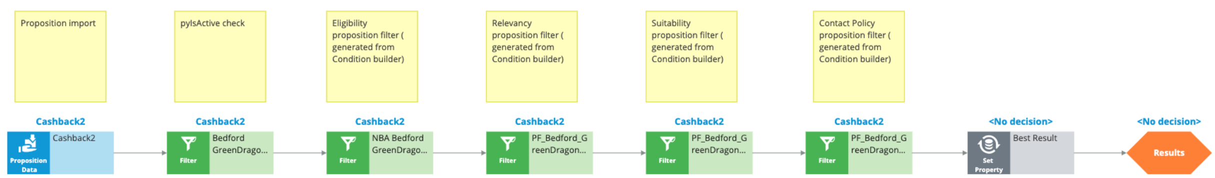
As you use the Next-Best-Action Designer user interface to define strategy criteria, the system uses these criteria to create the Next-Best-Action Strategy framework. This framework leverages best practices to generate Next-Best-Action decision strategies at the enterprise level. These decision strategies 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.



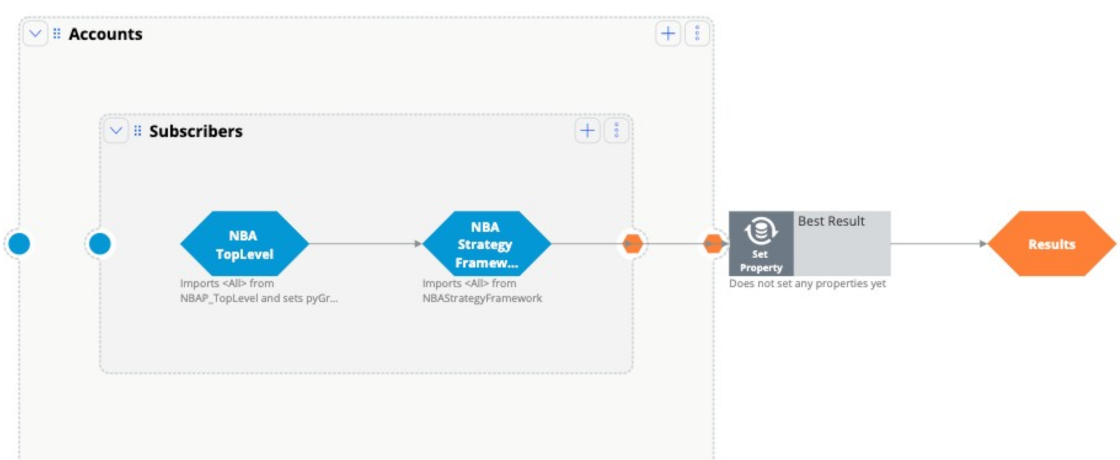
If you want to modify the strategy later, you can do that from Next-Best-Action Designer's simple and transparent interface.

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, Relevance and Suitability rules.

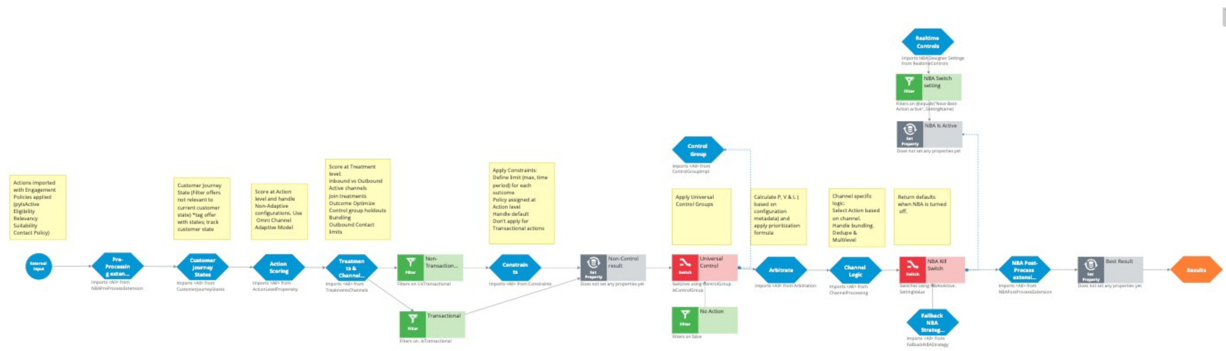


The strategy then passes these results to the strategy framework for processing.

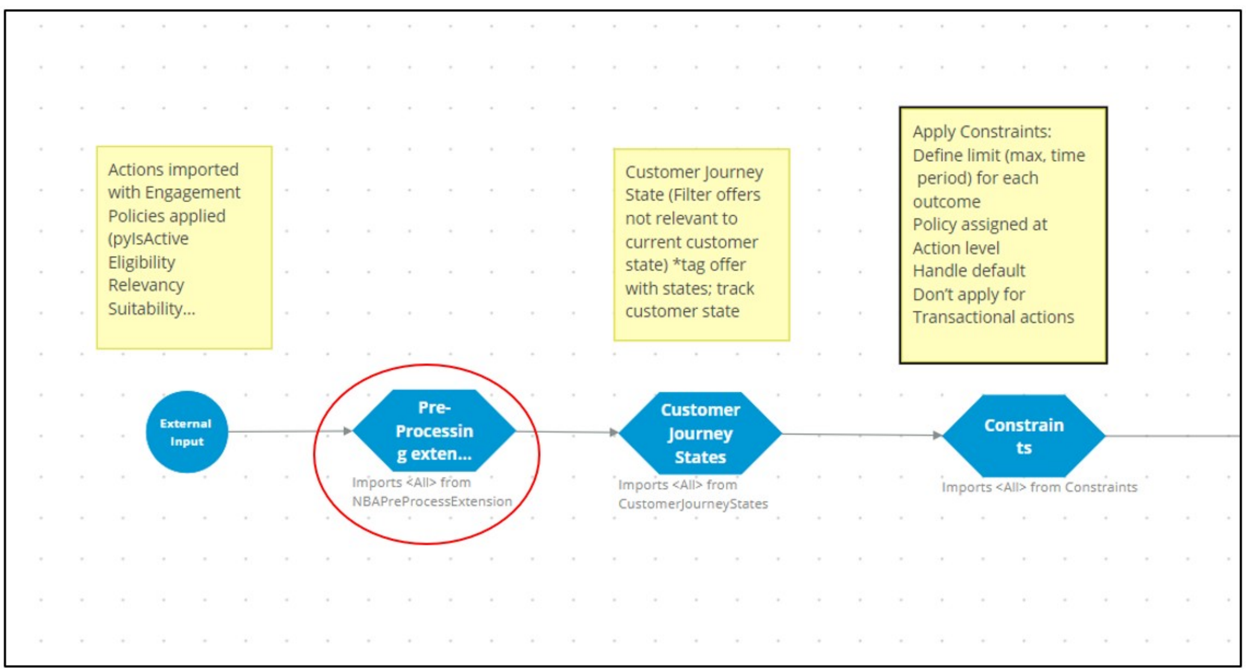


There are several extension points within the framework. An extension point is an empty rule or activity that is intended to be overridden to meet the specific needs of the application. When building an implementation of the current framework, the decision strategy designers must override the empty activity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

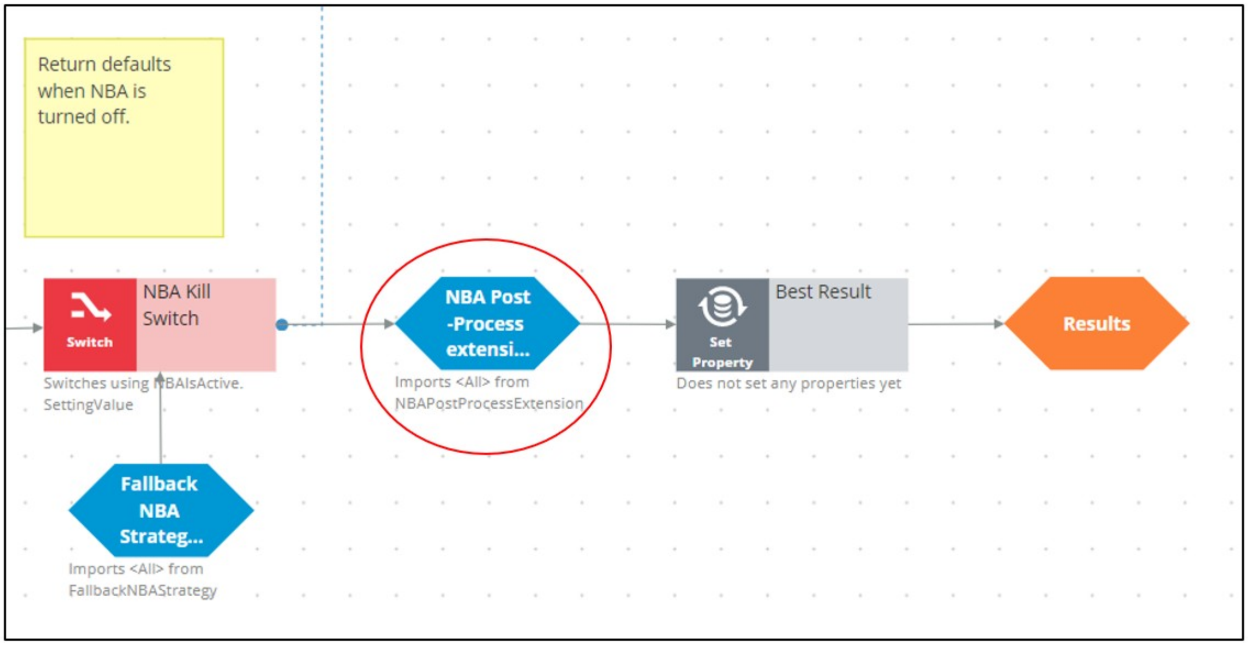
This is the NBA framework strategy when applied to each of the Actions.



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 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.

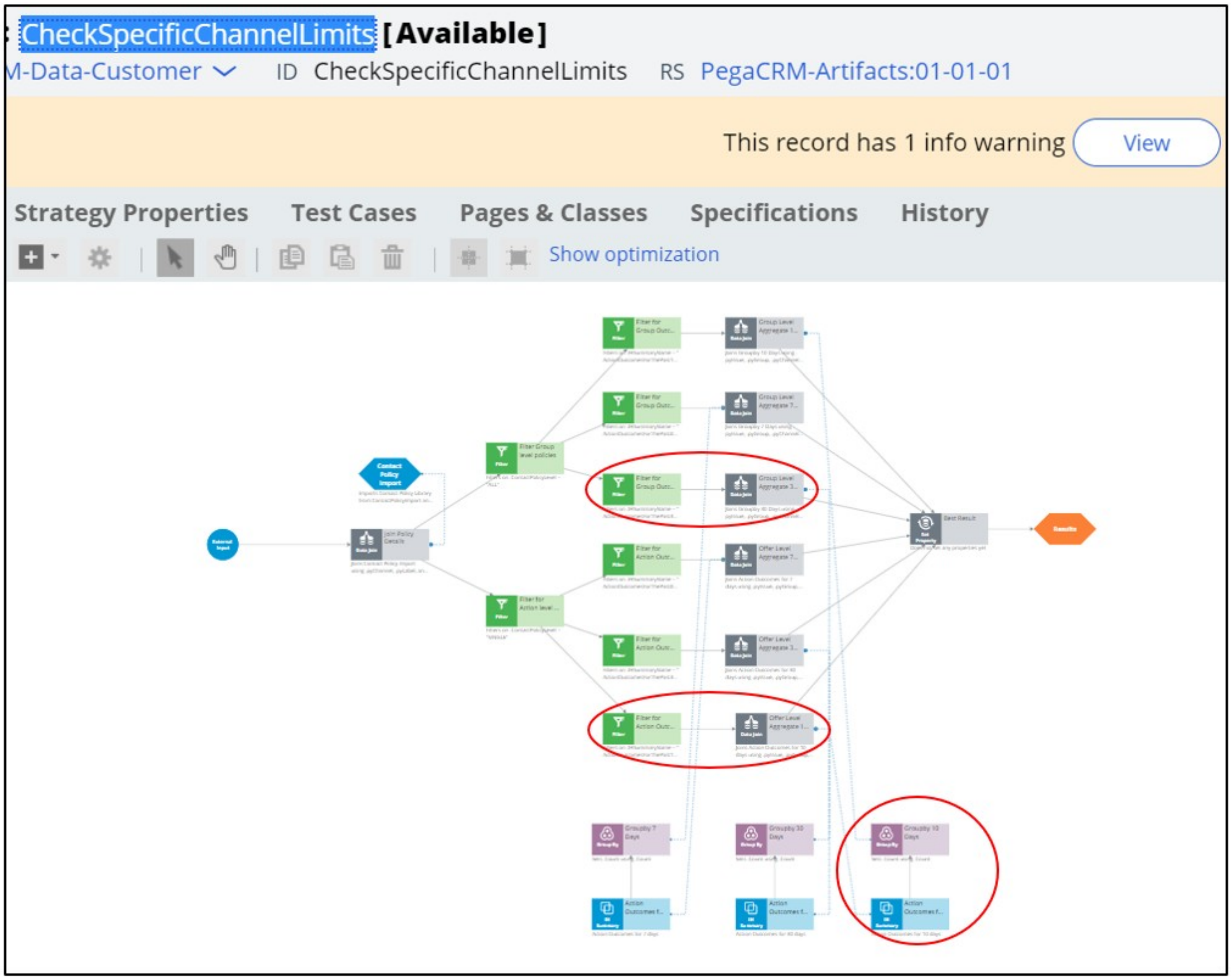
Also, the generated framework has some extension points where you can create strategies.

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.

Decision strategy canvas

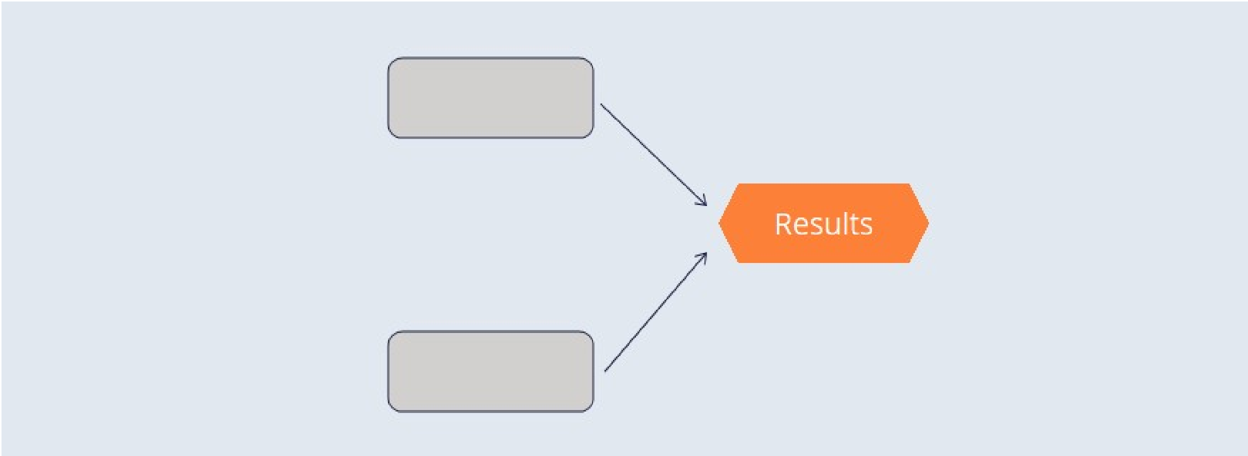
Introduction

Decision strategies drive the next best action and comprise a unit of reasoning represented by decision components. You use the Proposition Data component to import actions into a strategy canvas. The sequence of the components in the canvas determines which action is selected for a customer.

Click the Play button to learn more about decision strategies.

Screen1: U+ business scenario

U+, a telecom organization, wants to promote two new phones in the contact center: iPhone and Galaxy. Click the + icons to learn more about the elements of a decision strategy that is created for this requirement.



Decision Components: A decision strategy is comprised of building blocks called decision components. You can add and connect components to implement the business requirements.

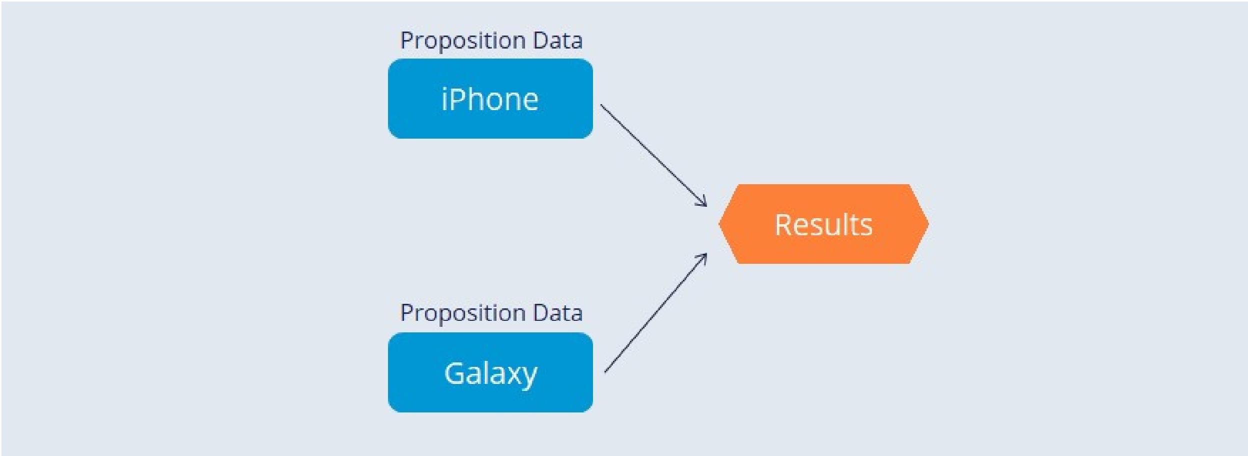
Arrows: An important element of the strategy canvas is the arrow. An arrow connects two decision components. A solid line means the data is copied from one component to another.

Strategy Canvas: In Pega, business users visually design decision strategies on what is known as a strategy canvas.

Screen2: Proposition Data component

The Proposition Data decision component imports the properties of an action. The result of this component is a flat list of all the properties.

Click the + icons on the proposition components to examine the components' results.



iPhone: This Proposition Data component outputs the Price and the Cost properties of the iPhone action.

Name: iPhone

Price: 150

Cost: 100

Galaxy: This Proposition Data component outputs the Price and the Cost properties of the Galaxy action.

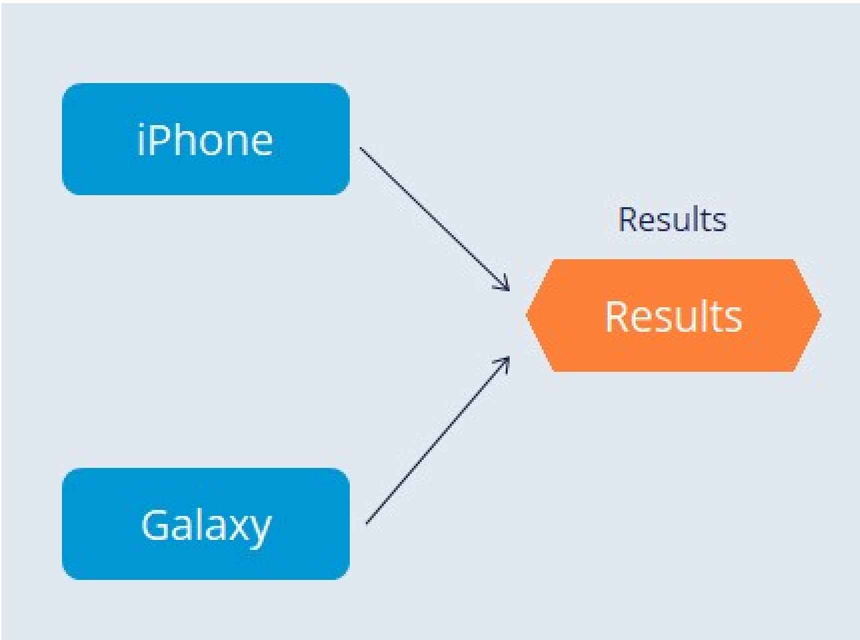
Name: Galaxy

Price: 250

Cost: 150

Screen3: Results component

Another decision component is the Results component. Each strategy always contains one Results component, which defines the output of the decision strategy.



How many actions do you think this strategy outputs?

0

1

2

Feedback: Because both iPhone and Galaxy are connected to the Results component with a solid arrow line, this decision strategy outputs two actions.

Dynamic pricing

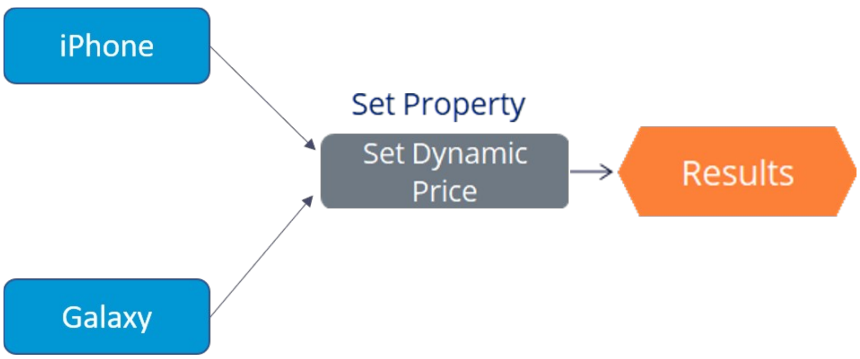
U+ Bank wants a dynamic Price for all offered actions. If the Customer value of a customer is higher than 60, the bank wants to offer a 10% discount to the customer.

To meet the new requirement, you must enhance the existing strategy to set the value of the Price based on Customer value. Changing the Price dynamically based on the Customer value makes the pricing customer-centric.

Set Property component

The Set Property component is used to dynamically alter the value of an action property based on a customer property. You use this component to set values to properties that are output by the strategy.

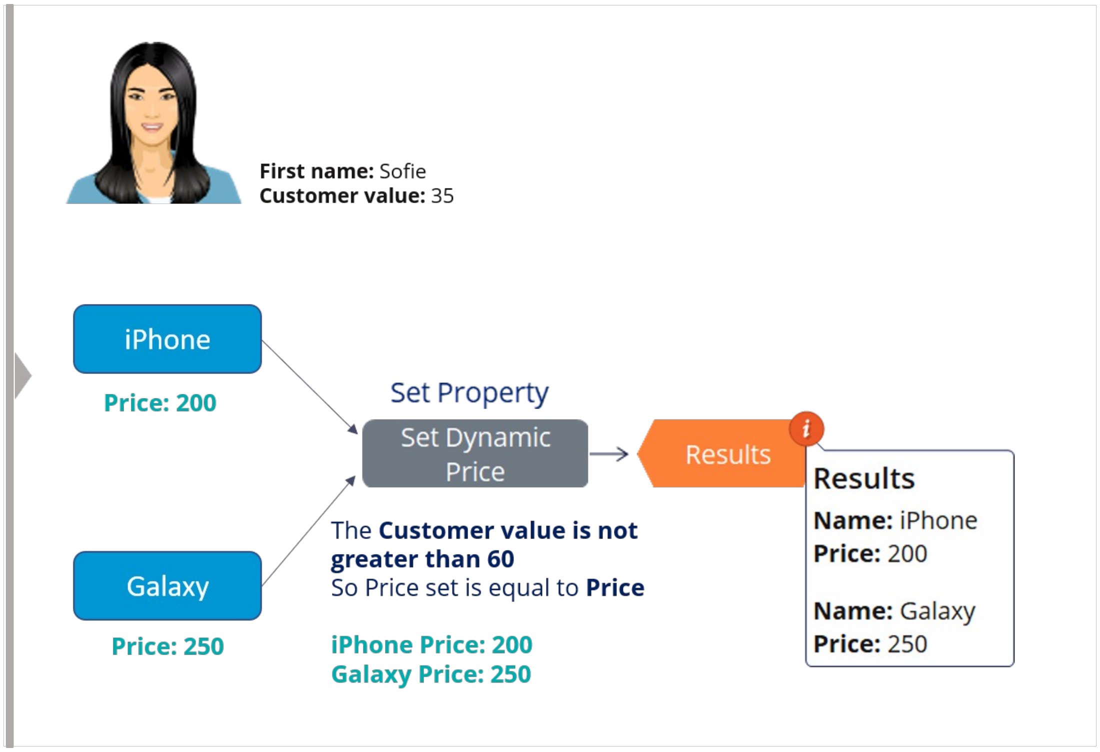
You can set properties to a constant or calculated value.

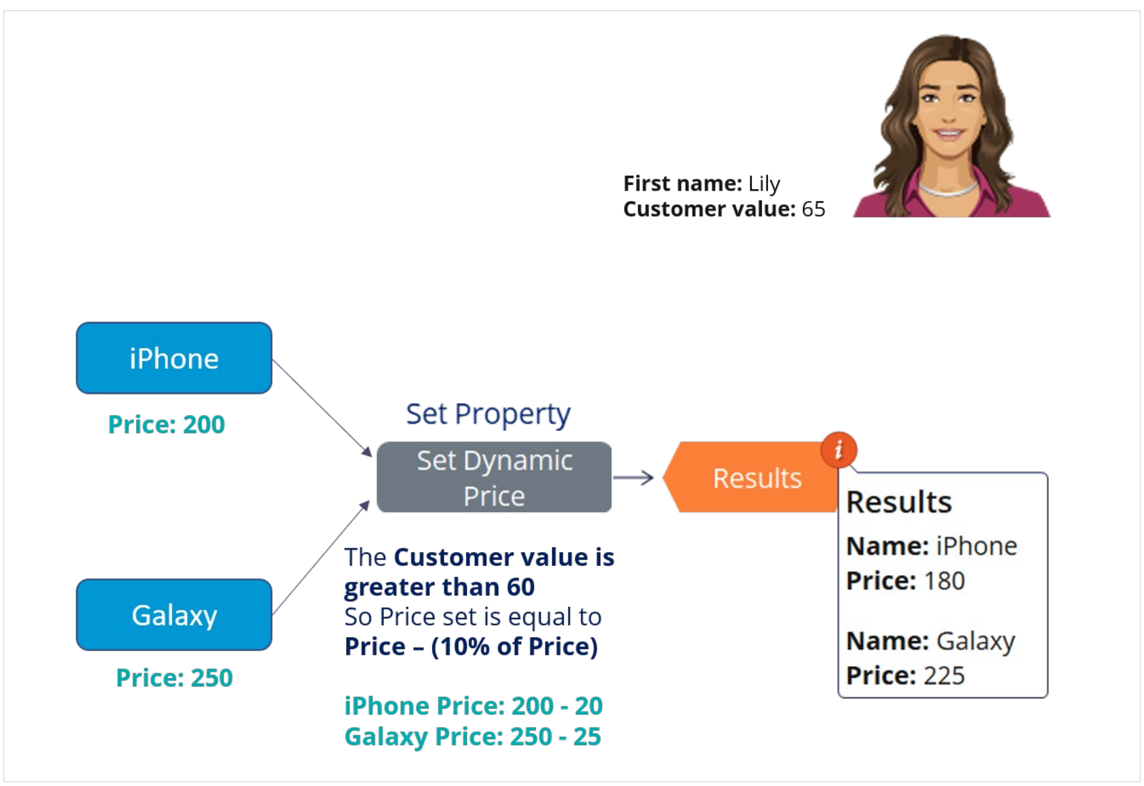


Example

Consider two customers: Sofie and Lily with customers value 35 and 65 respectively.

In the center of the following image, slide the vertical line to see how Sofie and Lily's Customer value affects the Price of the action offered to them.


First name: Sofie
Customer value: 35
iPhone
Price: 200
Galaxy
Price: 250
Set Property
Set Dynamic Price
Results
The Customer value is not greater than 60
So Price set is equal to Price
iPhone Price: 200
Galaxy Price: 250
Results
Name: iPhone
Price: 200
Name: Galaxy
Price: 250


First name: Lily
Customer value: 65
iPhone
Price: 200
Galaxy
Price: 250
Set Property
Set Dynamic Price
Results
The Customer value is greater than 60
So Price set is equal to Price – (10% of Price)
iPhone Price: 200 - 20
Galaxy Price: 250 - 25
Results
Name: iPhone
Price: 180
Name: Galaxy
Price: 225

Action ranking

U+ wants to offer the most profitable action to its customers.

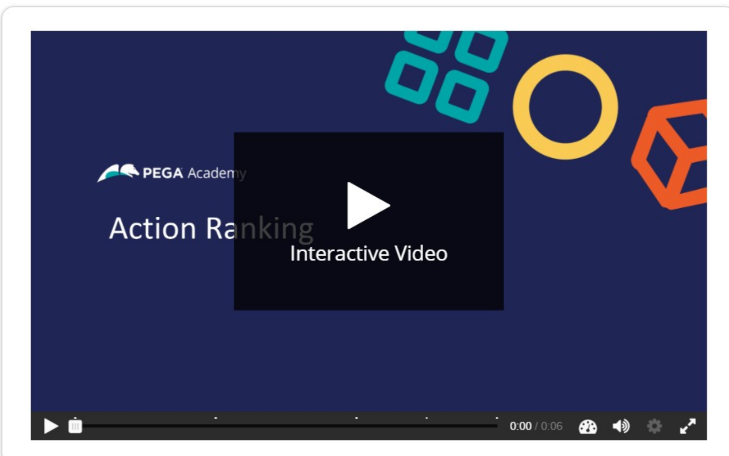
To enhance this strategy based on the new requirement, you need a new decision component that can rank the actions based on Profit and select the highest ranked action.

Profit is calculated based on Price and Cost action properties.

Prioritize component

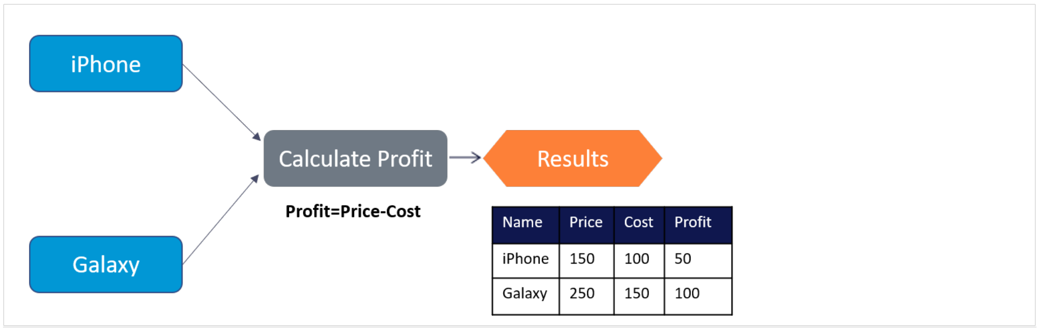
The Prioritize component is a decision strategy component used to rank actions. The Prioritize component is also used to select the top 1, top 2, or arbitrary top-n actions.

Click the Play icon to learn more about action ranking in detail with the help of a sample strategy.



Screen 1:

The initial strategy outputs price, cost and the profit calculated by the Set Property component. Click Add Prioritize to add the Prioritize component to the strategy.

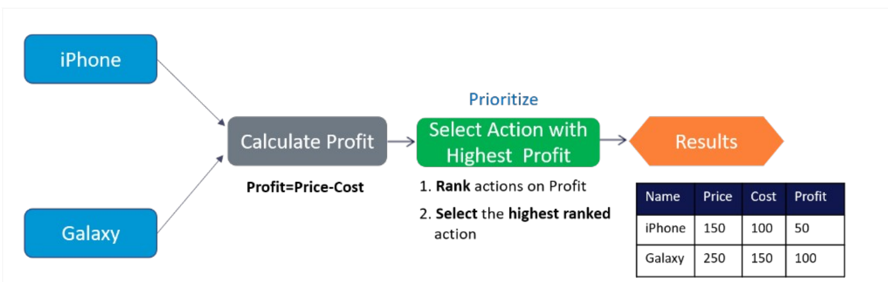


Name	Price	Cost	Profit
iPhone	150	100	50
Galaxy	250	150	100

Add Prioritize

Screen 2:

The Prioritize decision component can perform two operations: rank actions based on an expression, and select the highest ranked action. Click Rank actions to see how the component rank the actions.

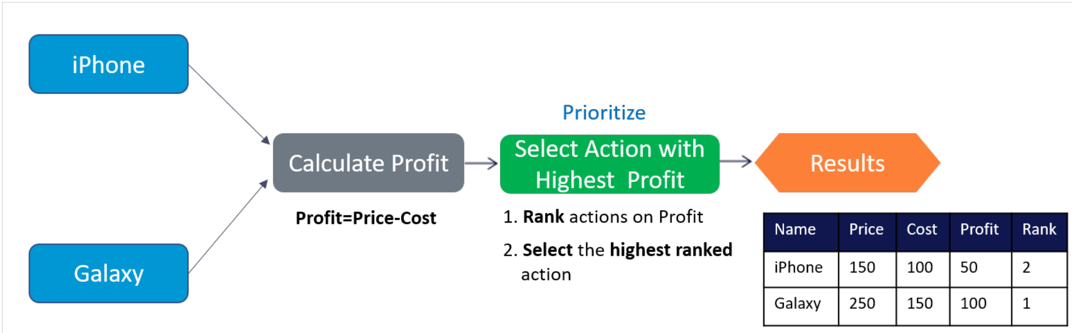


Name	Price	Cost	Profit
iPhone	150	100	50
Galaxy	250	150	100

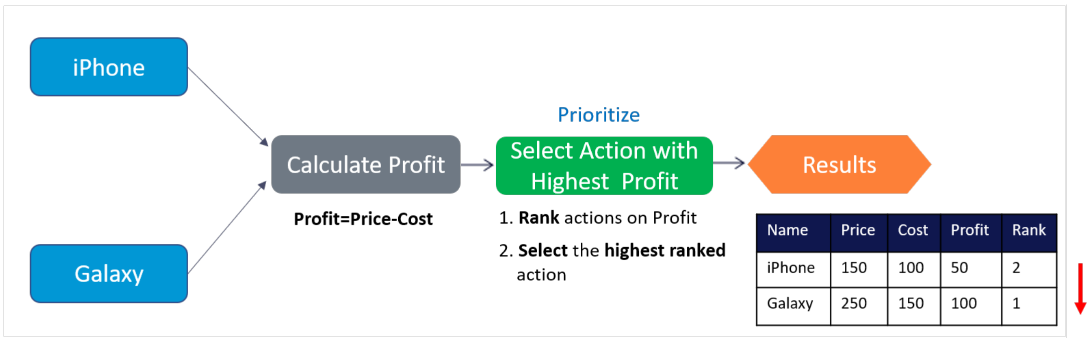
Rank actions

Screen 3:

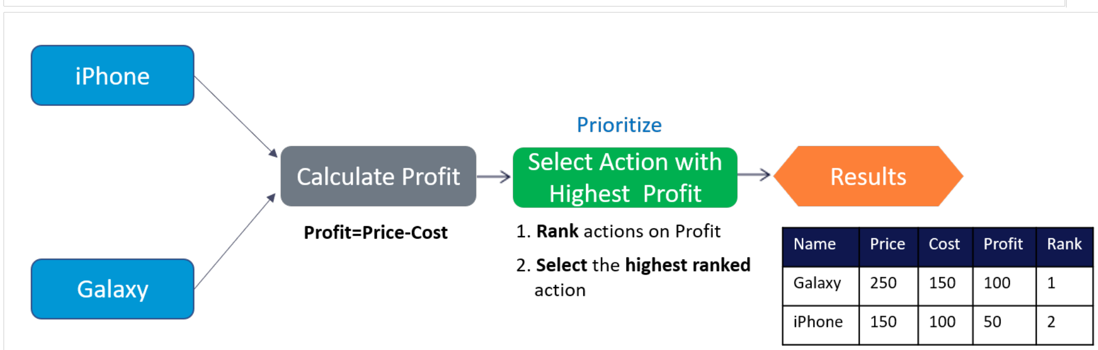
Once the actions are ranked, the prioritize component selects the highest ranked action. Click Select highest to see the action selected.



Name	Price	Cost	Profit	Rank
iPhone	150	100	50	2
Galaxy	250	150	100	1



Name	Price	Cost	Profit	Rank
iPhone	150	100	50	2
Galaxy	250	150	100	1

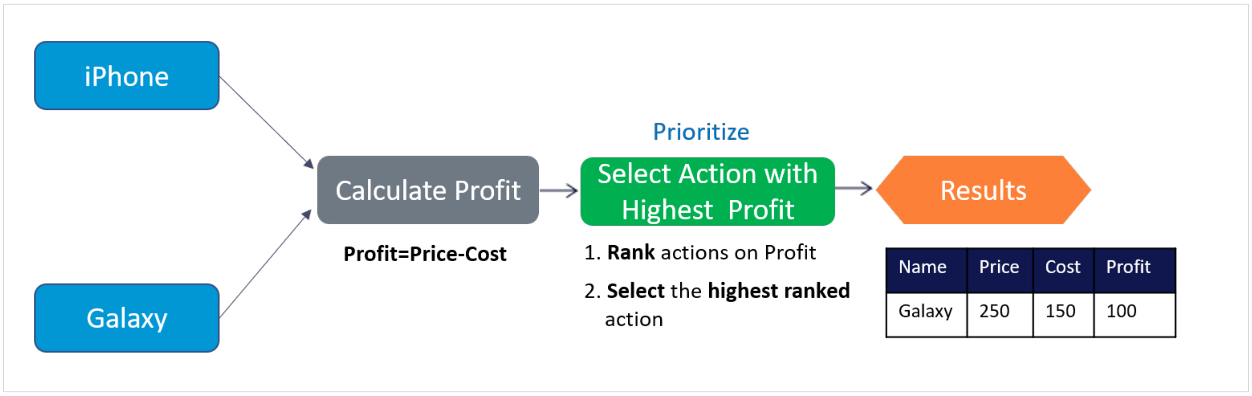


Name	Price	Cost	Profit	Rank
Galaxy	250	150	100	1
iPhone	150	100	50	2

Select highest

Screen 4:

You can see that the output of the result component is the highest selected action: Galaxy with a profit of 100.



Quiz

Examine this strategy and then answer the following question to check your knowledge on action ranking.

What does the Results component of the strategy contain?

Sony with profit 150

LG with profit 100

Panasonic with profit 50

Feedback: The Prioritize decision component ranks the actions and selects the highest ranked action. Hence, the Results component of the strategy contains Sony with a profit of 150.

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.

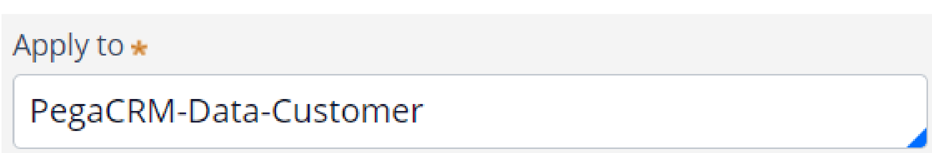
The Strategy-Results class limits the output of the strategy to the actions contained in the Business issue and Group.

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.

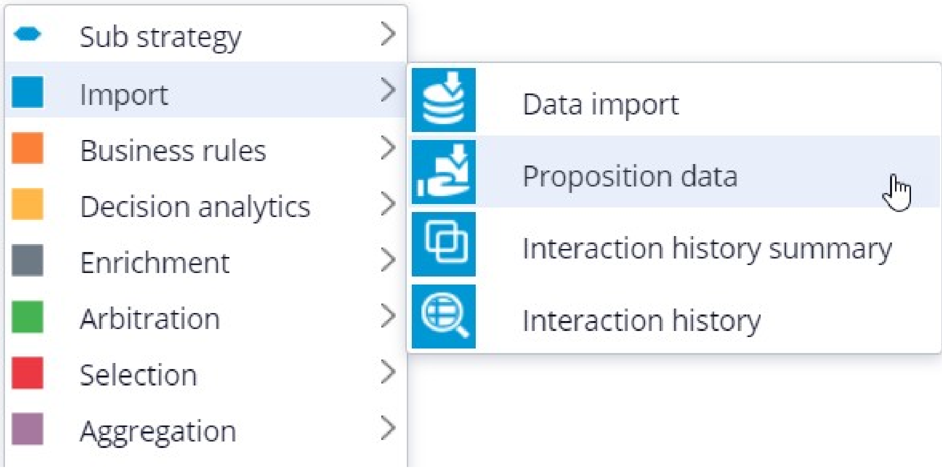


You can now start building the strategy. Right-click on the canvas to get the Context menu, which shows all component categories.

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.

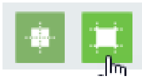
The description, which will appear under the component on the canvas, will also be generated automatically.

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.

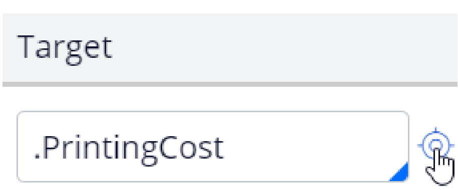
Click 'Add Item' to create the equation that will calculate the printing cost for each of the components.

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 BaseCost + 5 * LetterCount.

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

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

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

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

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

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

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

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

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

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

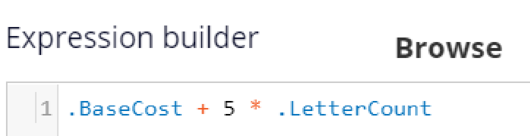
Suppose you want to use a customer property in your Expression, such as Age or Income.

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

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

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

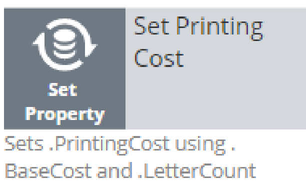
Finalize the Expression.



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.



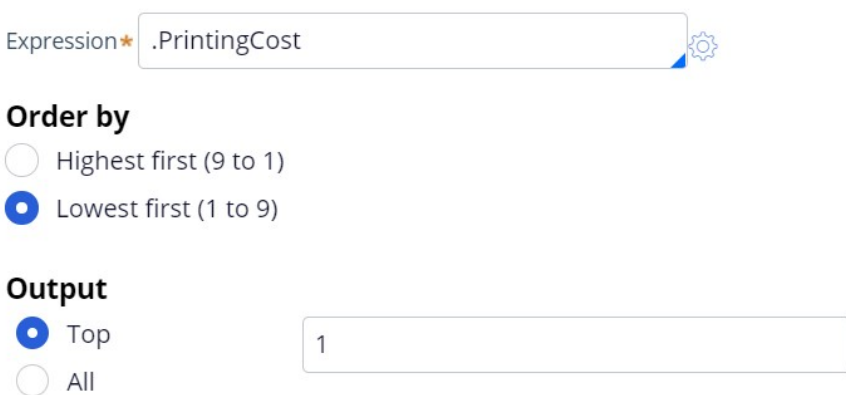
Now you want to ensure that the actions will be prioritized based on the lowest printing cost. So, you need to add the Prioritize component from the Arbitration category.

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.

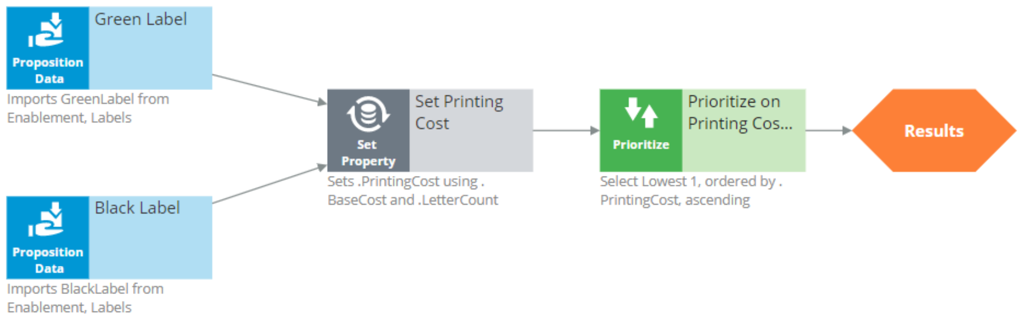
Here you can select the order in which the top actions are presented. Since you are interested in the lowest printing costs, configure it accordingly.

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.



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

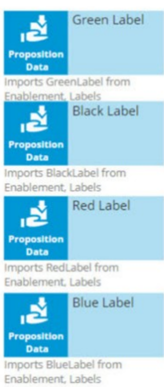
Using Pega Decision Management, you do not need to be an expert in programming, math or data science to design and execute sophisticated decision strategies that engage your customers throughout the customer journey. With its highly intuitive graphical canvas, Pega Decision Management enables you to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

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.

Rank	Name	BaseCost	LetterCount
1	Green Label	10	10

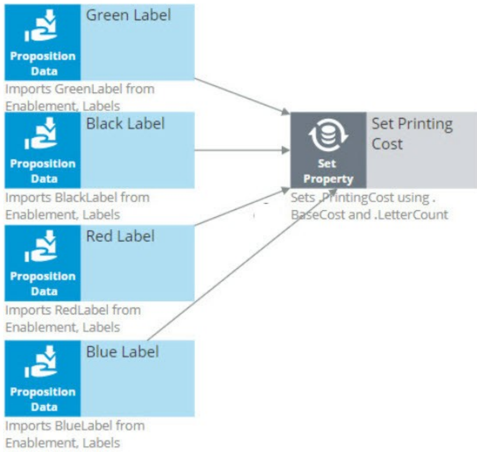
Rank	Name	BaseCost	LetterCount
1	Black Label	20	10

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.

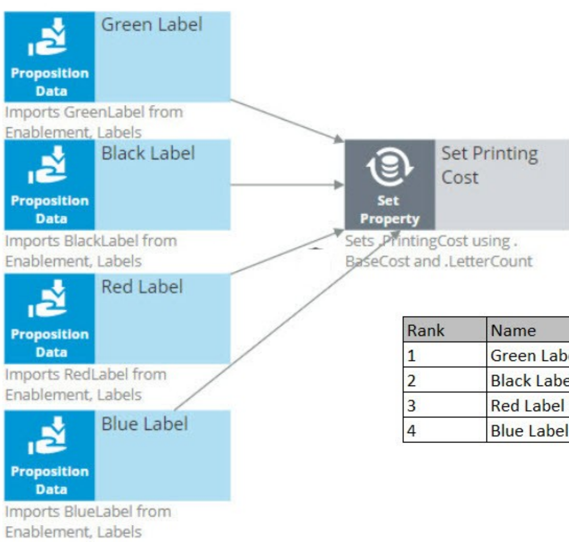
On the strategy canvas, components are connected by drawing arrows from component to component. So, what do these arrows mean exactly?



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.



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

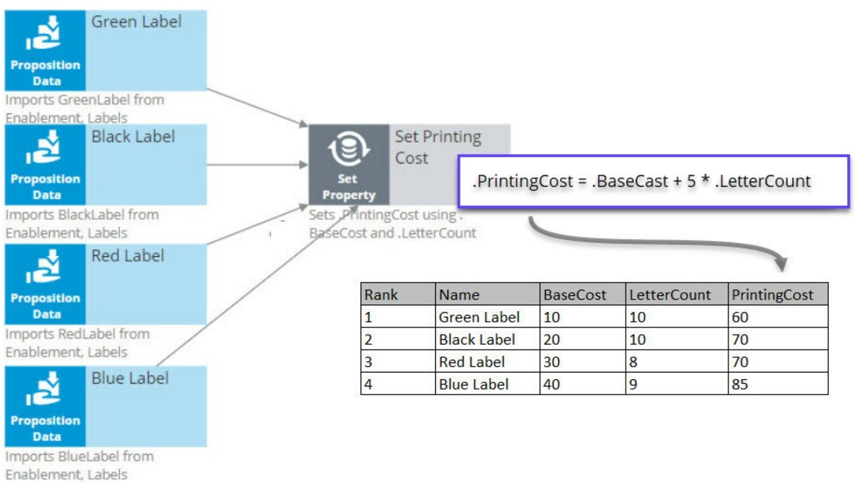
In the Set Property component, the Rank is determined by the order in which the actions are received by the component.

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

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

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

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



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

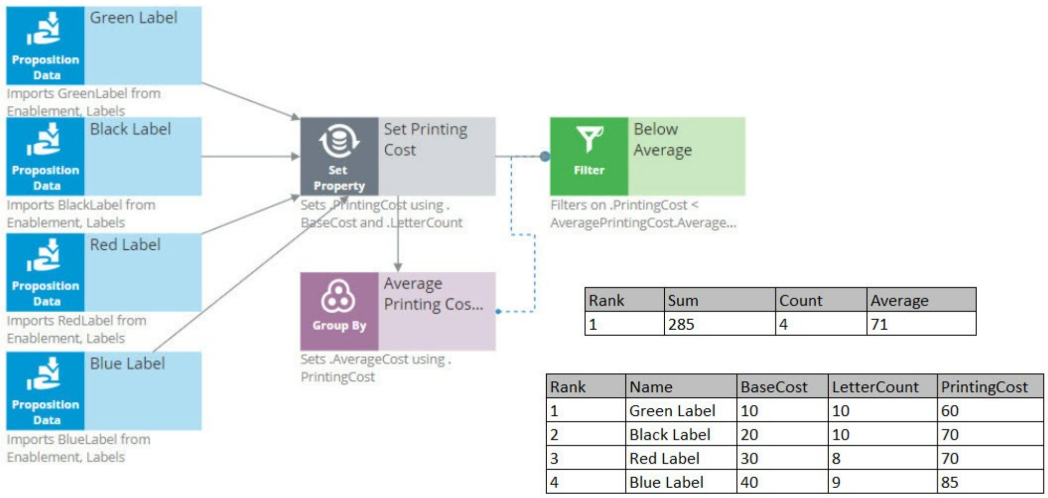
Because we are combining the data in our four Proposition Data components into one Set Property component, we only need to add one PrintingCost property to the new component, and it automatically computes the printing cost for all four actions.

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

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

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

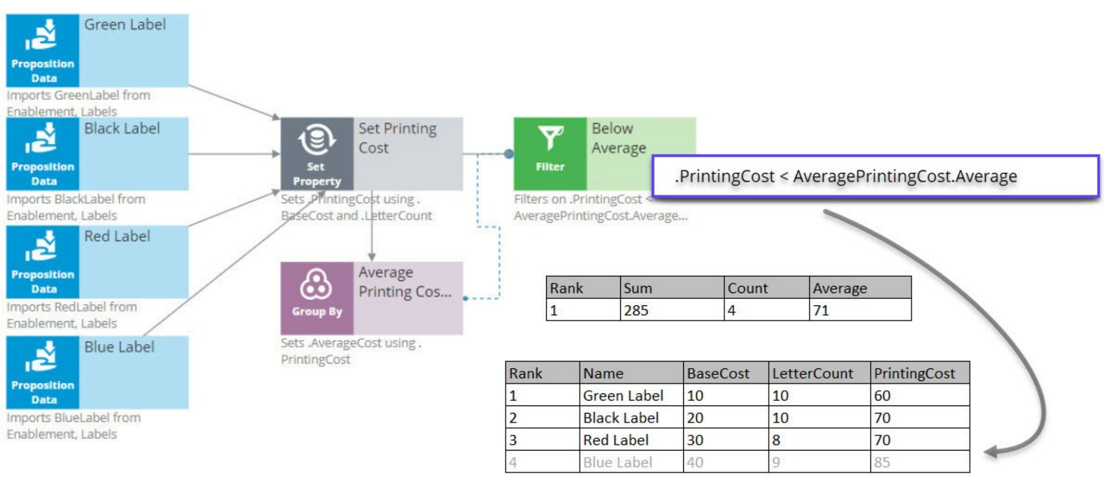
A 'Group by' component offers essential aggregation capabilities, like Sum and Count, that are used in many decision strategies. We will use it to calculate the average printing cost.



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.

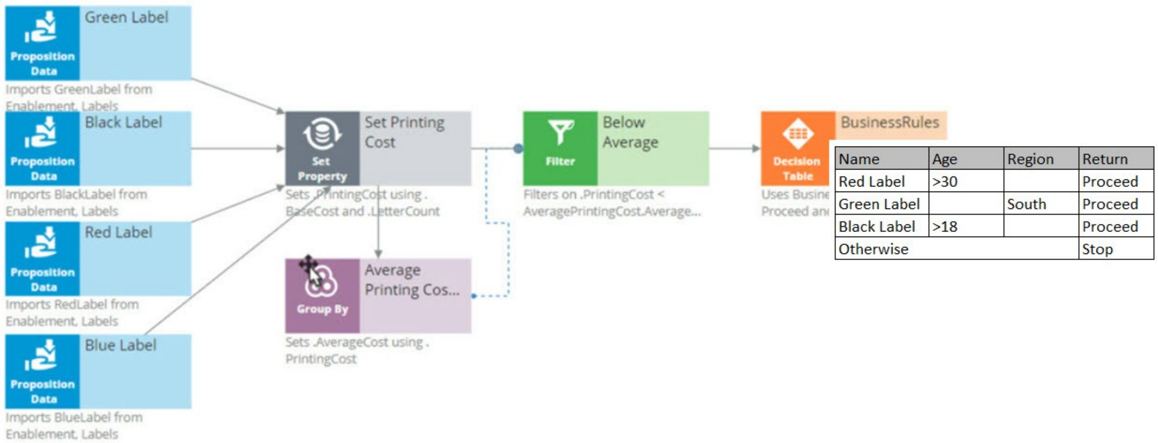
The Rank 1 average equals 71 in the ‘Group by’ component. This means that the filter will allow Label actions through that have a printing cost lower than 71.

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.

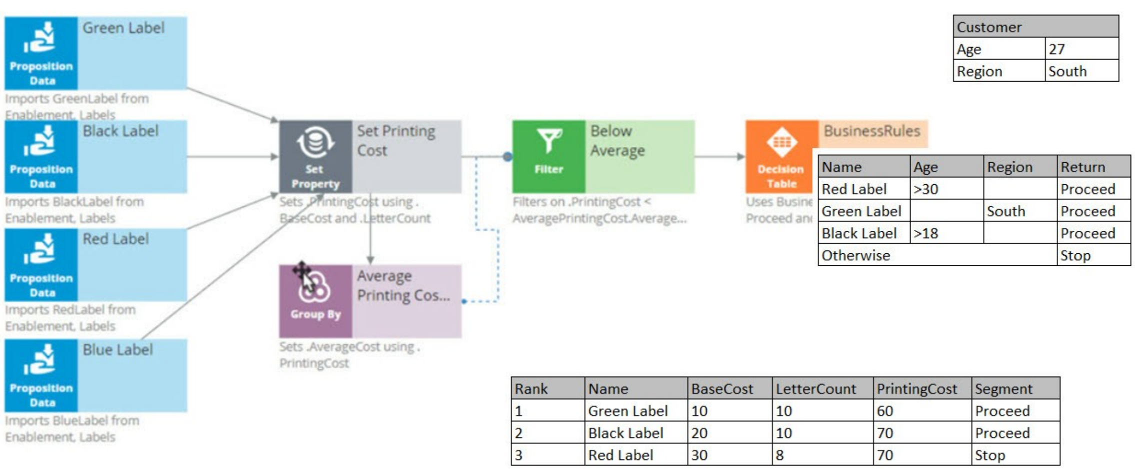
The business criteria say that the Red Label action can be offered if the customer's age is over 30 and they are from any region. If these criteria are met, the Return value is 'Proceed'.

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.

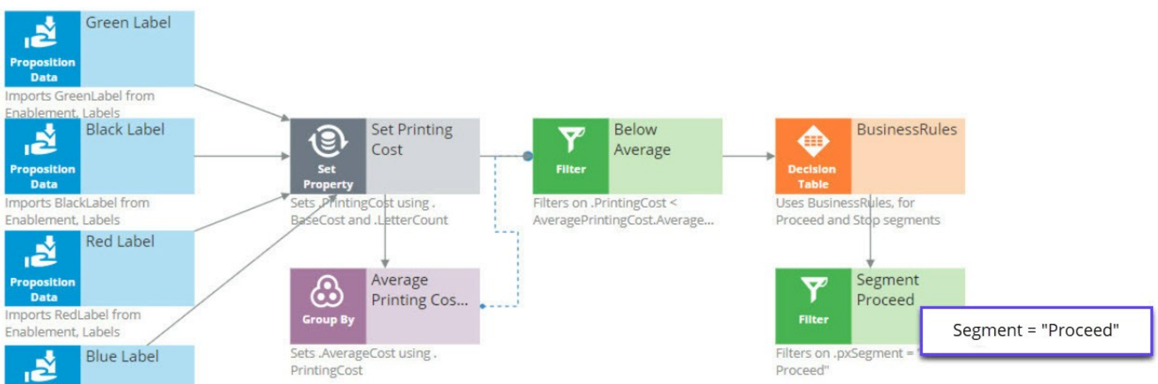
Next, it looks at the Black action and sees that the criteria for Black is that the customer's age is greater than 18. This customer is 27.

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.



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

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

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

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

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

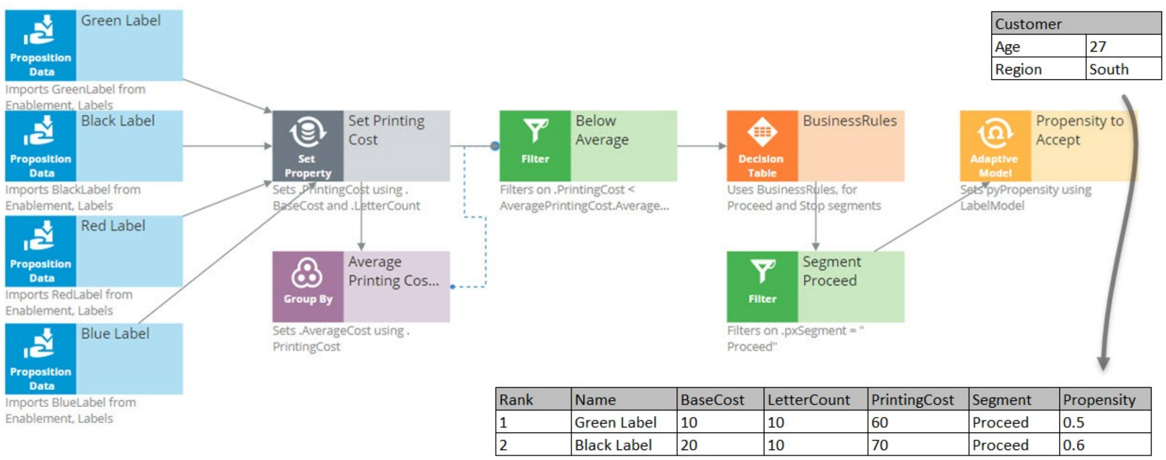
First is the Green Label. Green is allowed through, which means its properties will be available in the new component.

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

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

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

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



Customer	
Age	27
Region	South

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

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

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

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

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

The Adaptive Model determines the propensity.

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

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

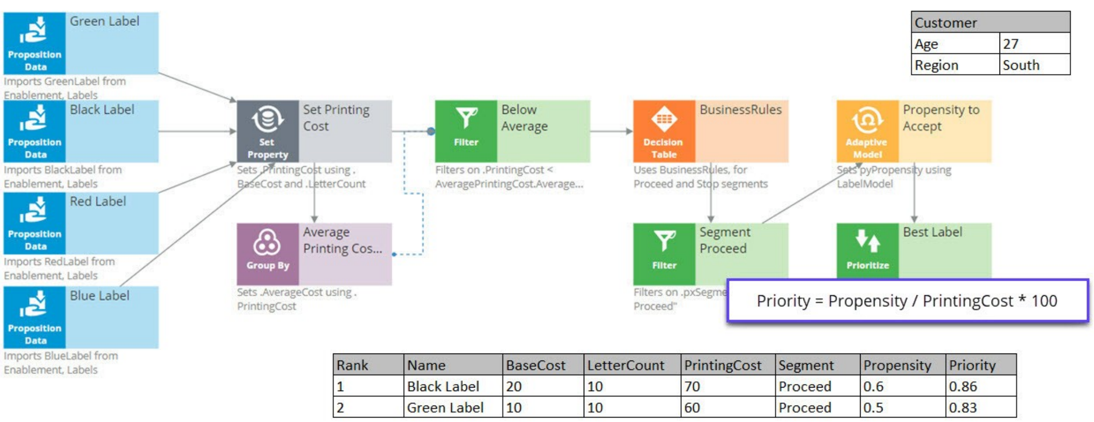
For the Black Label action, the likelihood turns out to be 0.6.

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

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

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

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



Customer	
Age	27
Region	South

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

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

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

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

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

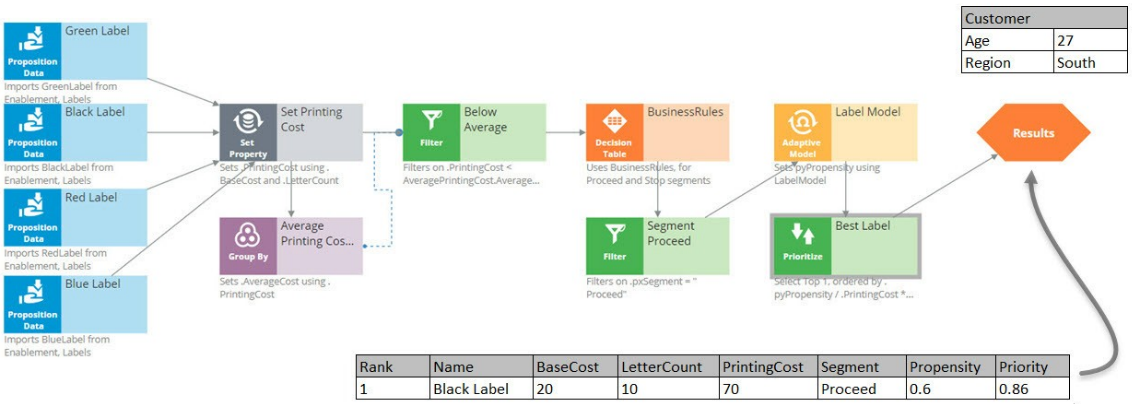
Because 0.86 is higher than 0.83, the Black Label action is now ranked number one.

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

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

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

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



Customer	
Age	27
Region	South

Rank	Name	BaseCost	LetterCount	PrintingCost	Segment	Propensity	Priority
1	Black Label	20	10	70	Proceed	0.6	0.86

Creating predictions

Description

Predicting customer churn is one among many business use cases involving predictive models. Pega Customer Decision Hub™ uses predictions that use predictive models to improve one-to-one customer interactions. Learn how to create a new prediction in Prediction Studio and use the new prediction in an engagement strategy.

Learning objectives

- Create a new prediction that uses a predictive model
- Use a prediction in an engagement strategy

Creating a prediction

Introduction

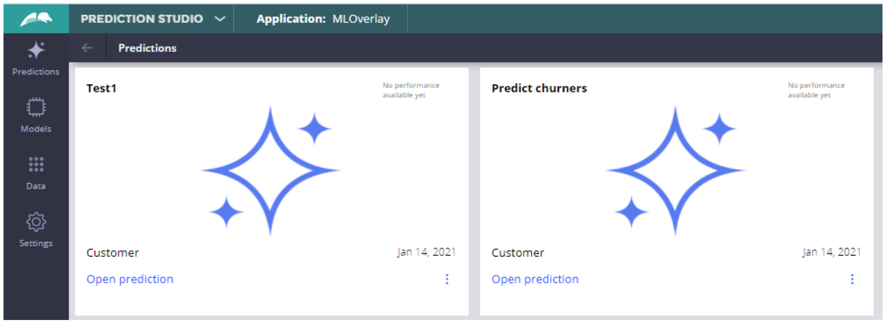
Acquiring new customers can be more costly than retaining current customers. U+ Bank uses Pega Customer Decision Hub™ for their customer engagement and wants to reduce the churn rate. Learn how to create a new prediction in Prediction Studio that calculates the likelihood that a customer might churn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

Transcript

This demo shows you how to create a new prediction in Prediction Studio. 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 now ready for use in Customer Decision Hub.

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

Using predictions in engagement strategies

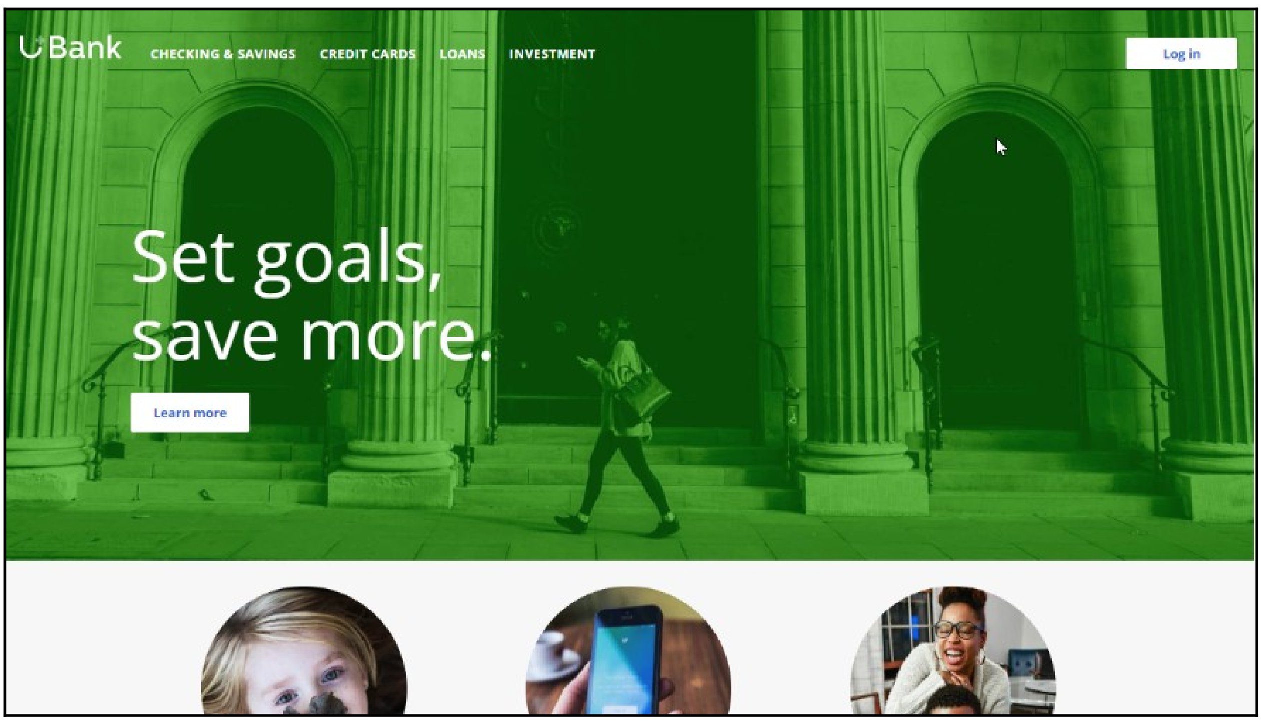
Introduction

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.

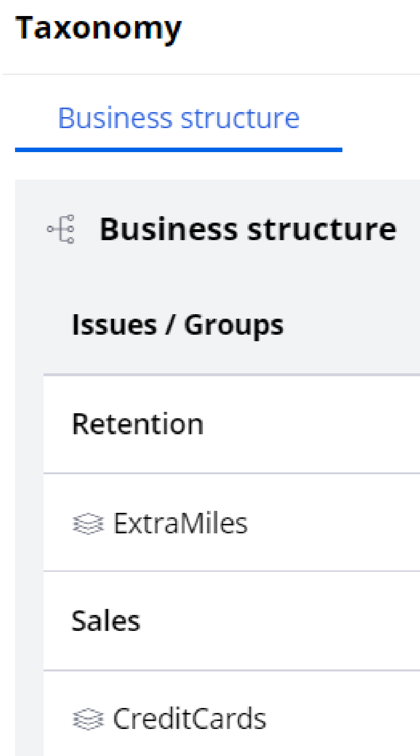
Transcript

This demo shows you how to use a prediction in an engagement strategy to determine customer applicability for a retention offer.

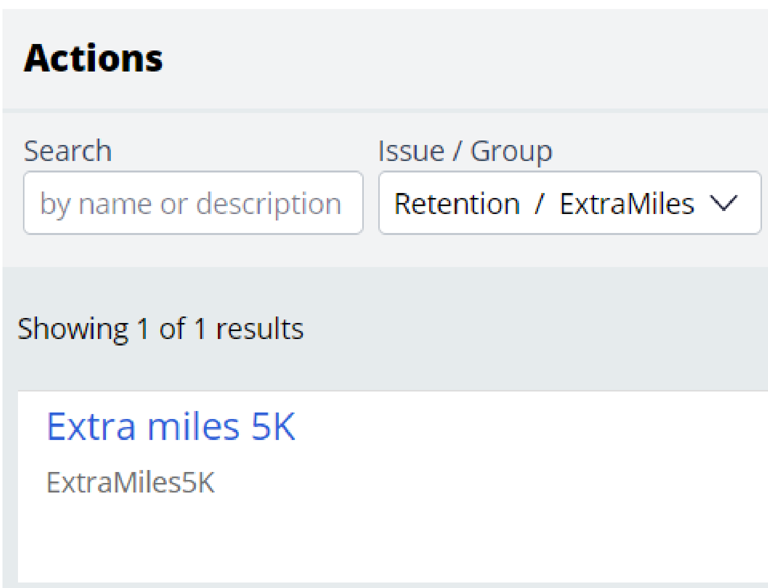
Currently, U+ Bank is cross-selling on the web by showing various credit cards to eligible customers who log in to its website. The bank now wants to show a retention offer, instead of a credit card offer, to customers who are likely to churn in the near future. The credit card offers are shown only 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.



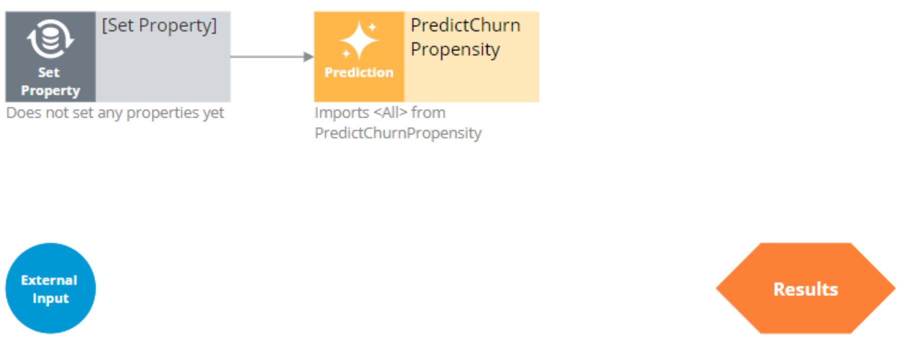
This ExtraMiles group contains a retention offer, Extra Miles 5K.



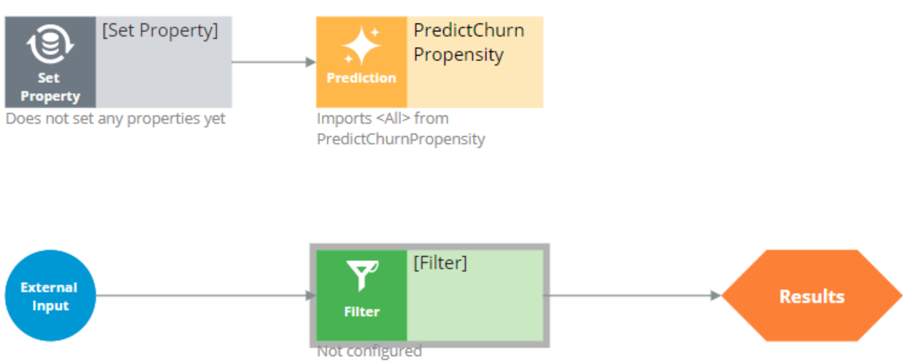
The next step is to create an applicability condition that makes a customer qualify for a retention offer when there is a high likelihood that the customer might churn. A data scientist has created a prediction that identifies these high-risk customers. When you open the prediction in Prediction Studio, 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.

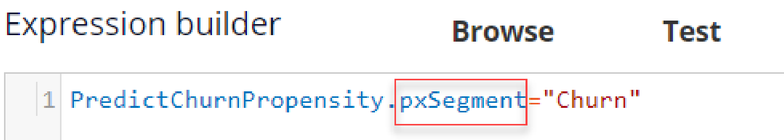
To define the applicability condition, you create a decision strategy to output a retention offer only if the response label of the prediction is **Churn**. Add a **Prediction** component to the canvas and configure it to reference the churn prediction. Add a **Set Property** component and connect it to the Prediction component. You can configure the Set Property component at a later point to accommodate parameterized fields.

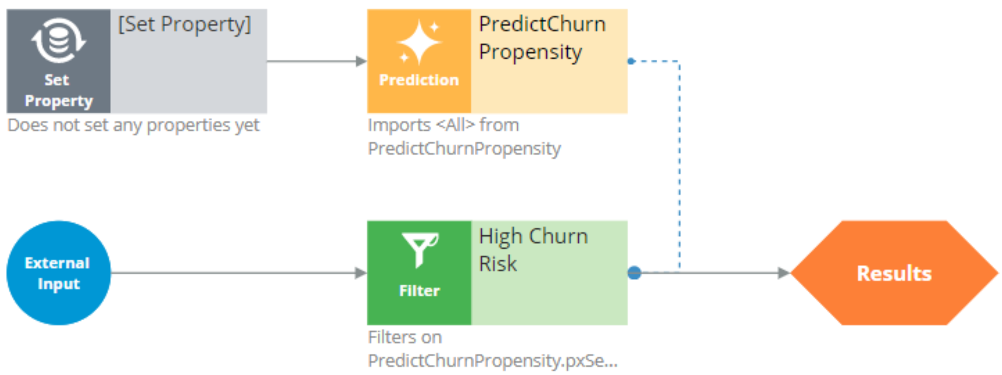


Next, add a filter component to filter out the loyal customers and pass retention offers to high churn risk customers only.

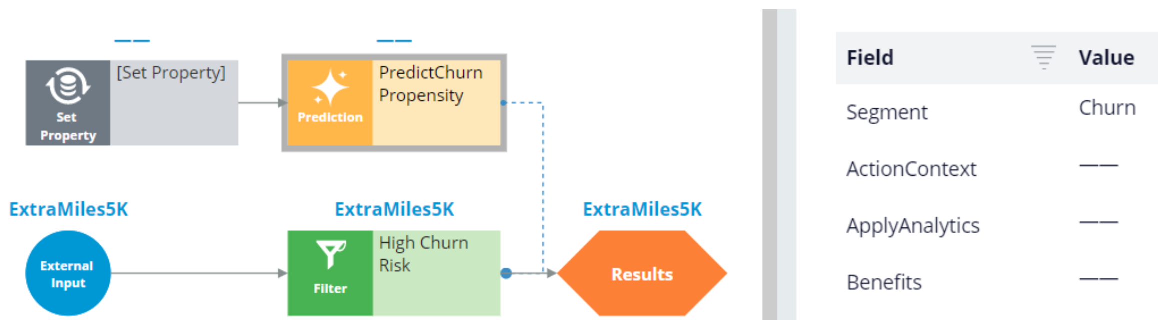


The filter condition is defined to output a retention offer when the pxSegment property of the prediction is equal to **Churn**.

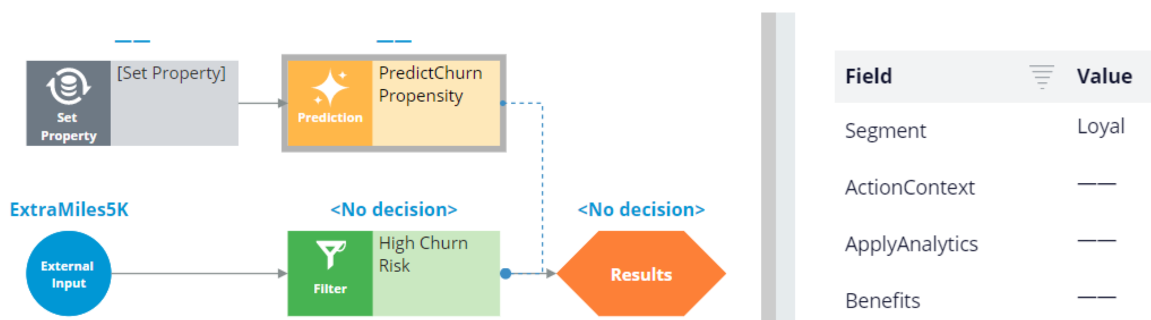




Next, test the strategy using two customer profiles, **Troy** and **Barbara**. For external inputs, consider all available retention offers. The strategy outputs a result for **Troy** because the result of the prediction is **Churn**.



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



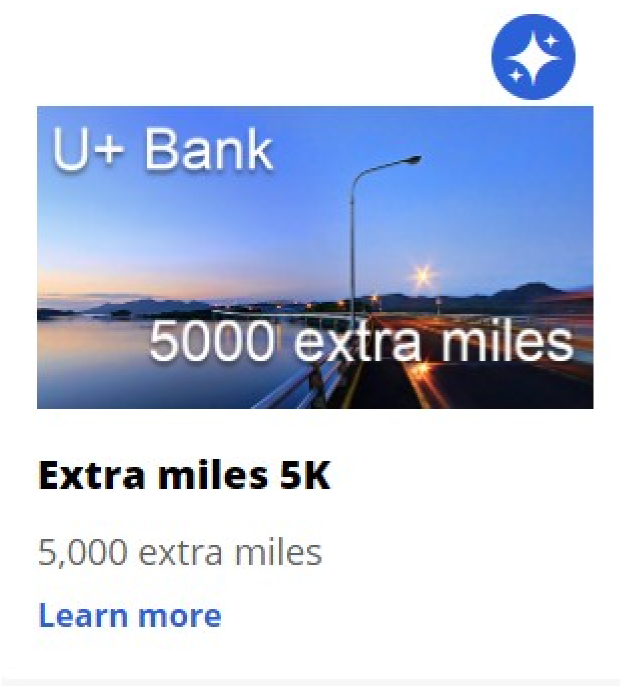
By checking in the strategy, you commit your changes so that they go into effect. You can now use this strategy in the Next-Best-Action Designer engagement policy as an applicability condition.

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.

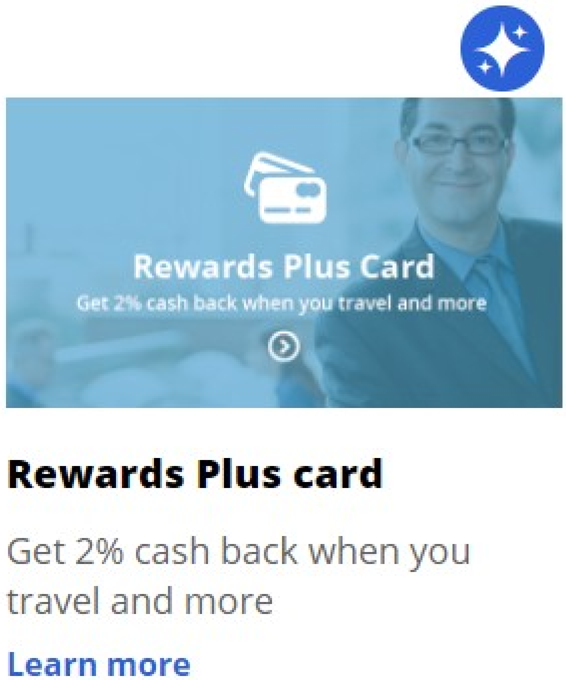
The second business rule you need to implement is: U+ Bank wants to show credit card offers to low-risk customers only; meaning the **CreditCards** group is not applicable for high-risk customers. To implement this rule, modify the **Applicability** section of the **CreditCards** group. The condition 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.



Now, when you log in as **Barbara**, notice that the credit card offer is displayed because she is predicted to remain loyal for now.



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

Creating predictive models

Description

In Prediction Studio, three option to leverage historical data are available: creating models using Pega machine learning, importing models created in a third party tool and referencing external models. Learn how to create, import and reference predictive models that can be used in driving the next best action.

Learning objectives

- Describe the role and usage of predictive models in the Pega landscape
- Use Pega machine learning to build predictive models
- Import third party predictive models
- Use machine learning services
- Explain the model transparency settings in Prediction Studio

Predictive models

Introduction

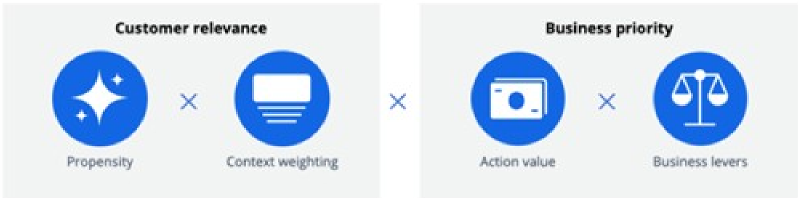
Enhance decision strategies with predictive models built on customer interaction data and let Pega Customer Decision Hub™ bring even more relevance to every customer engagement. Build models using Pega's machine learning capabilities, import models built with third-party tools and incorporate the latest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

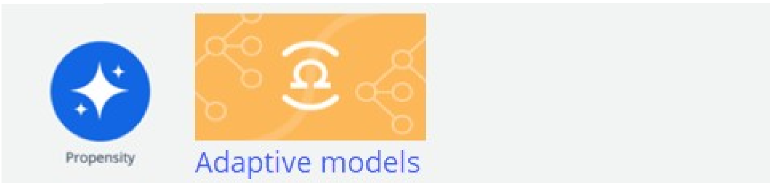
Next best actions balance customer relevance and business priorities by selecting the actions with the highest priority.

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.

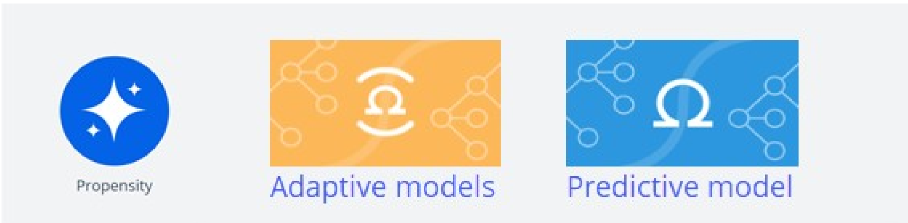
This is calculated by predictive models. In Pega, self-learning Naive Bayes models, which are generated for each action, are a key feature.



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.



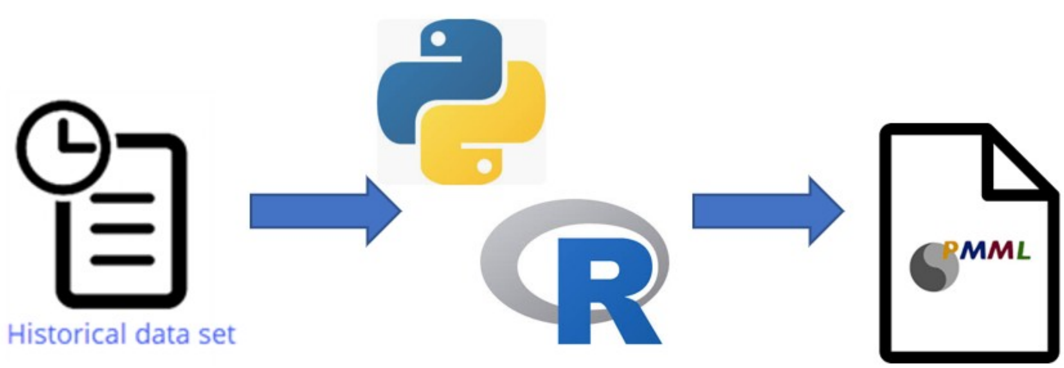
The first method is to use Pega machine learning. You can import a file containing the historical customer interaction data set and build a model in Prediction Studio.

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 modelling 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.

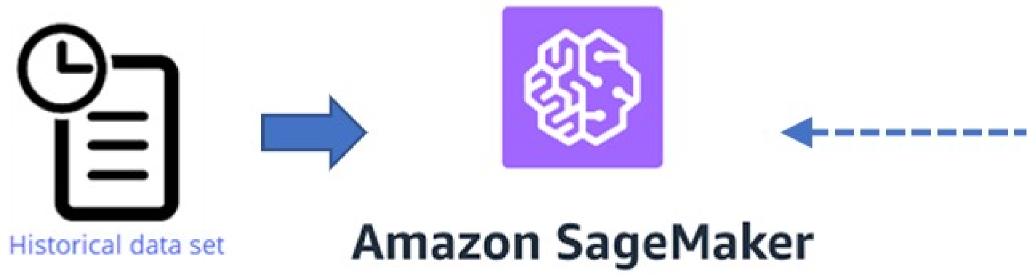
When decision strategies using the imported models execute, the models are executed inside the Pega platform.

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

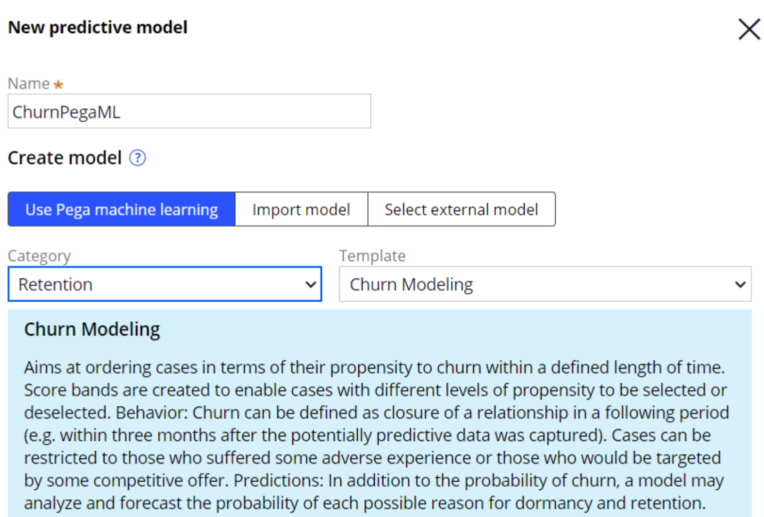
Prediction Studio offers several options for leveraging customer data to create predictive models. Learn how to develop powerful and reliable models that can predict customer behavior, such as offer acceptance, churn rate, credit risk, or other types of behavior by using Pega machine learning.

Transcript

This demo will show you to 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.

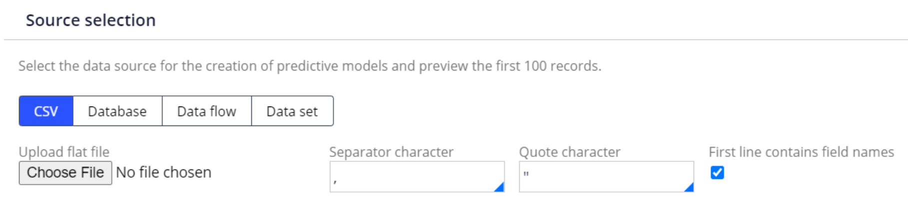
When starting to build a new model, you will be presented with the option to create a model on a template that is used for streamlining model development. One of these is churn modeling.



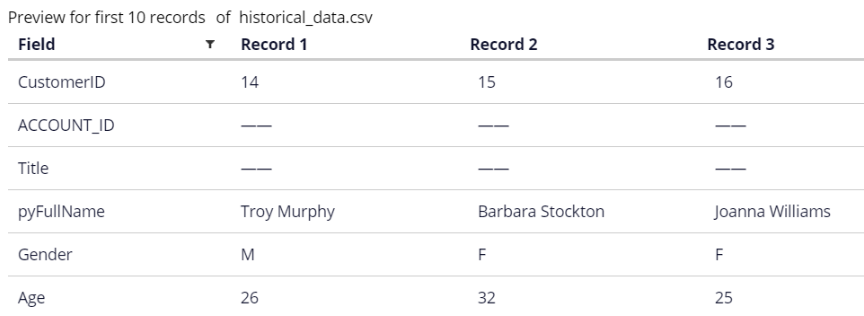
The model build itself consists of 5 steps: Data preparation, data analysis, model development, model analysis, and model selection.

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.



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

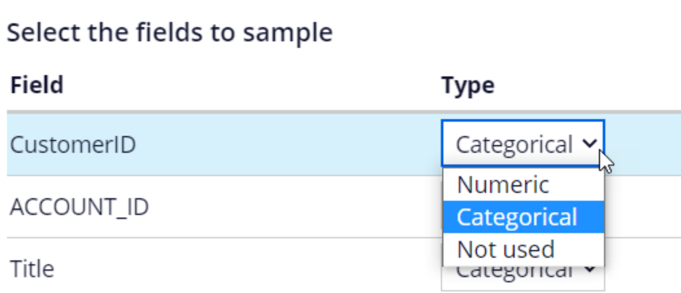

Preview for first 10 records of historical_data.csv

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.

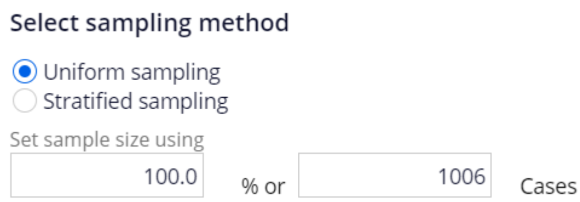


By default, all fields are considered potential predictors. When setting predictors, it's important to use some common sense.

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'.

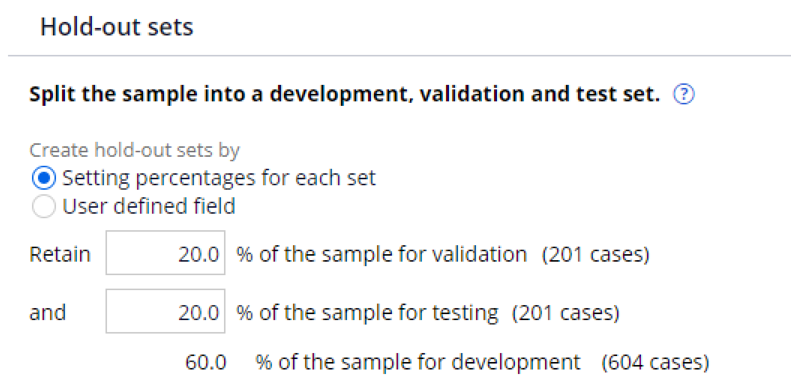
If the data contains a relatively small number of cases, you will want to use 100% of the records. If the data source is large, a sample will be sufficient.



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.



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

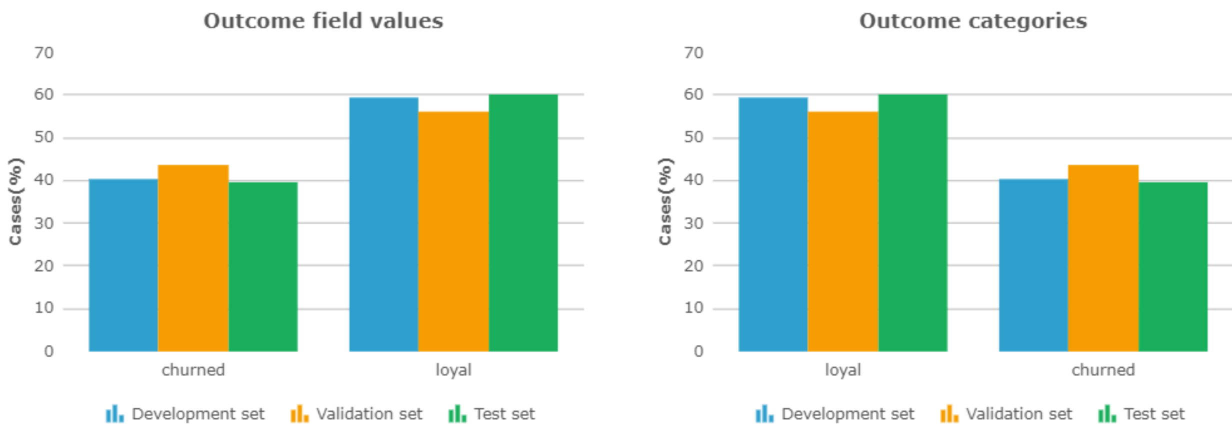
Define the outcome to be predicted. Predict a binary outcome, only categorical fields can be selected for this. Or predict a continuous outcome, only numeric fields can be selected for this.

Outcome type	Outcome field to predict
Binary	Segment

Map possible values of outcome field to outcome category

	Value		Outcome category
Map	loyal	to	loyal
Map	churned	to	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 52.0 Apply

Change role | Reports | New virtual field | Reset | Show groups

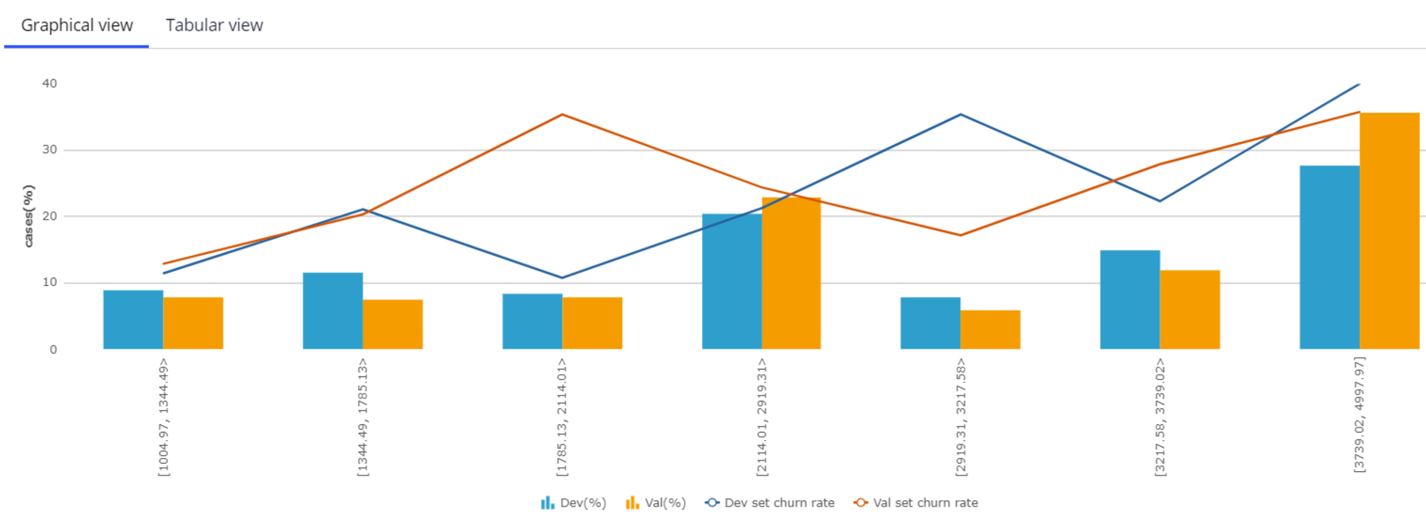
Predictor	Type	Role	Binned intervals	Grouped intervals	Grouped performance
RiskScore	Numeric	VALUE	200	11	90.27
AverageSpent	Numeric	PREDICTOR	200	7	**68.28**
MonthlyPremium	Numeric	PREDICTOR	200	10	**64.84**
Age	Numeric	PREDICTOR	70	8	**63.87**

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*CLV is such a virtual field. The performance of this new predictor is higher than that of the individual fields.

Predictor	Type	Role	Binned intervals	Grouped intervals	Grouped performance
RiskScore	Numeric	PREDICTOR	200	11	**90.27**
AverageSpent	Numeric	PREDICTOR	200	7	**68.28**
Income*CLV	Numeric	PREDICTOR	200	10	**65.08**
MonthlyPremium	Numeric	PREDICTOR	200	10	**64.84**
Age	Numeric	PREDICTOR	70	8	**63.87**
Income	Numeric	PREDICTOR	200	10	**63.74**
CLV_VALUE	Numeric	PREDICTOR	200	9	**63.47**

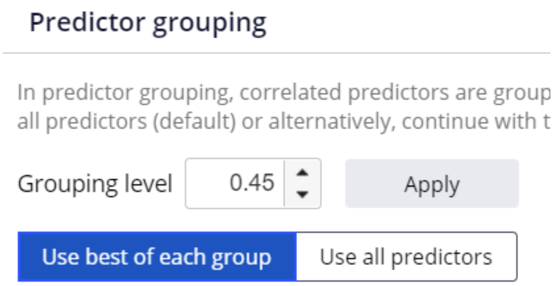
Data analysis creates a binned, ordinal view of individual predictors. Both Binning and Granularity are automatically set but can be manually adjusted.



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.



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.

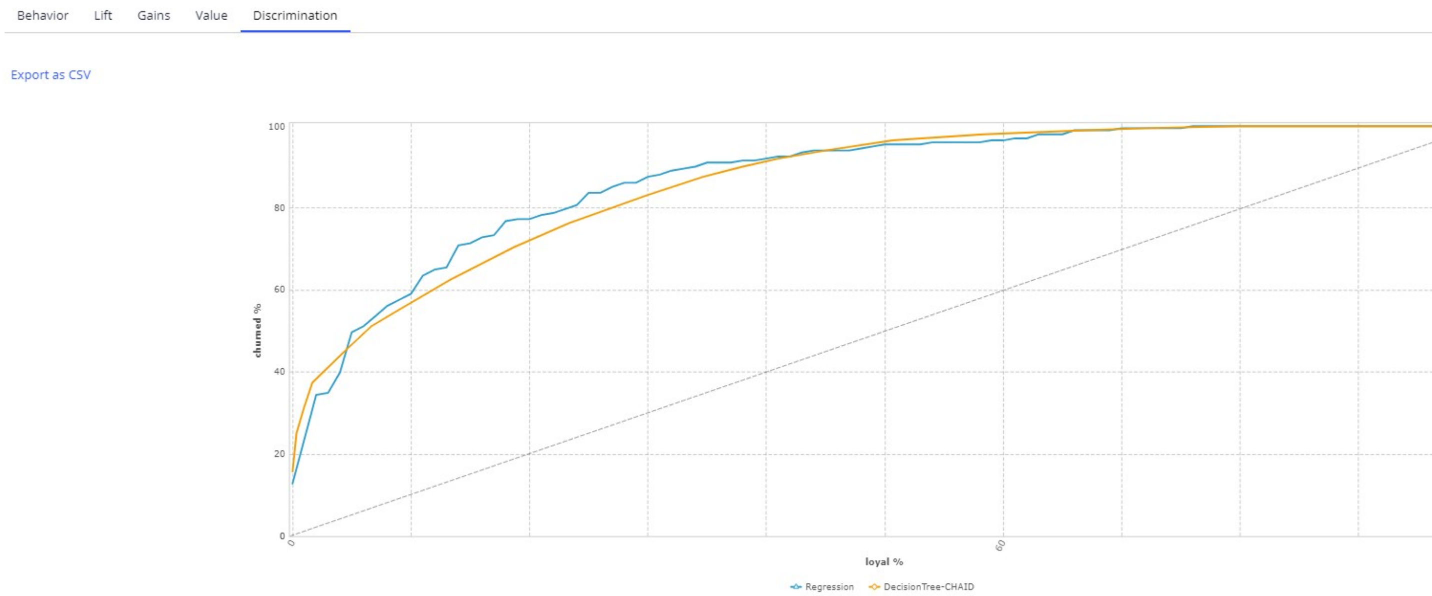
The purpose of the next step, Model Analysis, is to select the best model for your use case.

In the ‘Score comparison’ step, you can compare the scores generated by the models in terms of behavior, lift, gains and discrimination.

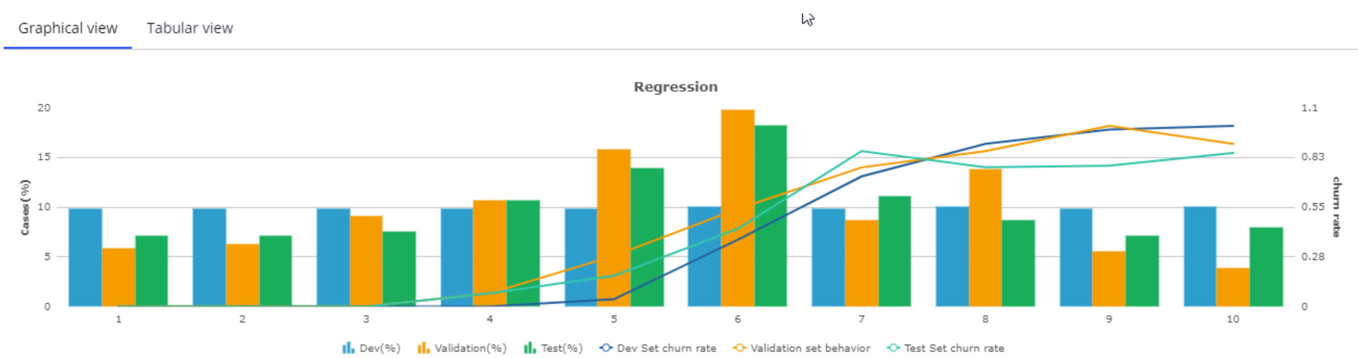
Prediction Studio uses Area Under the Curve (or AUC) to measure the performance of predictors and models.

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.

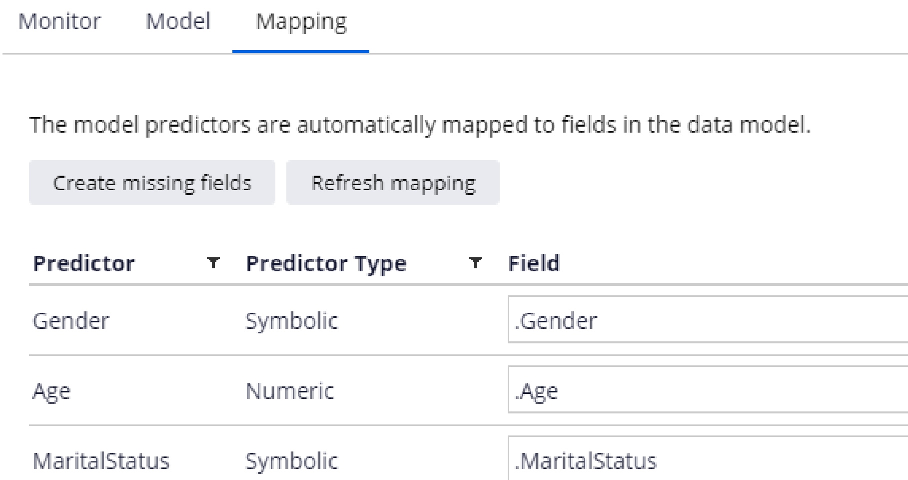
In the ‘Class comparison’ step, you can analyze and compare models after the score distribution has been adjusted.

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.

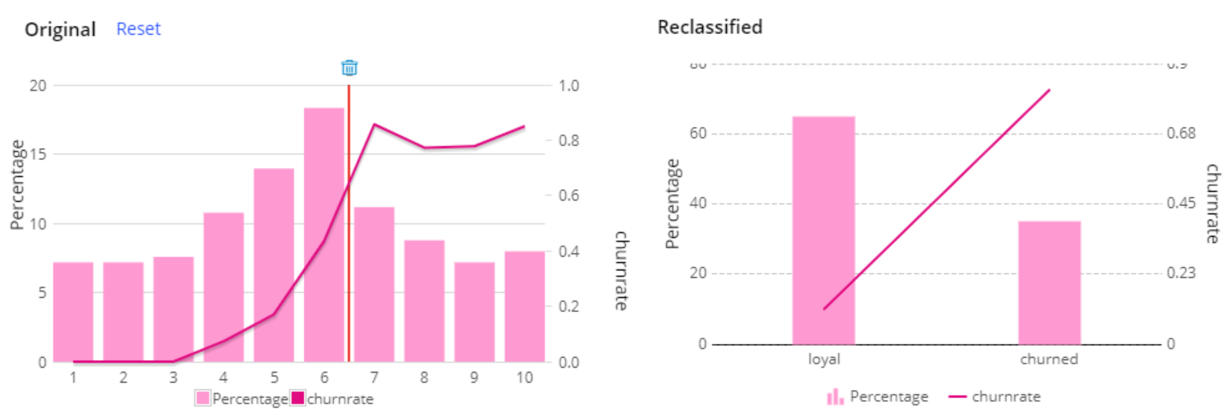
Before you can save the model, check the mapping of the predictors to the properties in the customer class.

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.



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

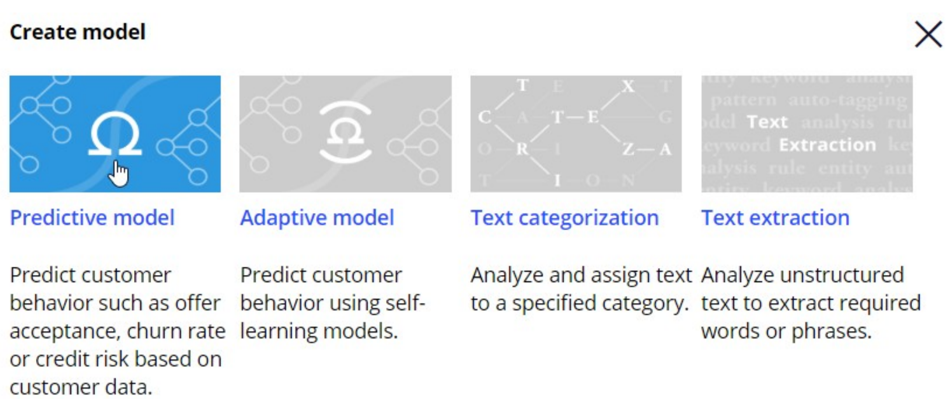
During a Pega Decision Management implementation project, you may discover that the company already uses predictive models. These assets can be reused in Pega Decision Management to help make customer predictions.

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.

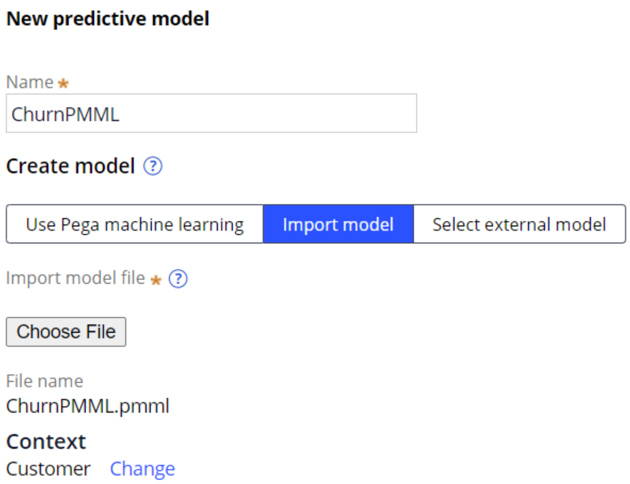
Prediction Studio supports two external model formats. First, you can import models in the Predictive Model Markup Language (PMML) format. PMML is an XML-based language aimed at easily sharing predictive models between applications. It is the de facto standard for representing not only predictive models,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

Additionally, you can import models built with H20.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 H20 models are identical and start with creating a new predictive model strategy component.

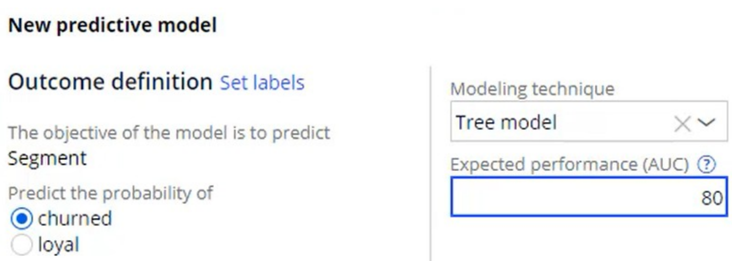


Prediction Studio offers three options for creating a predictive model: using Pega machine learning, importing a previously built model, or using an external model.

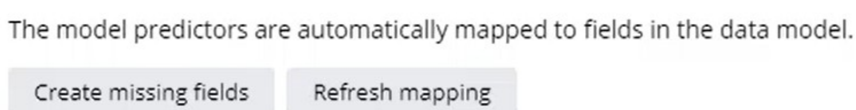
To leverage an existing model file, select the **Import model** option. Upload the PMML or H2O model file. The default context of the model is the **Customer class,** where the customer data model properties are stored. You can change this class if required.



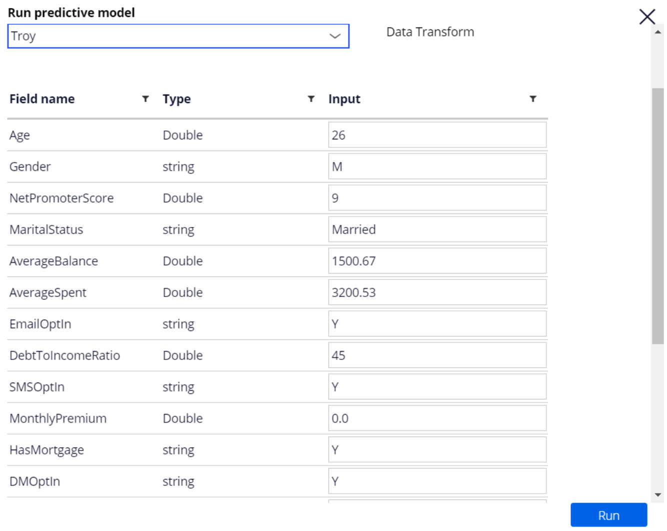
In the **Outcome definition** dialog box, you define which probability you would like to predict and the expected performance of the model, which is used as a benchmark when monitoring the model.



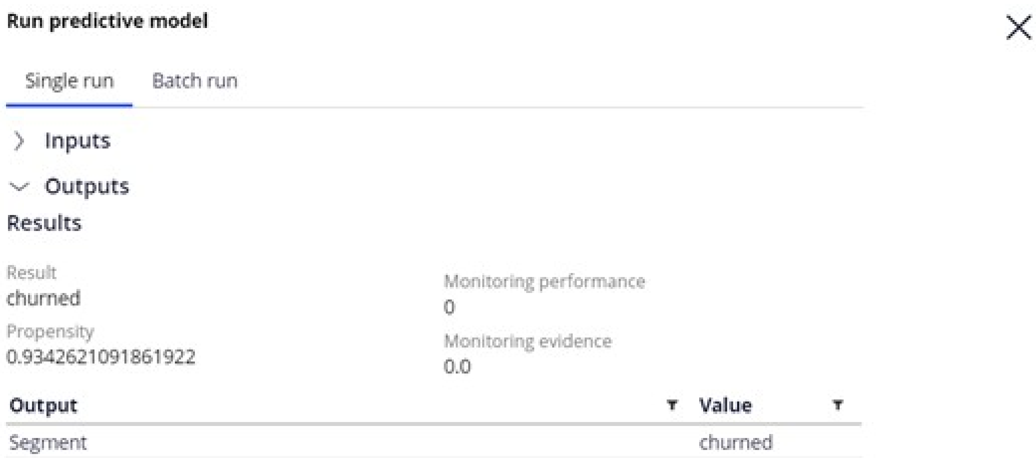
Import the model and, on the **Mapping** tab, make sure that all predictors are mapped to fields in the data model. Missing fields can be created, but this should be discussed with the system architect beforehand.



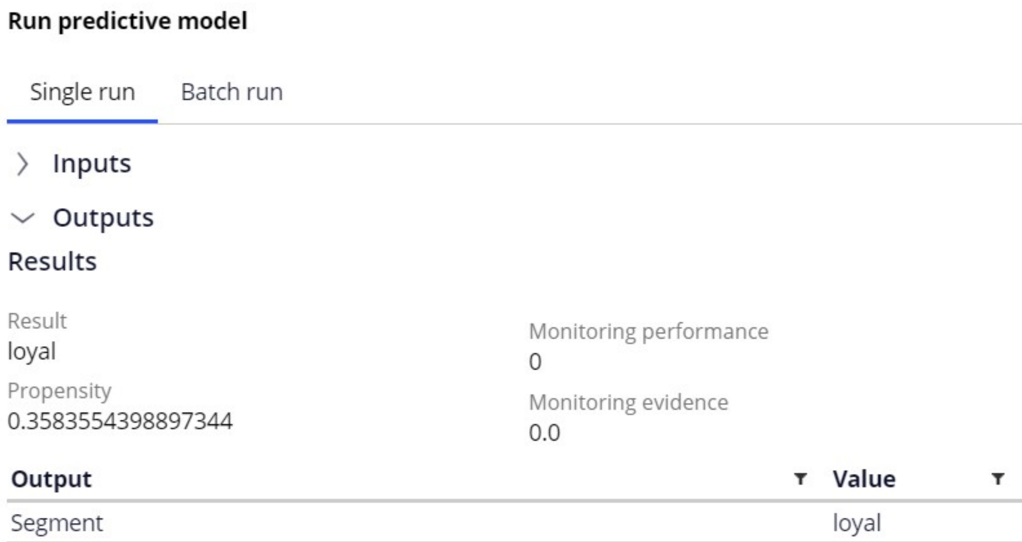
After the model is saved, you can test it for a single customer or run it for a batch of customers.



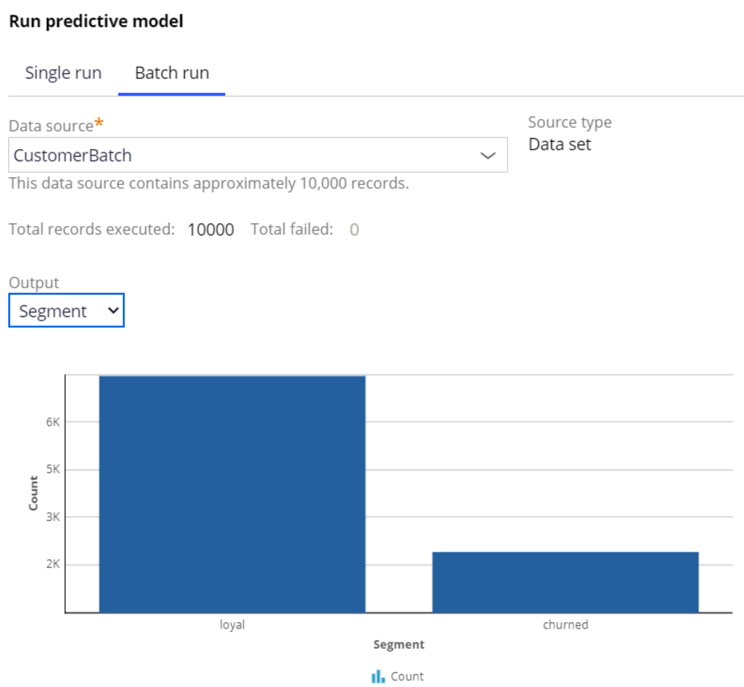
When you test the model for a single customer, you can use a data transform as input data. When customer Troy is used as the data source, the model predicts that he is likely to churn. The model also outputs his propensity to churn, which is, in this case, 93.42%.



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



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.



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.

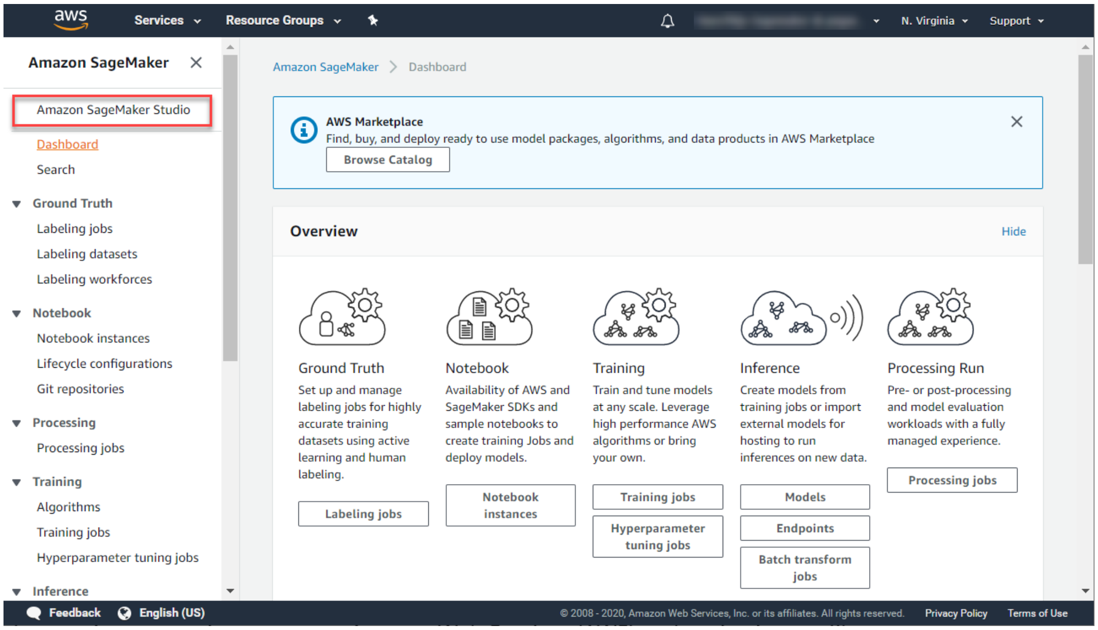
Transcript

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.

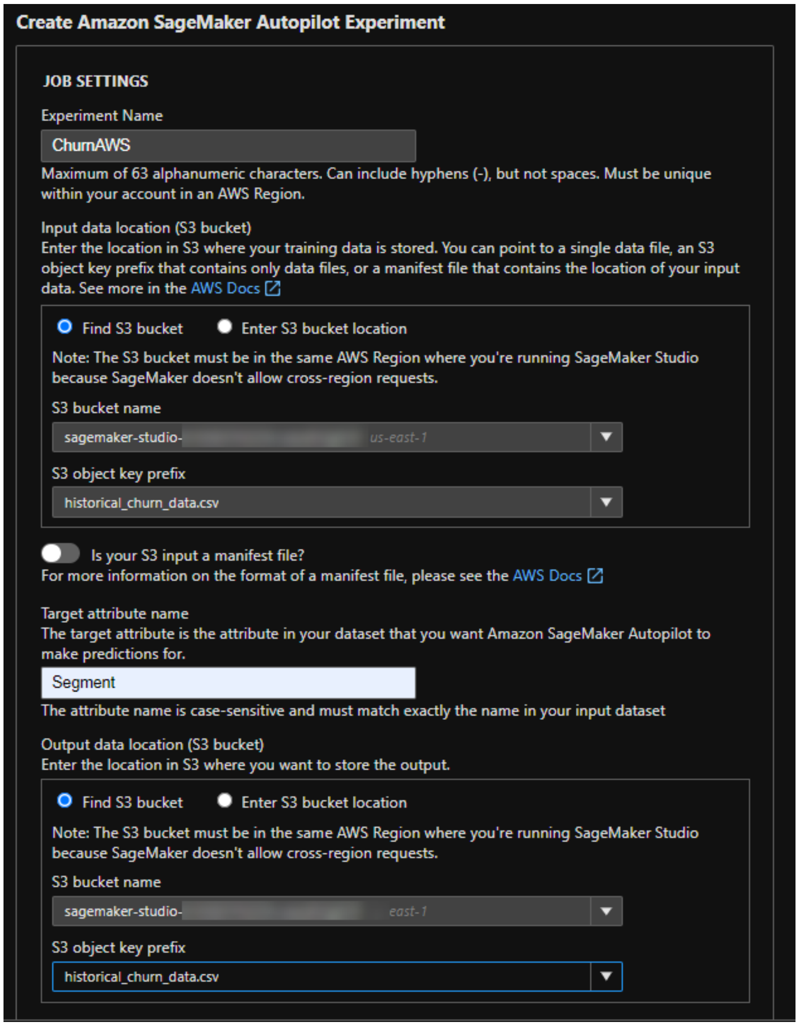
We will showcase this using Amazon SageMaker. The steps are similar to using other machine learning services such as Google AI Platform. Using a machine learning service instead of a model that runs locally may involve costs and possible down time of the service.

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.

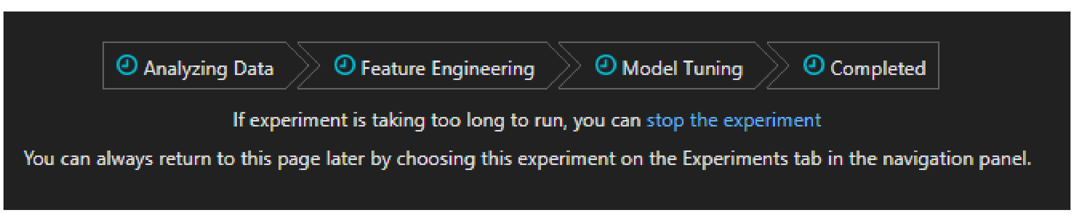
Amazon SageMaker allows you to build, train and deploy machine learning models in a fully managed service. The Autopilot feature automates this process and trains and tunes the best machine learning models for classification or regression, based on your data. After setting up your AWS environment, you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



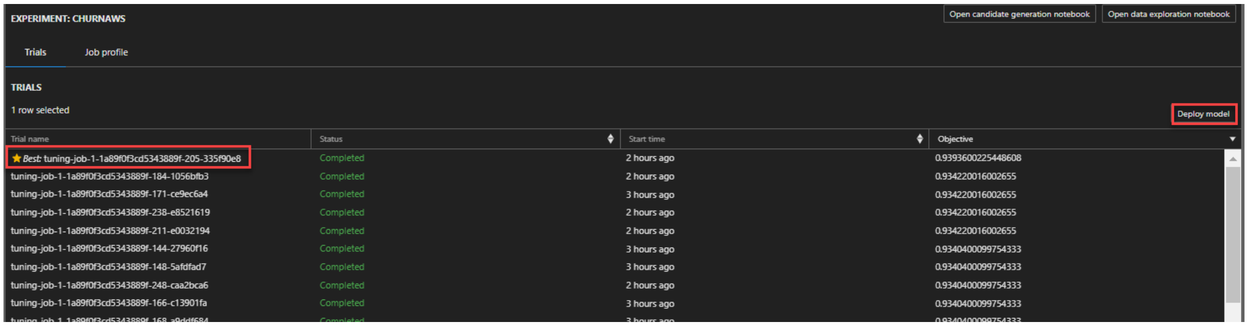
In the Job settings, select the data file you want to build the model on, specify the outcome field, choose the location where the output should be stored and create the experiment.



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

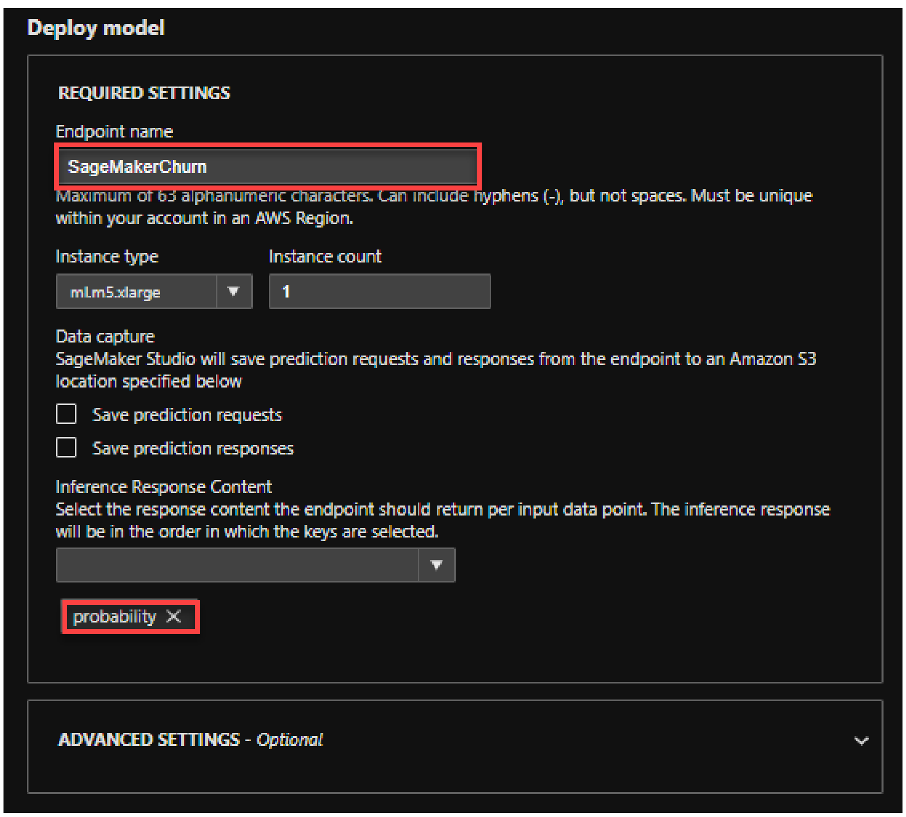


To deploy the best candidate model, select the tuning job with the highest Objective value. This value indicates the predictive power of the model.

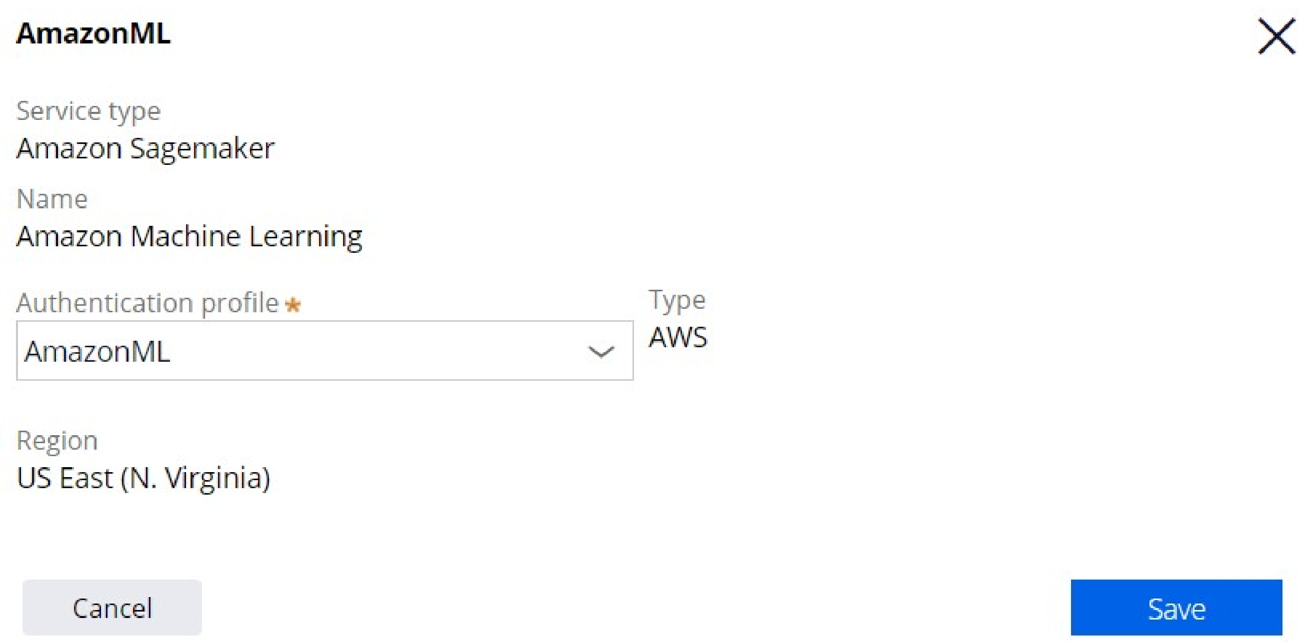


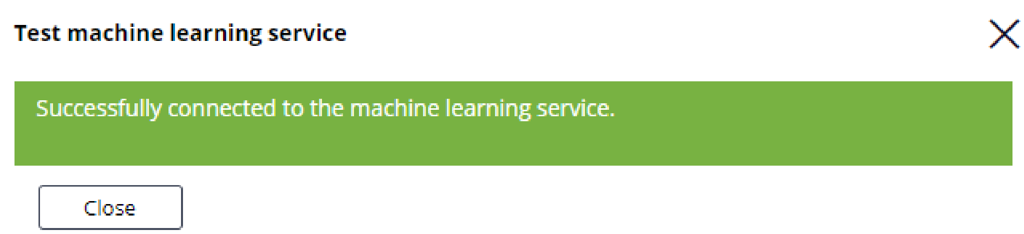
An endpoint that can be reached from Pega is automatically created. A binary classification, as in this example, predicts if an event will happen or not, based on a cut-off value. By default, the response content for a binary model is set to this ‘predicted_label’.

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.

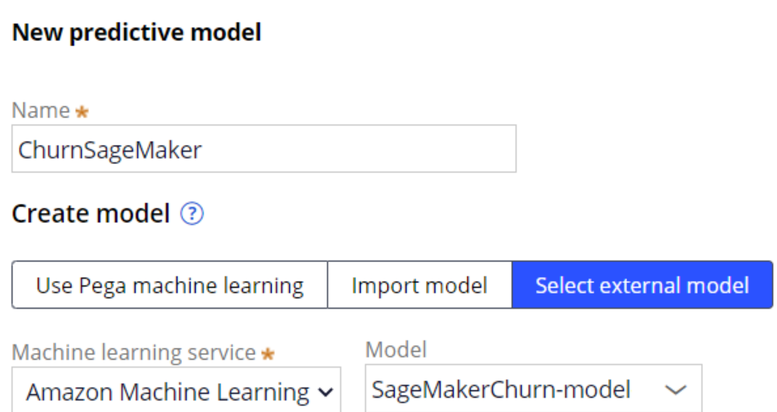


In Prediction Studio, you can define a machine learning service to connect to your cloud service instance. To move messages securely to and from Pega, the system architect has set up an authentication profile.





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.



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

ModelMetadataTemplate.json - Notepad

```
{
	"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"
			]
		}
	}
}
```

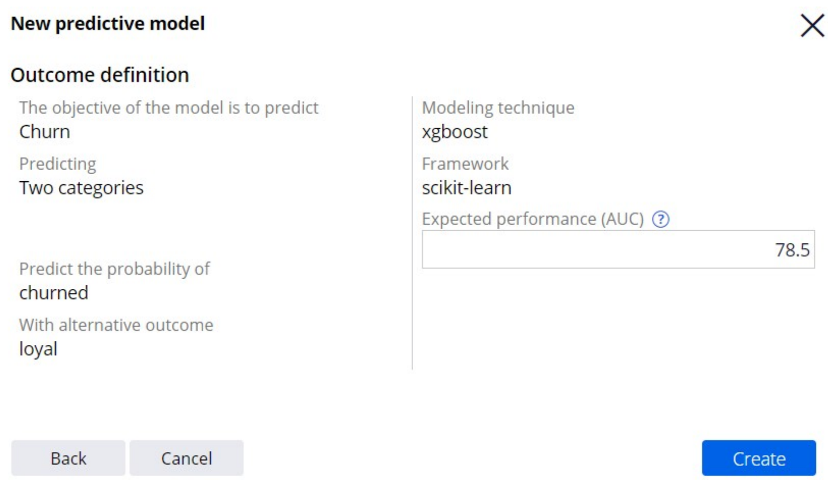
Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

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.

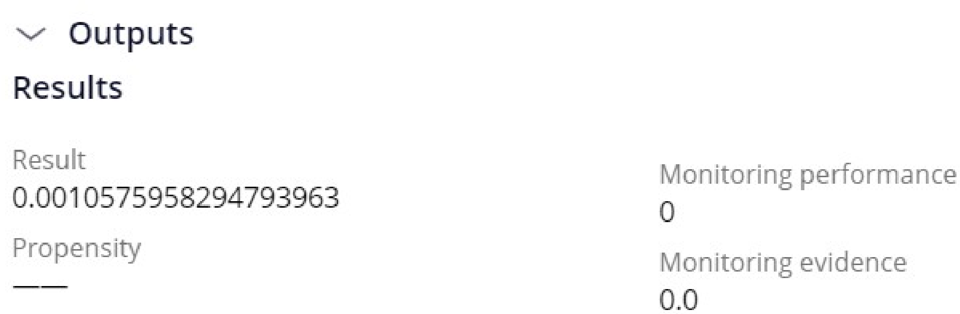
Next, set the correct context of the model if required. The default context is the customer class. You can review the model metadata, such as the objective of the model and the type of problem to solve, before proceeding.



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.

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



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?

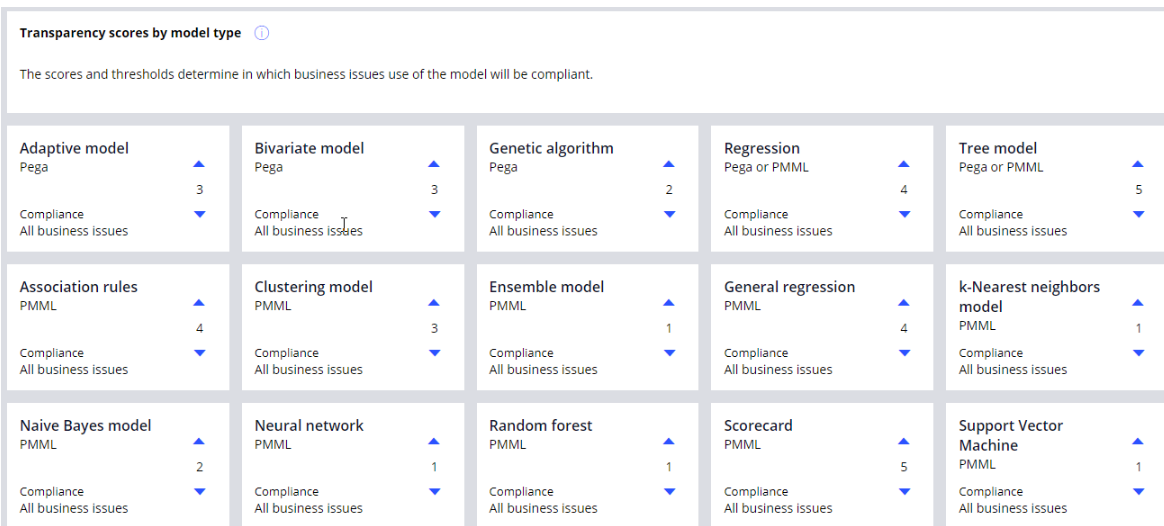
- The high level steps involved in creating a model using Amazon SageMaker Autopilot.
- How to connect to external machine learning services and run a model externally.

Model transparency

Transparency score

Transparent artificial intelligence is becoming an important requirement for many businesses. In risk management, decisions need to be explainable, and opaque predictive models are not allowed. In marketing, the policy for the transparency of models might be less strict and allow for the use of opaque models.

Each model type that comes with Pega Platform™ is assigned a transparency score ranging from 1 to 5, where 1 means that the model is opaque, and 5 means that the model is transparent. Highly transparent models are easy to explain, whereas opaque models might be more powerful but difficult or not possible to explain. For example, a decision tree has a high transparency score, whereas a neural network model has a low transparency score.



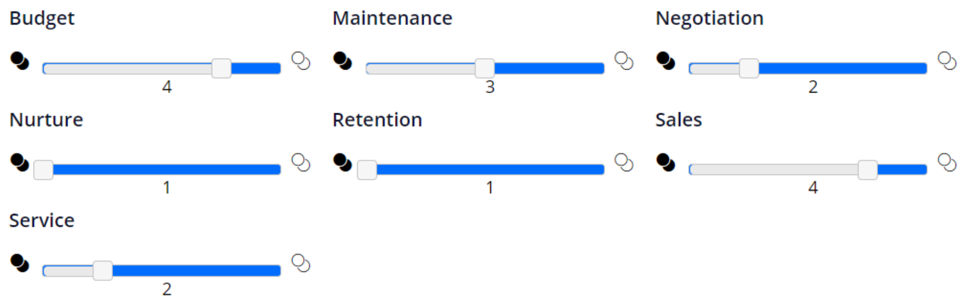
Configuring the model transparency policy

Depending on the company policy, models are marked as compliant or non-compliant for a specific business issue. By default, the transparency threshold is set to 1 and all model types are allowed in all business issues. Lead data scientists can modify transparency thresholds for different business issues. For example, they can increase the threshold for Budget to indicate that opaque models are non-compliant in that area. To set the thresholds in Prediction Studio, go to the Settings section and choose Model transparency policies.

Transparency threshold per business issue

Models with a transparency score above or equal to the threshold of business issues, are compliant. A high transparency score of 5 indicates that models are fully auditable.

Range of scores 1(Opaque) 5 (Transparent)



MLOps

Description

Learn how to use Machine Learning Operations (MLOps) to replace the predictive model that drives a prediction with a new model. You can import a predictive model that is built on an external platform or connect to a machine learning service. Deploy the model in shadow mode. In shadow mode, the candidate model ingests production data but does not affect decisioning until it is promoted to the active model status.

Learning objectives

- Deploy a new model in shadow mode
- Promote the new model to the active model status

MLOps process

Introduction

Learn how you can improve the performance of your predictions by using a standardized Machine Learning Operations process (MLOps). MLOps lets you replace a low-performing predictive model that drives a prediction with a superior model created in a third-party platform.

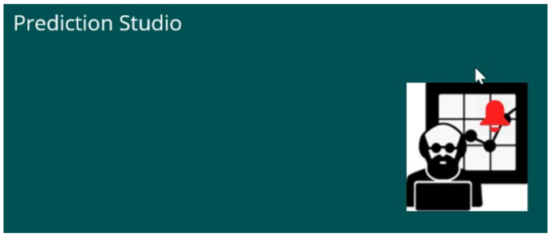
If a candidate predictive model is deployed in shadow mode, it can be monitored with real production data without impacting the business outcomes. If the model proves effective, it is deployed as the active model.

Transcript

This video shows you how to update a predictive model in a prediction.

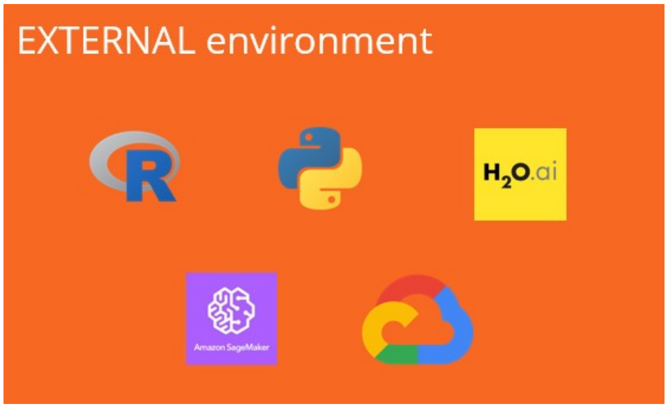
In the standardized Machine Learning Operations (MLOps) process, the active model is replaced with a better one in a production environment by using the shadow mode option. A prediction is driven by an adaptive model, a predictive model, a scorecard model, a field model, or a combination of these models.

You can replace a model in the production environment at any time through the MLOps process. As a data scientist, you may respond to a Prediction Studio notification that an active model does not generate enough lift, and decide to replace the low-performing model with a high-accuracy external model. Or you can update a prediction regularly, for example, whenever you develop a new model.

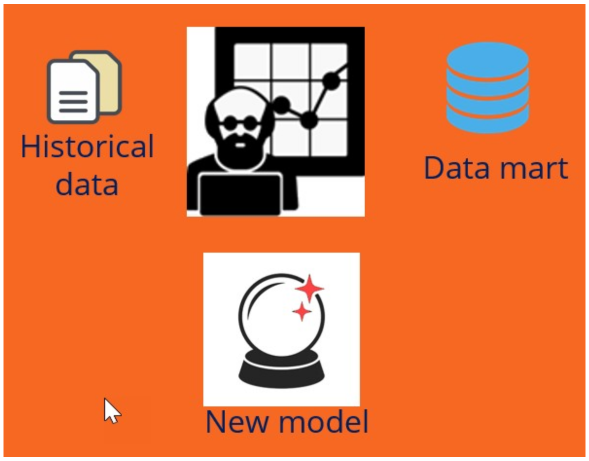


To build a new model, you can use Pega machine learning or an external environment. You can use data science tools that can export models in the PMML format, such as R or Python.

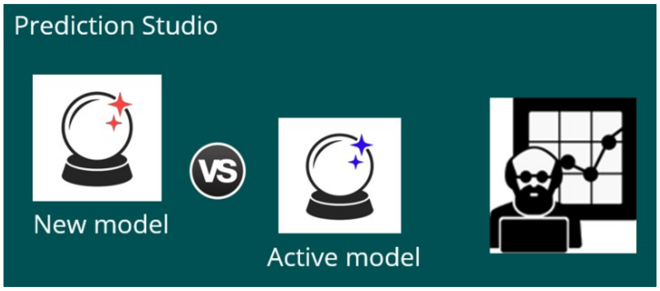
The H20 format is another option. You can also connect to the Amazon SageMaker or Google Cloud machine learning services.



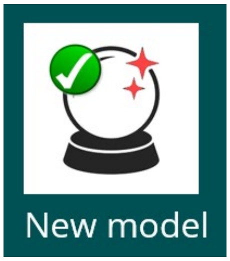
You can utilize the historical data of models captured by the system by importing these records into your external environment of choice. The historical data can be combined with data from other sources to build a new model.



Once the new predictive model is developed, you validate the active model and the candidate model against the same data set to compare their metrics in Prediction Studio.

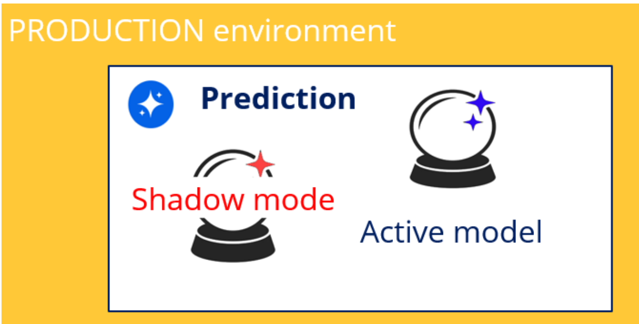


If the candidate model outperforms the active model, approve the model.



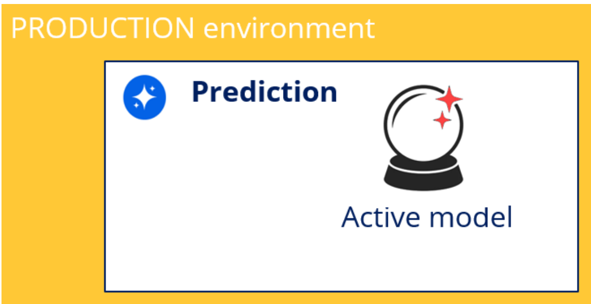
You can choose to replace the active model immediately or place the new model in shadow mode.

The new model is then promoted to the production environment in a revision.



If you deploy the new model in shadow mode, the new model is exposed to the production data but does not drive the prediction yet. Shadow mode allows you to monitor the model performance in a production environment before deploying it as an active model.

After monitoring the prediction for some time, you can promote the shadow model to active.



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

- How the model driving a prediction is updated with a new predictive model
- How the shadow mode allows monitoring of a new model in a production environment

Placing a predictive model in shadow mode

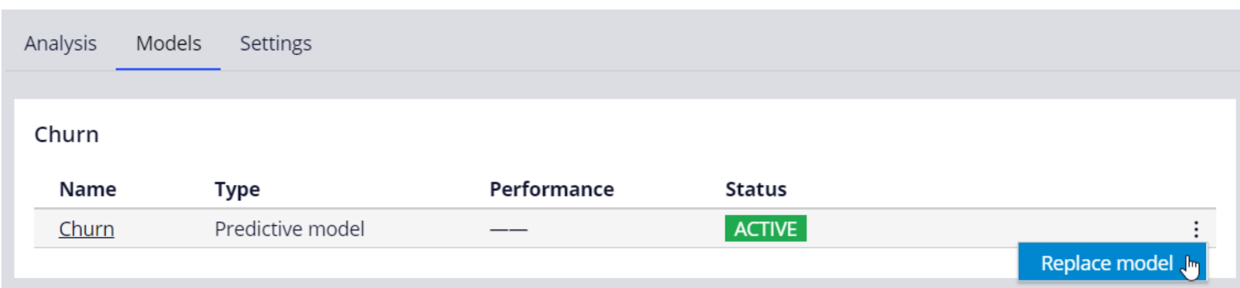
As a data scientist, you can approve changes to the models that drive predictions for deployment to the production environment. You can change models independently or respond to a Prediction Studio notification that a prediction does not generate enough lift.

To improve the performance of a prediction, you can replace a low-performing model with a high-accuracy external model that you upload to a Pega repository or directly to Prediction Studio. As a result, you start a standard approval and validation process to deploy the model update to production. Before you approve any changes, you can compare the candidate model with the existing model based on data science metrics, such as score distribution or lift.

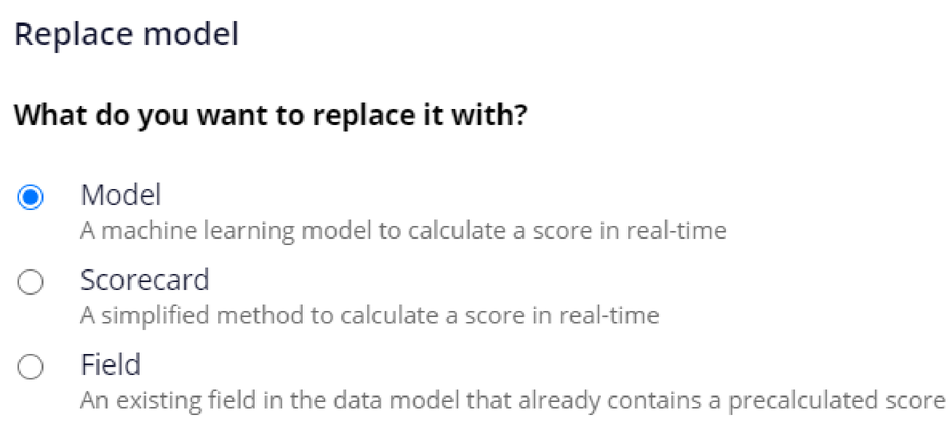
Model deployment

In Pega Customer Decision Hub™ environments, changes to models that you approve in Prediction Studio are deployed to production through Pega 1:1 Operations Manager and the Business Change pipeline.

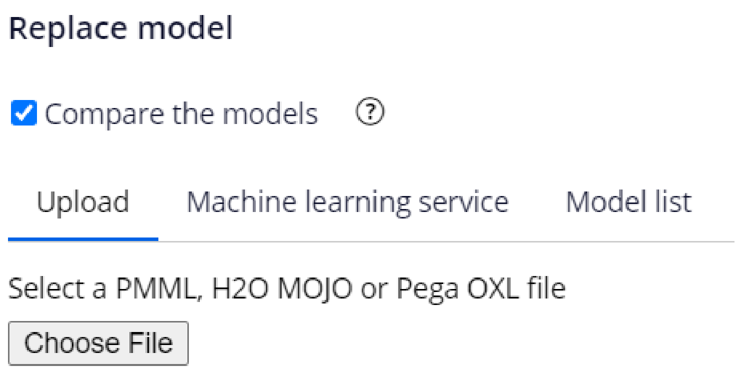
The process begins when you update a prediction in your non-production environment.



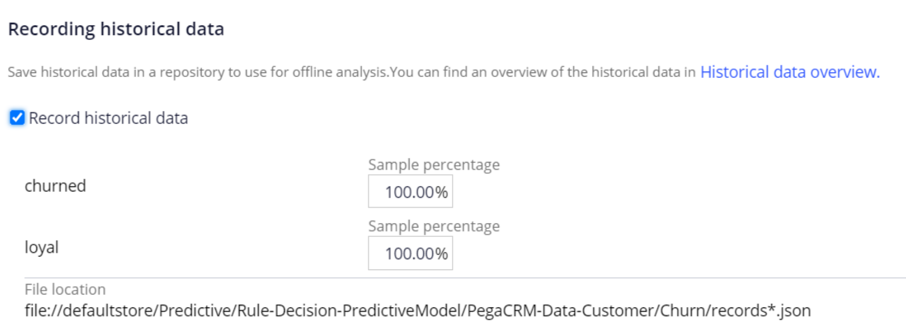
You can replace the active model with a predictive model, scorecard, or field in the data model that contains a score.



You can select a model from Prediction Studio, upload a PMML or H2O model to Prediction Studio, or connect to an external model that you developed on Amazon SageMaker or Google AI Platform.

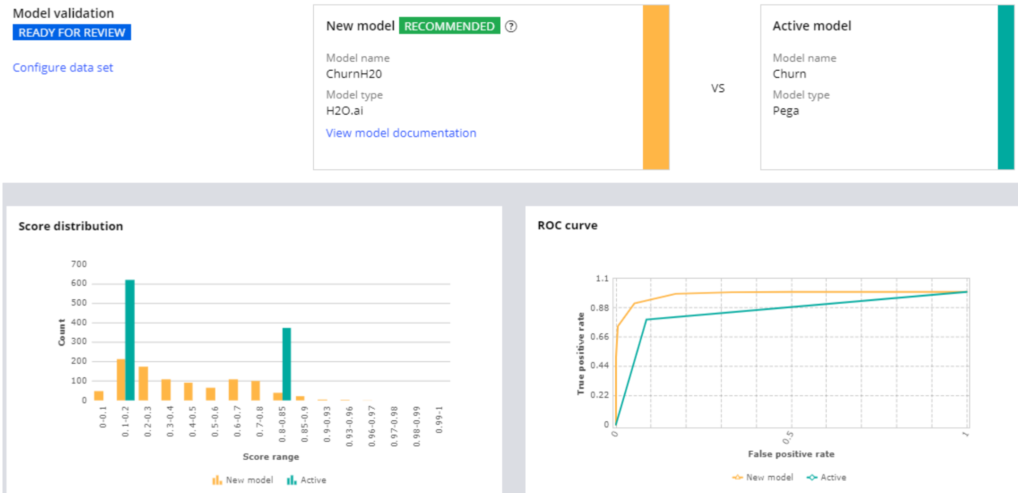


When creating a machine learning model in a third-party environment, you can use historical data from Pega Platform™ and other sources to train the model.



You can validate the candidate model against a data set and compare the new model with the current model.

This analysis provides relevant metrics to help you decide which model has better performance with a static data set.



After you evaluate the models, you can approve or reject the candidate model for deployment to production. You can place the model in shadow mode or immediately replace the current model with the new model.

Evaluate ChurnH20

Evaluate the model and provide your feedback. ?

Evaluation

- (•) Approve new candidate model and start shadowing (recommended)
- ( ) Approve candidate model and replace current active model
- ( ) Reject candidate model

In a Pega Customer Decision Hub environment, the system creates and resolves a change request in Pega 1:1 Operations Manager. A team lead can verify the changes in the rules and the relevant documentation. The change request is packaged into a revision, and a deployment manager can promote the prediction with the candidate model to production.

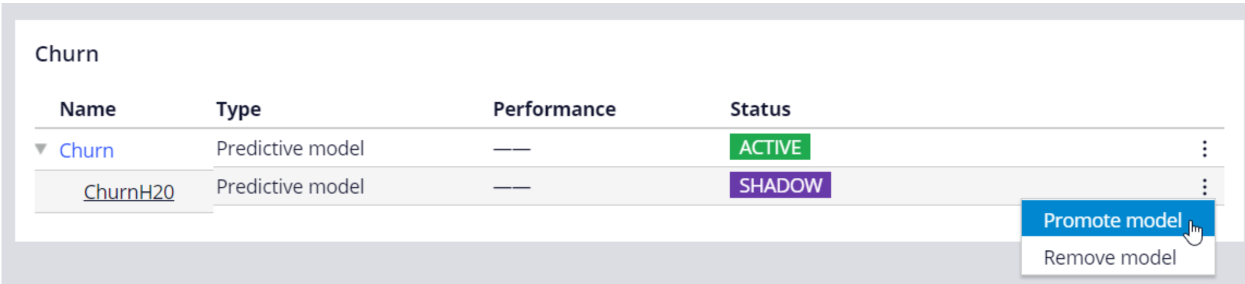
If you deploy the candidate model to production in shadow mode, it runs alongside the original model, receives production data, and generates outcomes, but the outcomes do not impact business decisions.

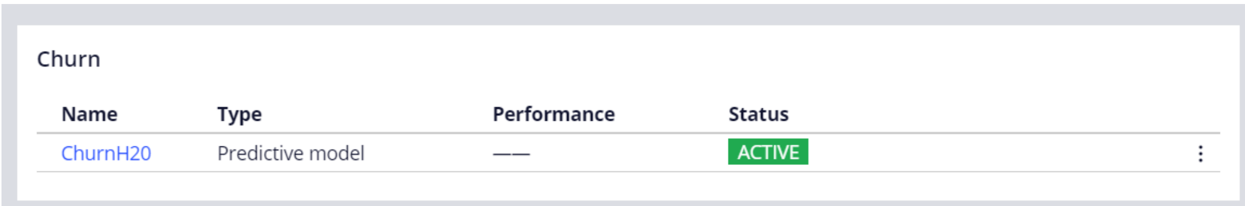
Churn

Name	Type	Performance	Status
Churn	Predictive model	——	ACTIVE
ChurnH20	Predictive model	——	SHADOW

Model promotion to active

If the model proves ineffective, you can reject it and add another model to the prediction. If the new model performs well in production, you can promote it to the active model position.





Adaptive analytics overview

Description

Pega Adaptive Decision Manager (ADM) is a component that allows you to build self-learning adaptive models that continuously improve predictions. ADM can automatically detect changes in behavior, which enables business processes and customer interactions to adapt to the changes in real time.

Learning objectives

- Identify potential predictors for adaptive models
- Describe the outcomes and advanced settings of an adaptive model

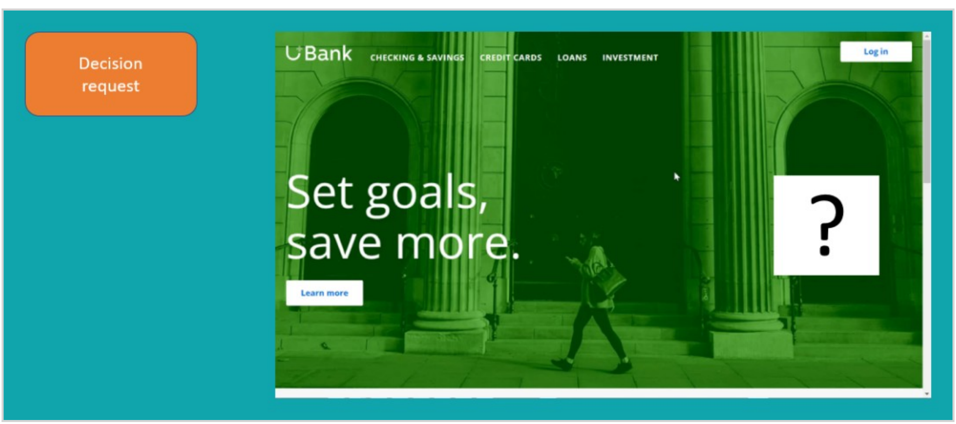
Adaptive analytics

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.

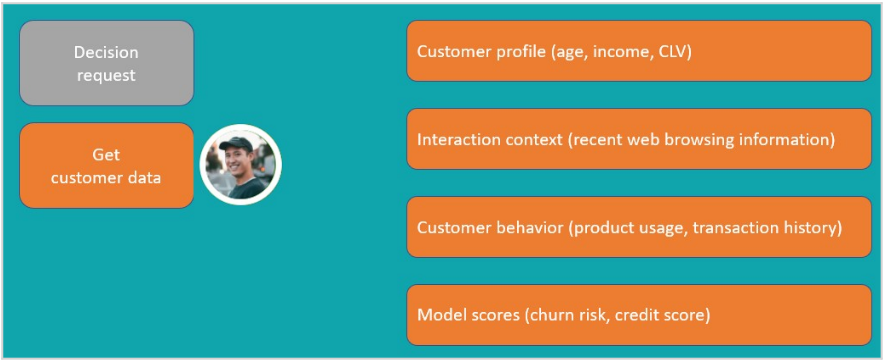
Transcript

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.

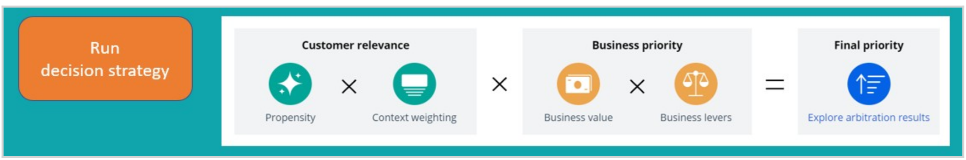
Adaptive Decision Manager (ADM) is a component of Pega Decision Management that businesses can use to implement online, adaptive models that drive predictions about customer behavior, like clicking or ignoring a web banner that offers a credit card on a bank’s website. When customer Troy logs in to the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



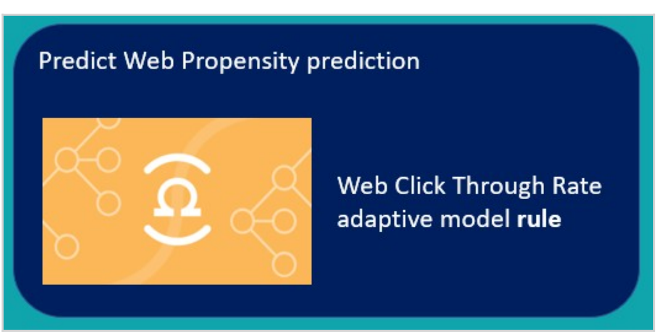
Using this data as input, the system runs a decision strategy to determine which credit card is the best offer for Troy, balancing customer relevance and business priority.



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.

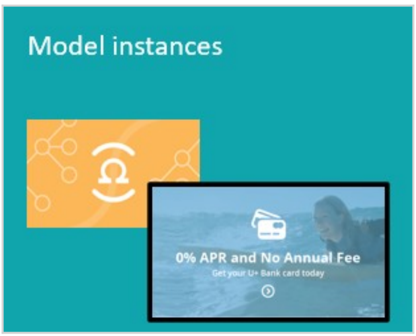
If Troy is interested and clicks on the web banner, ADM records the outcome as target behavior. If he ignores the banner, ADM records the outcome as alternative behavior. The prediction predicts the probability that a customer shows the target behavior.

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.

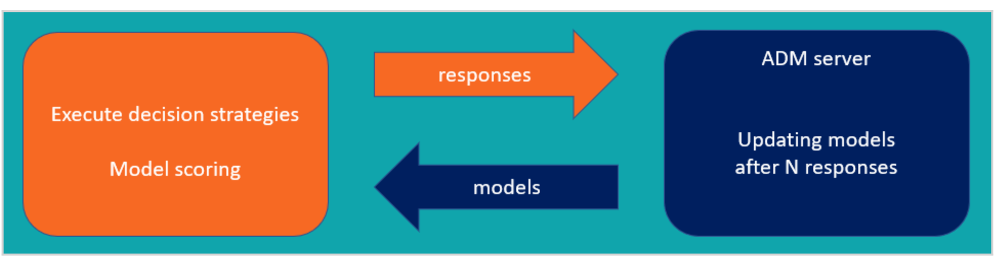


A data scientist configures the settings of both the prediction and the adaptive model rule, including the customer fields that are available as features to the model rule. Customer fields that are unsuitable as features, for example the customer ID, should be excluded.

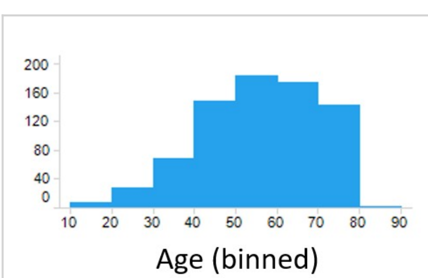
An adaptive model rule typically generates many adaptive model instances without human intervention, because each unique combination of an action, treatment, direction, and channel, will generate a model the first time a decision strategy runs that references the model.



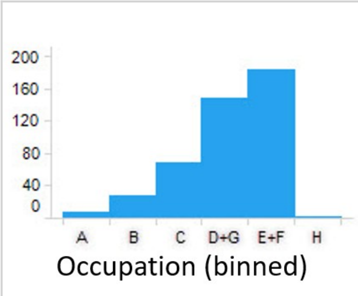
ADM captures the responses and updates the adaptive model instances regularly, so they continuously learn from customer responses and adapt to account for changing customer interests and needs. The ADM server is physically separated from the nodes that process decision strategies and model executions, 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.

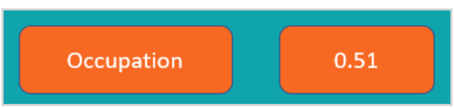


For symbolic predictors, ADM groups values with similar customer behavior.

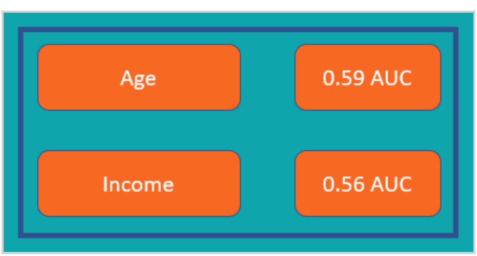


The granularity of the binning is a trade-off between performance and the statistical robustness of the predictor. Appropriate default settings for binning are provided and should only be changed by an experienced data scientist for specific use cases.

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.

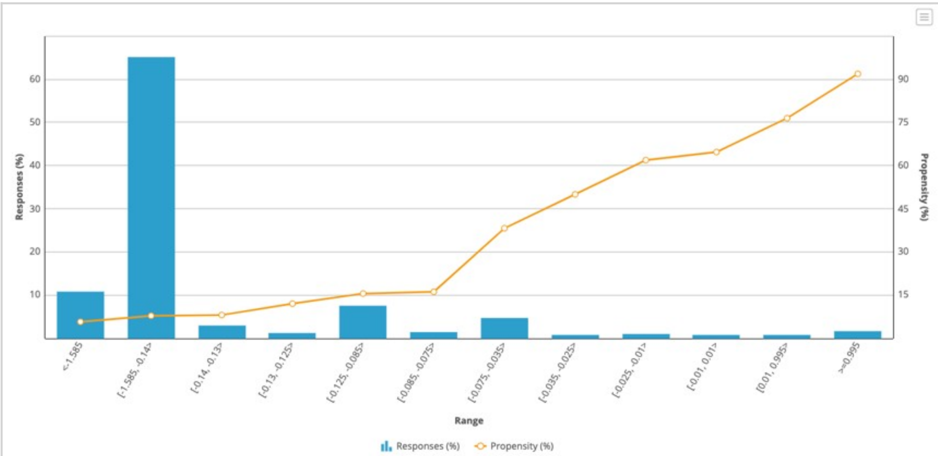
This simple and scalable calculation is based on Bayes' theorem, which says that the probability of A, if B is true is equal to the probability of B, if A is true, times the probability of A being true, divided by the probability of B being true.

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

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.

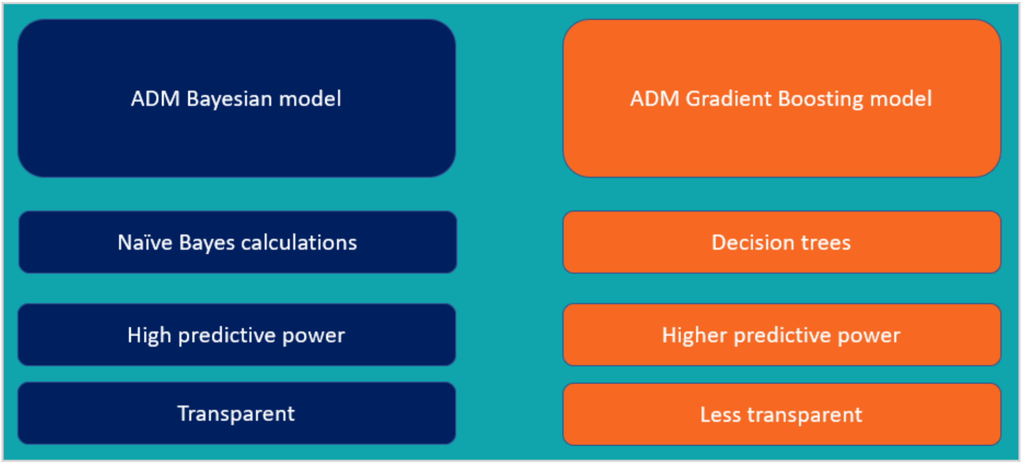
Score (Log odds) = Ln(P/(1-P))

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.



When you create a new adaptive model, you have a choice between either an ADM Bayesian adaptive model rule based on Naïve Bayes calculations, or building an ADM gradient boosting, or AGB, model rule, which is based on decision trees. In contrast to ADM Bayesian model instances that are focused on a single combination of action, treatment, direction, and channel, an AGB model instance learns from all combinations of actions and contexts.

AGB models can achieve higher predictive power to deliver more accurate predictions, which leads to higher success rates, retention, and lifetime value. However, as AGB model are more complex, they come with an important trade-off between accuracy versus transparency. So, your choice will depend largely on the transparency requirements of your use case.



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.
- How AGB models can deliver more accurate predictions.
- How the choice of a Bayesian model or an AGB model depends on the transparency requirements in a specific use case.

Predictors and outcomes of an adaptive model

Predictors

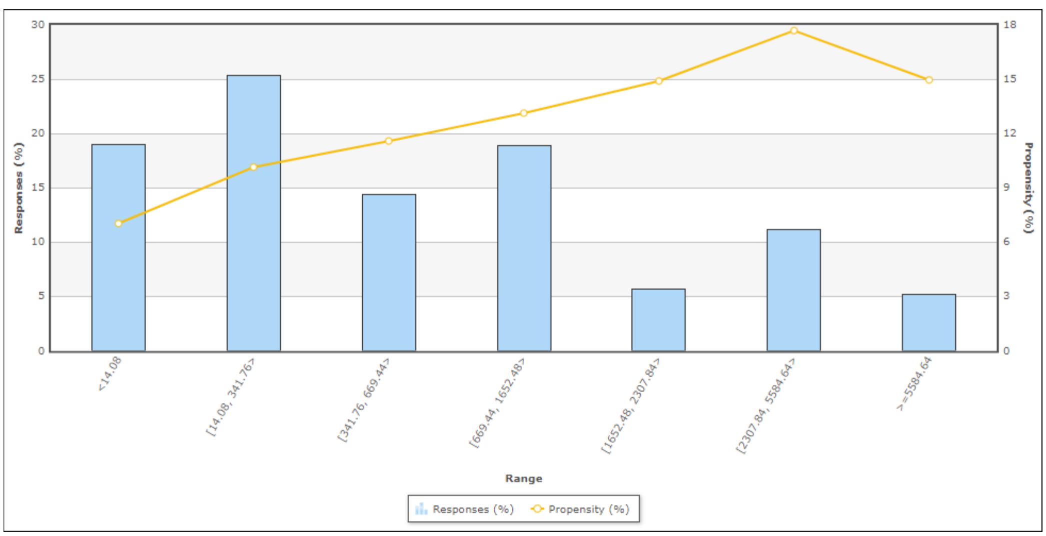
Configuring adaptive models involves selecting potential predictors and setting outcomes that identify positive and negative customer behavior. Unless you are a highly experienced data scientist, it is strongly recommended to leave the advanced settings at their default

The input fields you select as predictor data for an adaptive model play a crucial role in the predictive performance of that model. A model's predictive power is at its highest when you include as much relevant, yet uncorrelated, information as possible. In Pega, it is possible to make a wide set of candidate predictors available, as many as several hundred or more.

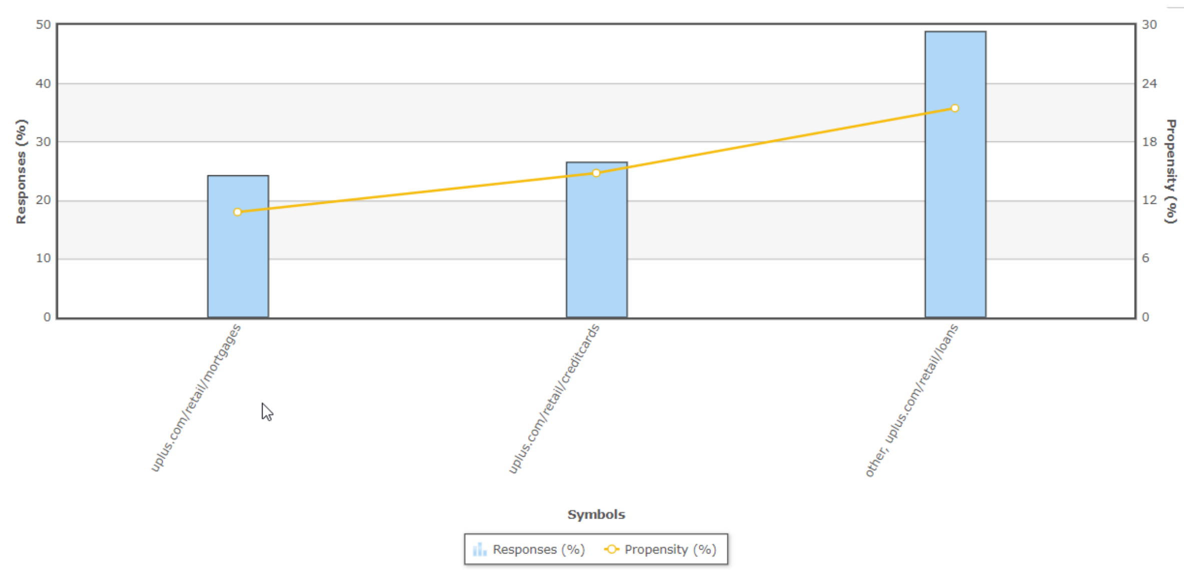
Adaptive Decision Manager (ADM) automatically selects the best subset of predictors. ADM groups predictors into sets of correlated predictors and then selects the best predictor from each group, that is, the predictor that has the strongest relationship to the outcome. In adaptive decisioning, this predictor selection process repeats periodically.

You can use several data types in adaptive analytics, including:

Numeric data - Basic numeric data such as age, income, and customer lifetime value can be used without any preprocessing. Your model automatically divides that data into relevant value ranges by dynamically defining the bin boundaries.



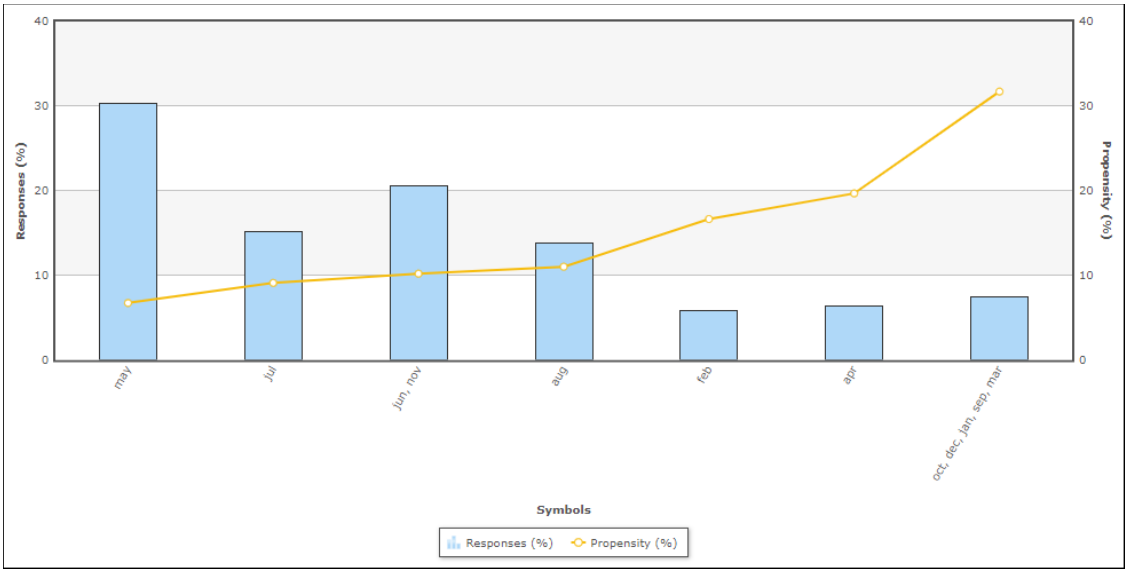
Symbolic data - You can feed predictors with up to 200 distinct string values without any preprocessing. Such data is automatically categorized into relevant value groups, such as the **PreviousWebpage** predictor in the following example. For predictors with more than 200 distinct values, group the data into fewer categories for better model performance.



Customer identifiers - Customer identifiers are symbolic or numeric variables that have a unique value for each customer. Typically, they are not useful as predictors, although they might be predictive in special cases. For example, customer identifiers that are handed out sequentially might be predictive in a churn model, as they correlate to tenure.

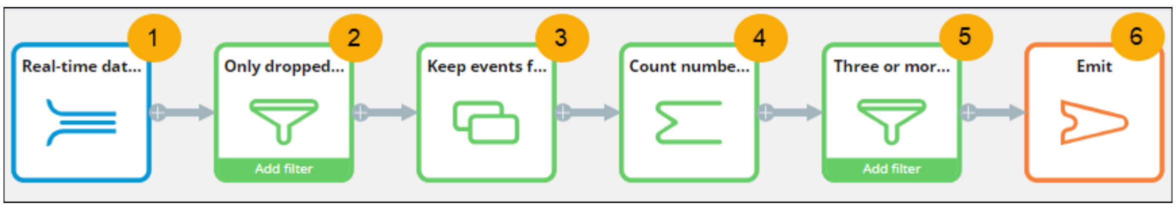
Codes - For meaningful numeric fields, feed code fragments to the model as separate predictors. Simple values require only basic transformation. For example, you can shorten postal codes to the first 2 or 3 characters which, in most countries, denote geographical location.

Dates - Avoid using absolute date/time values as predictors. Instead, take the time span until now (for example, derive age from the DateOfBirth field), or the time difference between various pairs of dates in your data fields (such as the DurationLastSubscription field). Additionally, you can improve predictor performance by extracting fields that denote a specific time of day, week, or month.



Text - Do not use plain text to create predictors without any preprocessing; it contains too many unique values. Instead, extract values such as intent, topic and sentiment to use as predictors. Pega features a Text Analyzer rule for this purpose.

Event streams - Do not use event streams as predictors without preprocessing, aggregate the data instead. Pega features event strategies for this purpose. As an example, this event strategy detects dropped calls.



First, (1) it listens to a real-time dataset; then (2) it filters out dropped customer calls; next (3) it stores the terminated calls for one day; (4) it counts the number of terminated calls within the one-day timeframe; and (5) it creates an event if three calls are terminated within the one-day timeframe; lastly, (6) it emits the event. The aggregates can be stored and used like any other symbolic or numeric field.

Interaction History - Past interactions are usually very predictive. You can use the Interaction History (IH) to extract fields such as the number of recent purchases, the time since last purchase, and so on. To summarize and preprocess IH data for predictions, use IH summaries. Several predictors based on IH summaries are enabled by default (and require no additional setup) for all new adaptive models. These are the group that was referenced in the last interaction, the number of days since the last interaction, and the total number of interactions.

Multidimensional data - For models that inform the initial customer decision, things such as lists of products, activities, and transaction outcomes are useful sources of information for predictors. Use your intuition and data science insight to determine the possibly relevant derivatives, for example, number-of-products, average-sentiment-last-30-days, and so on.

Interaction context - To increase the efficiency and performance of your models, do not limit the data to customer data alone. By supplementing decision process data with the interaction context, you can adjust the predictions for a customer and provide different outcomes depending on their context. Contextual data might include the reason for a call, or the way the customer uses the website or mobile app to interact with the company, etc.

Customer behavior and usage - Customer behavior and interactions, such as financial transactions, claims, calls, and complaints, are typically transactional in nature. From an adaptive analytics perspective, you can use that data to create derived fields that summarize or aggregate this data for better predictions. Examples of this type of data include average length of a call, average gigabyte usage last month, and the increase or decrease in usage over the last month compared to previous months.

Model scores - Scores from predictive models for different but related outcomes as well as other data science output might be predictive as well. If you decide to use scores as predictors in your models, evaluate whether the models that include such a score perform better at the model level by verifying the area under the curve (AUC) and success rate metrics.

Summary

In summary, to achieve the best results, use predictors that provide data from many different sources, including:

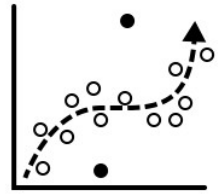
Customer profile data such as age, income, gender, and current product subscriptions. This information is usually part of the Customer Analytic Record (CAR) and is refreshed regularly.

Interaction context data such as recent web browsing information, call reasons, or input that is gathered during a conversation with the customer. This information can be highly relevant and, therefore, very predictive.

Customer behavior data such as product usage or transaction history. The strongest predictors of future behavior typically contain data about past behavior.

Model scores, which are scores derived from the off-line execution of external models.





Outcomes

The responses that indicate positive or negative behavior must be identified. When predicting the click-through rate for a web banner, the default value for positive behavior is **Clicked** and the default value for negative behavior is **NoResponse**.

Applications may use different words to identify positive or negative behavior, for example, **Accepted** may be identified as positive behavior and **Rejected** may be identified as negative behavior. You can add these values when needed.

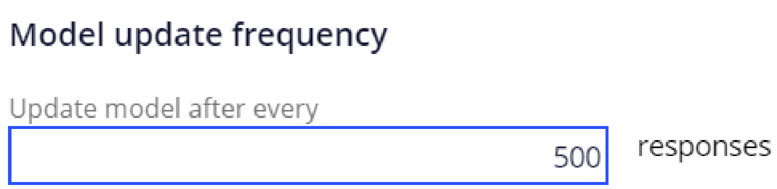
Advanced settings of an adaptive model

Default values

The default values for the adaptive model advanced settings are based on best practices and should only be changed by a highly experienced data scientist.

Update frequency and scope

When a model is updated, Prediction Studio re-trains the model with a specified number of responses. You can set the number of responses that will trigger the update.



You can also set the scope of the update. By default, all responses received during each update cycle are used. If you want to assign more weight to recent responses when updating a model, use a subset of the responses.

By default, all historical data is used to monitor the performance of the model. If required, model performance can be monitored for the most recent responses.

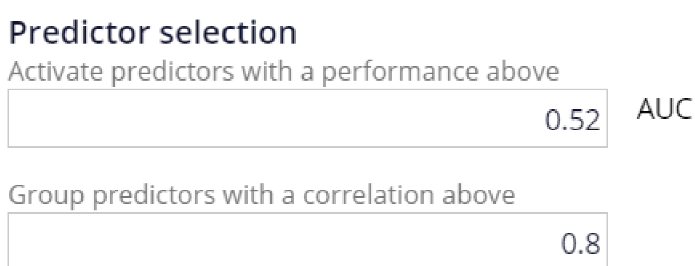
Grouping

The default values for **Grouping granularity** (the granularity of predictor binning) and **Grouping minimum cases** (the minimum percentage of cases per interval) are based on best practices and should not be changed casually.

The higher the value for **Grouping granularity**, the more bins are created. This value represents a statistical threshold that indicates when predictor bins with similar behavior are merged.

The **Grouping minimum cases** setting controls how predictor grouping is established. Higher values result in a decreasing number of groups, which can be used to increase the robustness of the model. Lower values result in an increasing number of groups, which can be used to increase the performance of the model.

The selection of the *active* predictors is guided by thresholds for predictor performance and the correlation between predictors.



The performance of a predictor is measured as the area under the curve (AUC). A higher value results in fewer predictors in the final model. The minimum AUC value is 0.5, therefore the value of the performance threshold should always be set to at least 0.5.

The value for the correlation between predictors determines when predictors are considered similar, and only the best of those predictors are used for adaptive learning. The measure is the correlation between the probabilities of positive behavior within pairs of predictors.

Adaptive model outputs

Model outputs

Adaptive models produce four outputs: Propensity, Evidence, Performance, and Positives.

Propensity is the predicted likelihood of positive behavior, for example, the likelihood of a customer accepting an offer. The propensity for every action starts at 0.5 or 50% (the same as a flip of a coin) because in the beginning, the model has no response behavior on which to base its predictions.

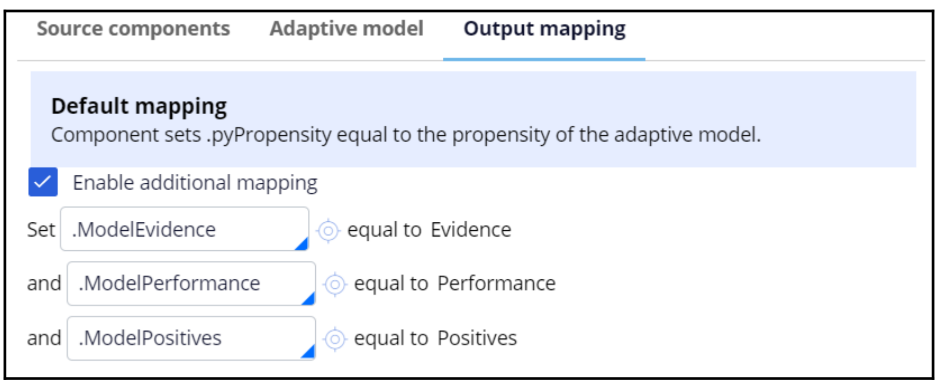
Evidence is the number of responses used in the calculation of the Propensity.

Performance is how well the model can differentiate between positive and negative behavior. Again, the initial value is 50%, with 100% being perfect performance. As a result, the performance value is somewhere between 50 and 100.

Positives is the number of positive outcomes that has been received by the model.

Mapping

In strategies, model propensity is automatically mapped to the strategy property called *.pyPropensity*. There is no automatic mapping for the Evidence, Performance or Positives outputs, but a strategy designer can manually map the outputs to any of the strategy properties under the **Output mapping** tab.



Optimizing AI in the NBA framework

Description

Learn how to improve the predictive power of your adaptive models by configuring additional potential predictors. 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.

Learning objectives

- Configure additional potential predictors for an adaptive model
- Configure parameterized predictors

Configuring an adaptive model

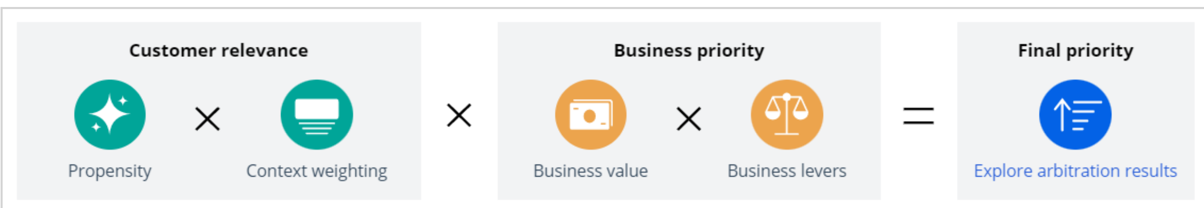
Explore how AI-based arbitration works for the website that U+Bank uses as a marketing channel to recommend more relevant banner ads to customers when they visit their personal portal.

As a data scientist, you can optimize the prediction that drives the decision of which banner to show to a customer by adding clickstream summaries that extend the customer profile with real-time behavioral data as potentially relevant predictors.

Transcript

This demo explores how AI-based arbitration works and shows you how to configure additional potential predictors for an adaptive model.

U+ Bank implements 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. The arbitration settings are defined in the Next-Best-Action Designer of Customer Decision Hub.



Arbitration aims to balance customer relevance with business priorities to decide which offer to show to the customer. To achieve this balance, numerical values that represent propensity, context weighting, business value, and business levers are multiplied to arrive at a prioritization value, which determines the top actions.

Propensity is the predicted likelihood of a customer showing the target behavior, in this case, clicking a web banner. The **Predict Web Propensity** prediction calculates the propensity. The **Web_Click_Through_Rate** adaptive model drives this prediction. The model calculates the propensity for each credit card offer for which a customer is eligible.

So, let’s see what is the next best action for 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.

Make Next-Best-Action decisions for the customer in the specified channel. The results are determined by enagagement policies, constraints and propensity.

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

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

	Name	Model evidence	Original model propensity	Final propensity	Rank	Results
	RewardsCard	0	0.5	0.9289	1	PASSED
	StandardCard	0	0.5	0.0511	2	PASSED

When the first request for a decision comes in through the TopOffers service, Customer Decision Hub creates an adaptive model instance for each of these offers. When created, the model evidence is zero as no responses have been captured yet. With zero evidence, the original model propensity is 0.5, or the flip of a coin.

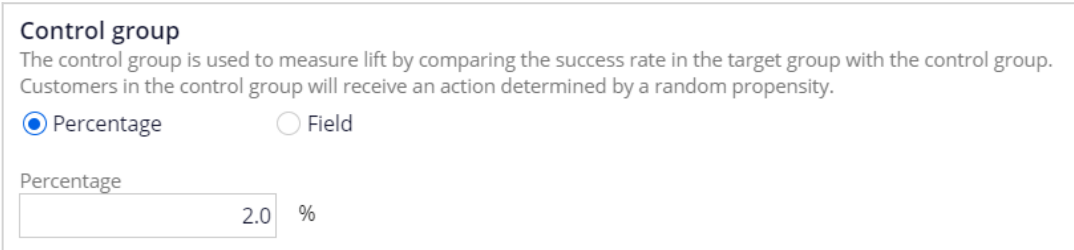
The final propensity that is used in the prioritization formula deviates from the original model propensity as it depends not only on the original model propensity but also on a mechanism that introduces noise while the evidence is low. The noise decreases as the model learns from the target and alternative 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.

Initially, an action is presented to only 2% of the population to minimize the possibility of customers receiving irrelevant offers. This percentage increases during the maturation of the model to 100%. A model is considered mature after it has received at least 200 positive responses.

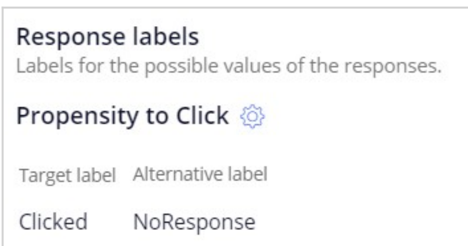
The Predict Web Propensity is the Customer Decision Hub prediction that calculates the final propensities for each combination of action and treatment in the inbound web channel.

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

The adaptive model calculates propensities for most customers, but a small subset of customers is in the control group. For this group of customers, the prediction generates random propensities.

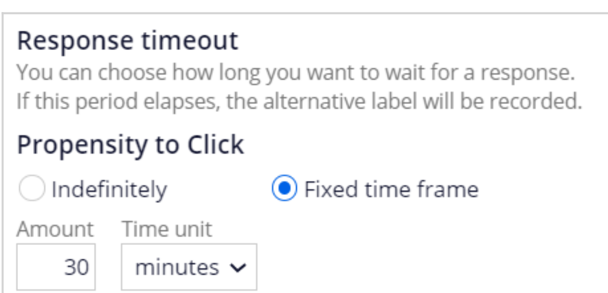


By comparing the success rate of model-based actions to random actions, you can see the impact of AI. The ratio of the success rate in the majority group and in the control group is called lift, and lift is an important KPI. The use of a control group also enables the models to explore alternative offers and remain flexible. The target response has a **Clicked** label by default. For the alternative response, the label is **NoResponse**.

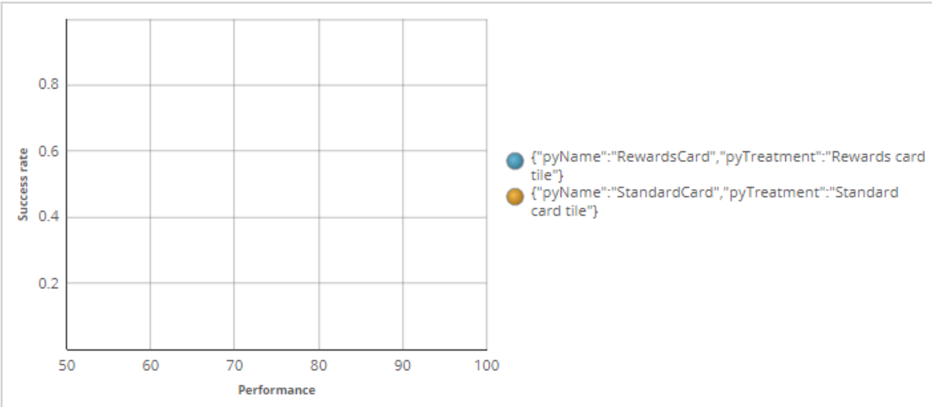


You can use additional response labels when an outcome has multiple possible labels. For example, when a channel passes multiple labels for the same outcome.

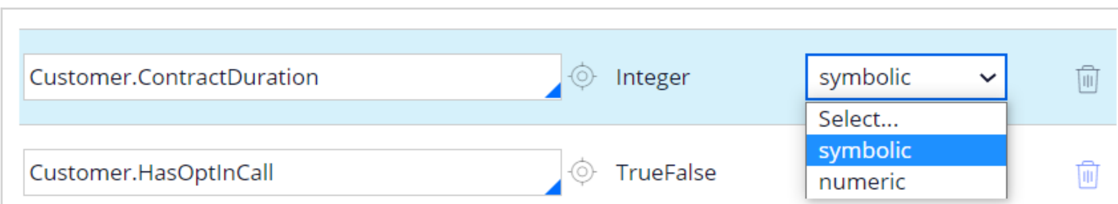
The **Response timeout** setting determines how long the system waits for a response from the customer after the impression. In a web scenario, the response timeout is 30 minutes by default, but an outbound channel requires a response timeout of several days to provide customers with enough time to respond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



The *Web_Click_Through_Rate* adaptive model drives the prediction. The decision request for Troy in Customer Profile Viewer prompts the creation of adaptive model instances for the combination of the Standard Card and Rewards Card offers and their inbound web treatments.



An adaptive model instance is a self-learning, online predictive model that uses machine learning to calculate propensity scores.It automatically determines the data fields that help to predict customer behavior. Predictors can be one of two types: numeric or symbolic. The system uses the property type as the default predictor type during the initial set-up, but you can change the predictor type. For example, when you know a numeric predictor has a small number of distinct values, such as when the contract duration is either 12 or 24 months, change the predictor type from **numeric** to **symbolic**.



Keep in mind that changing the predictor type effectively means removing and adding a predictor. A best practice is to make these changes early in the process, as there is no way to retain previous responses.

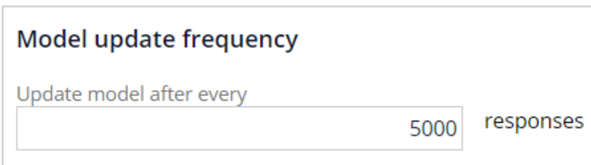
As a data scientist, you can enhance the model by adding additional fields. It is highly recommended to add many uncorrelated predictors, as the models figure out which ones to use. Additional predictors can include customer behavior, contextual information, past interactions with the bank, and even scores from external models. The FSClickstream page represents customer behavioral data that the system architect recently introduced.

The Interaction History (IH) dataset captures the customer responses. Aggregated fields from IH summaries are automatically provided to the models as predictors. IH summaries leverage historical customer interactions to improve the predictions.

Predictor	Aggregate	Field from interaction history
IH.{Channel}.{Direction}.{Outcome}.pxLastGroupID	Last	pyGroup
IH.{Channel}.{Direction}.{Outcome}.pxLastOutcomeTime.DaysSince	Last	pxOutcomeTime
IH.{Channel}.{Direction}.{Outcome}.pyHistoricalOutcomeCount	Count	

An example of a predictor is the group of the most recently accepted offer in the call center with the naming convention *IH.CallCenter.Inbound.Accepted.LastGroup*.

The model update frequency determines the number of responses that triggers an update of the adaptive model instance.



As a best practice, configure the **Model update frequency** so that model instances update every 2-4 hours on average. The default setting is 5000, which is suitable for a web banner with around 40 impressions per minute. Additionally, a model update occurs at least every 12 hours to ensure all recorded responses are regularly processed.

You can save historical customer responses to the offers for offline analysis in a repository.

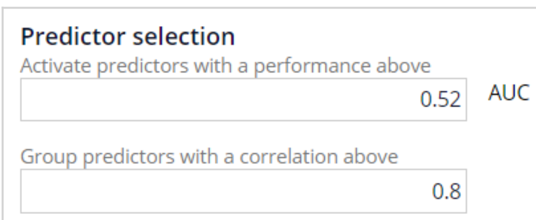
The default values for the advanced settings in an adaptive model are based on best practices. Only a highly experienced data scientist should change the default values. By default, the system uses all received responses for each update cycle, which suits most use cases. The option to use a subset of responses assigns additional weight to recent responses and increasingly less weight to older responses when updating a model.

The **Monitor performance for the last** field determines the number of weighted responses the model performance calculation uses for monitoring purposes. The default setting is 0, which means that the calculation uses all historical data.

Additional parameters determine the binning of the responses.

The **Grouping granularity** field determines the granularity of the predictor binning. A higher value results in more bins. The **Grouping minimum cases** field determines the minimum fraction of cases for each interval. The default setting is 5% of the cases. Together, these two settings control the grouping of predictors by influencing the number of bins. A higher number of bins might increase the performance of the model, but the model might also become less robust.

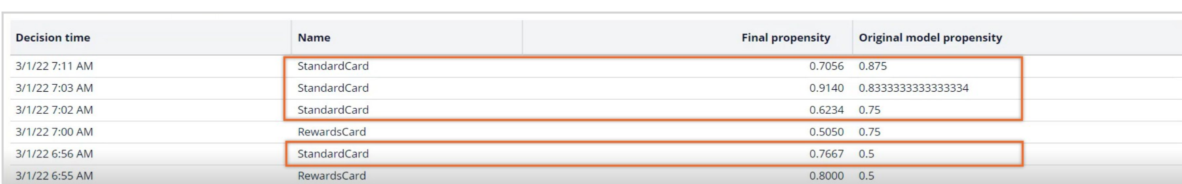
The system activates predictors that perform above a threshold. Over time, the system dynamically activates or deactivates the predictors when they cross the threshold.



Area Under the Curve (AUC) is a measure of the model performance of the predictor. It tells how well the predictor can distinguish between classes. The minimum AUC value is 0.5, so the value of the performance threshold should always be above 0.5.

The system considers pairs of predictors with a mutual correlation above a threshold as similar, groups them, and uses only the best predictor in a group for adaptive learning.

To test the adaptive learning on target behavior, log into the U+Bank website as Troy and click the web banner. In the Customer Profile Viewer, after repeatedly logging in as Troy and clicking on the web banner each time, you see that the **Original model propensity** goes up when a target response is recorded.



Decision time	Name	Final propensity	Original model propensity
3/1/22 7:11 AM	StandardCard	0.7056	0.875
3/1/22 7:03 AM	StandardCard	0.9140	0.8333333333333334
3/1/22 7:02 AM	StandardCard	0.6234	0.75
3/1/22 7:00 AM	RewardsCard	0.5050	0.75
3/1/22 6:56 AM	StandardCard	0.7667	0.5
3/1/22 6:55 AM	RewardsCard	0.8000	0.5

Likewise, when Troy repeatedly ignores the offer, alternative responses are recorded after the Response timeout elapses.

Decision time	Name	Final propensity	Original model propensity
3/1/22 7:51 AM	StandardCard	0.5915	0.5625
3/1/22 7:49 AM	StandardCard	0.7478	0.6428571428571429
3/1/22 7:47 AM	StandardCard	0.5859	0.75
3/1/22 7:45 AM	RewardsCard	0.8564	0.625
3/1/22 7:41 AM	RewardsCard	0.9000	0.8333333333333334
3/1/22 7:40 AM	StandardCard	0.9387	0.9

Consequently, the **Original model propensity** for Troy and customers like Troy decreases.

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

- How to request a decision for a customer in Customer Profile Viewer.
- How to configure additional potential predictors for an adaptive model.
- How to explore the original model propensities and the final propensities in Customer Profile Viewer.

Creating parameterized predictors

Introduction

Learn how to improve the predictive power of your adaptive models by creating parameterized predictors. 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.

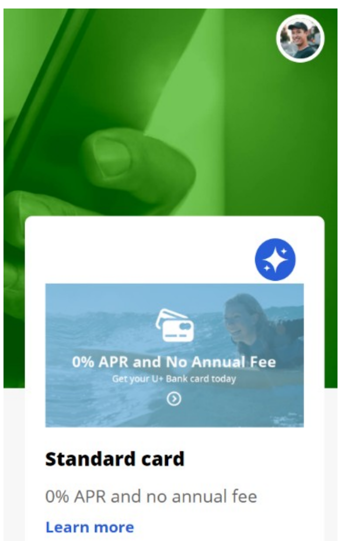
Challenge

To practice what you have learned in this topic, consider taking the Creating parameterized predictors challenge.

Transcript

This demo shows you how to create parameterized predictors for adaptive models.

U+ Bank shows personalized offers on their website when customers log in.

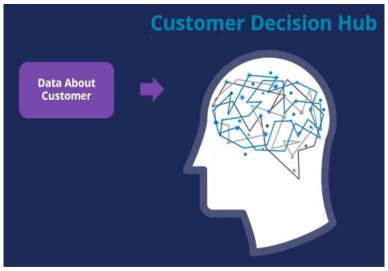


The bank relies on Pega Customer Decision Hub™ to decide which offer to show the customer.

Customer Decision Hub uses a prediction to predict the likelihood that a customer clicks on the offer.

The prediction ingests data about the customer to calculate the propensity.

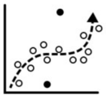
Ideally, this data includes the customer profile, interaction context, customer behavior and predictive model scores.



Customer profile

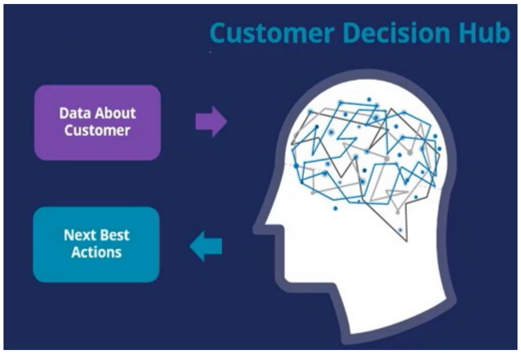
Interaction context

Customer behavior



Model scores,

Customer Decision Hub weighs the propensity to decide which offer to show on the website, balancing customer relevance and business priority.



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.

This video demonstrates the implementation of two new parameterized predictors that can add additional predictive power to the models.

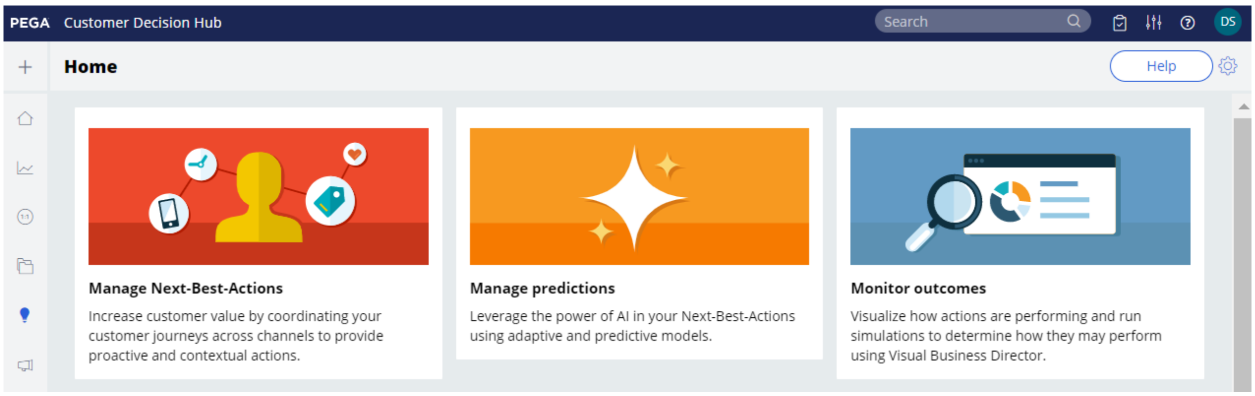
The first predictor is the ratio of customer visits to the Loans web page in the last 30 days and in the last 90 days.

A high value may indicate the increasing interest of the customer in the content of this page.

The second predictor is the score of a churn model running in Customer Decision Hub.

A customer that is likely to churn in the near future may be interested in different offers than a customer that is predicted to remain loyal.

Managing predictions is a regular data scientist task.



The prediction that calculates the propensity that a customer will click on the offer on the U+ Bank website is the **Predict Web Propensity** prediction.

One or more predictive models drive a prediction.

The **Web Click Through Rate** adaptive model drives the **Predict Web Propensity** prediction.

Supporting models

Name	Component name	Type	Performance	Status
Web_Click_Through_Rate	Web_Click_Through_Rate_Customers	Adaptive model	73.86 AUC	ACTIVE

Adaptive models automatically determine which fields are used as predictors, based on the individual predictive performance and the correlation between active predictors.

Models Predictors

Data last refreshed at
June 14,2021 01:41:36 AM

Refresh data

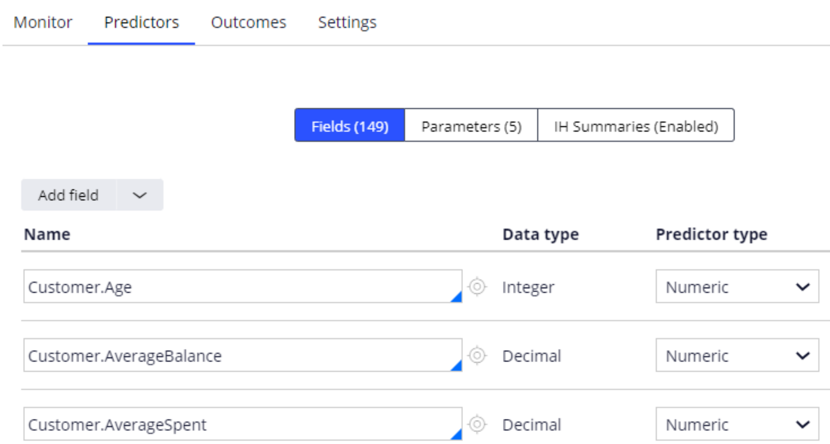
Predictor name	# Models ▼ active	# Models inactive	Minimum performance	Maximum performance	p
Customer.Age	4	0	56.74	79.25	
Customer.AverageBalance	4	0	54.78	61.26	
Customer.AverageSpent	4	0	57.87	88.48	
Customer.CLV_VALUE	4	0	53.85	61.05	
Customer.CreditScore	4	0	54.69	64.07	
Customer.MonthlyPremium	4	0	55.60	65.98	
Customer.NetPromoterScore	4	0	52.66	57.58	
Customer.RiskScore	4	0	54.01	66.65	
Customer.Gender	3	1	50.83	61.24	
Customer.LifeCyclePeriod	2	2	50.00	56.66	
Customer.DMOptIn	1	3	50.43	52.52	
Customer.MaritalStatus	1	3	50.29	52.72	
Customer.SubscriptionFlag	1	3	50.70	55.16	
Customer.BranchCode	0	4	50.00	50.00	

Only fields with a predictive performance above the threshold become active predictors in one or more models.

And, when predictors are highly correlated, they are grouped and only the best-performing predictor from the group is used.

It is therefore a best practice to make many uncorrelated fields available to the models as potential predictors.

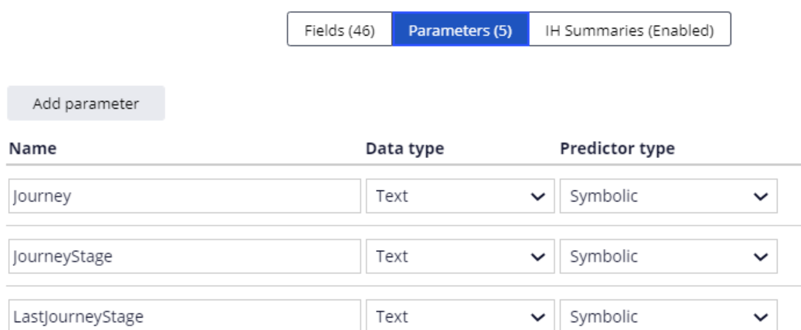
In an adaptive model rule, three distinct types of predictors can be defined.



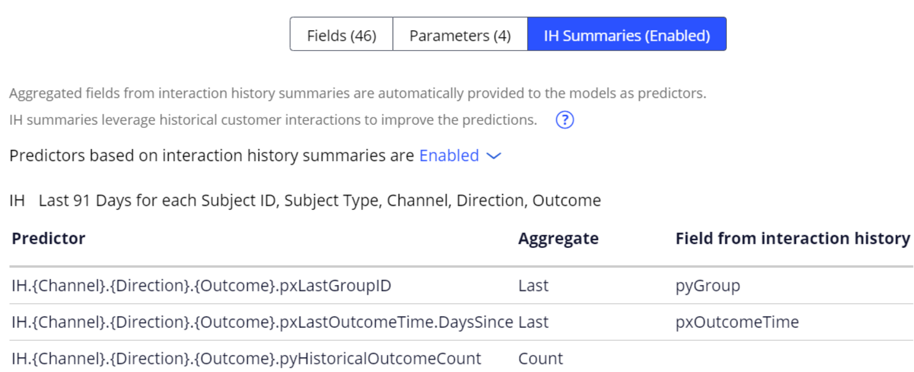
The first predictor type concerns fields that contain customer attributes, such as age and average account balance, and customer behavior data summarized in Customer Profile Designer.

The second predictor type is parameterized to reference attributes that are not part of the customer profile.

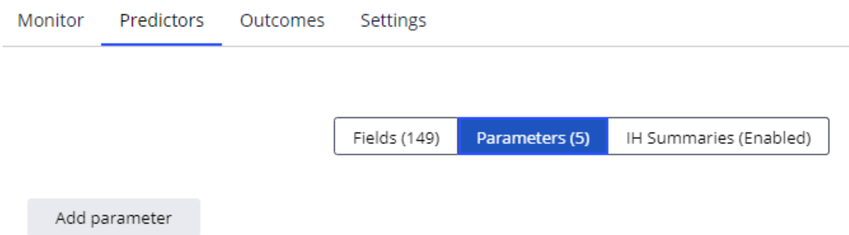
Examples that can provide additional predictive power include derived fields, such as the time of day, and model scores.



The third predictor type is an Interaction History summary, which leverages historical customer interactions.



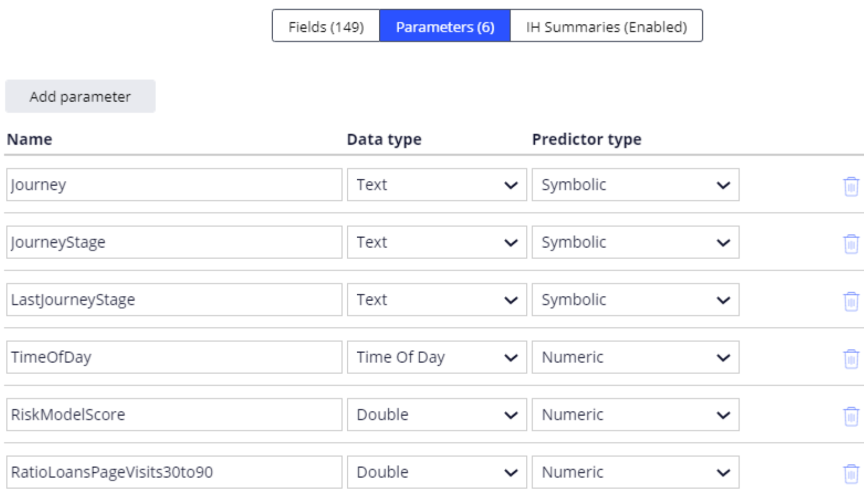
This demo focuses on parameterized predictors.



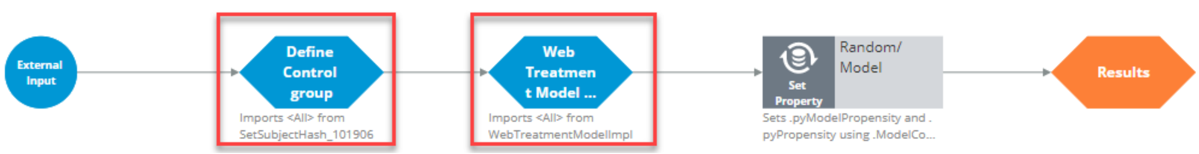
For adaptive learning, there is no difference between parameterized predictors and regular predictors.

To create a parameterized predictor, you add it in the adaptive model rule.

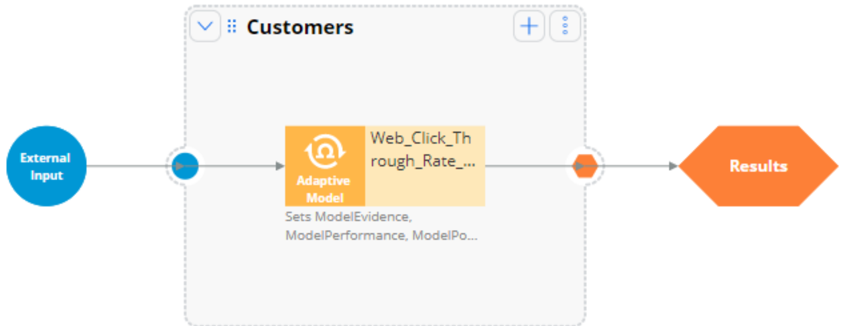
In the example of *Loans page views 30 days-to-90 days ratio*, the data type is double.



A prediction is implemented as a decision strategy.



The decision strategy defines the control group and contains a sub strategy that references the adaptive model rule that drives the prediction, in this case the **Web Click Through Rate** model.



The values of parameterized predictors are set in the adaptive model component in the decision strategy.

Supply data via

Parameterized predictors

RatioLoanPageVisitsLast30to90Day

The expression used for the new predictor says that if a customer has never visited the page in the last 90 days the value is set to zero …

… otherwise it is set to the number of visits in the last 30 days divided by the number of visits in the last 90 days.

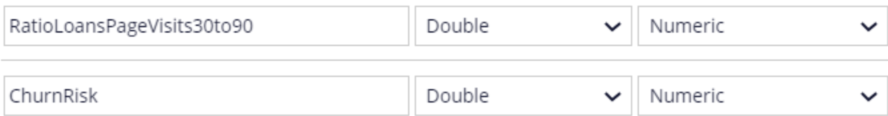
Expression builder **Browse** **Test**

```
IF(Primary.Customer.FSClickstream.LoansPageVisitsLast90Days=0,0,
divide(Primary.Customer.FSClickstream.LoansPageVisitsLast30Days,Primary.Customer.FSClickstream.LoansPageVisitsLast90Days))
```

When you add a parameter in the model rule, it automatically enables the field for input in the adaptive model component.

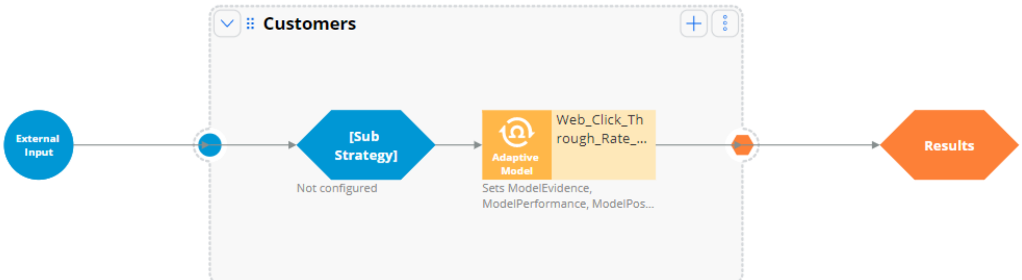
In the second use case, you want to include the score of a churn model running in Customer Decision Hub as a potential predictor.

Just as in the first use case, start by adding the new parameter to the adaptive model.

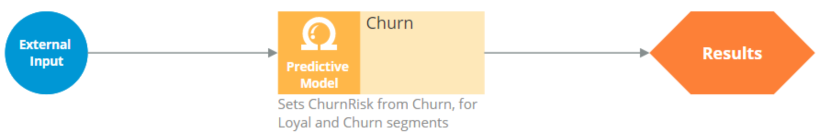


For this use case, you need to alter the prediction decision strategy.

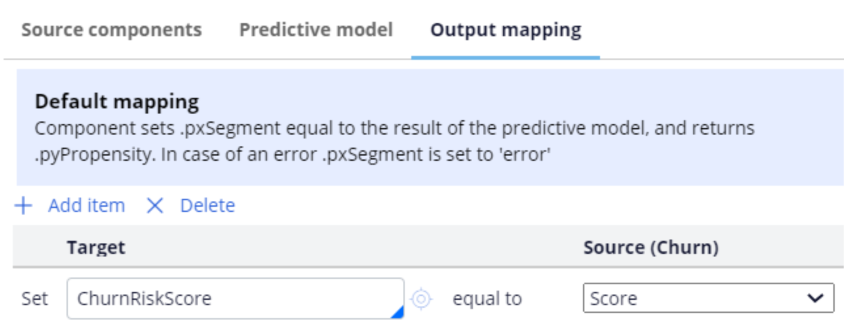
To access the model scores, you add an external sub strategy that returns the score of a predictive model that calculates the churn risk of a customer.



Create a new sub strategy that references the churn model on the customer page ...



... and map the score of the model to a new property in the Strategy Result class.



The **Score** output field of the churn model is a numeric field with values from 0 to 1000. A high value indicates a high churn risk.

You can now use this **ChurnRiskScore** property to populate the predictor in the adaptive model component in the decision strategy.

RatioLoanPageVisits30to90	@IF(Primary.Customer.FSClickstream.
ChurnRiskScore	.ChurnRiskScore

After refreshing the data, the two parameterized predictors are available to the models.

They are currently inactive, but they will become active predictors over time, when they prove to have predictive power.

The Churn model is now one of the supporting models in the prediction that calculates the likelihood that a customer clicks a specific offer.

Supporting models

Name	Component name	Type	Performance	Status
Churn	Churn	Predictive model	——	ACTIVE
Web_Click_Through_Rate	Web_Click_Through_Rate_Customers	Adaptive model	73.86 AUC	ACTIVE

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

How to create parameterized predictors

Monitoring adaptive models

Description

It is a regular data scientist task to inspect the health of the adaptive models and share the findings with the business. The predictive performance and success rate of individual adaptive models provide information that can help business users and decisioning consultants to refine business processes. The content of this moduleshowcases adaptive models used in Customer Decision Hub predictions that aim to optimize customer engagement but is equally relevant for case management predictions.

Learn how to monitor the performance of the adaptive models and how to export the raw data that adaptive models have processed to inspect and validate the predictors.

Learning objectives

- 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
- Export the raw data that is used by adaptive models

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.

Inspecting adaptive models

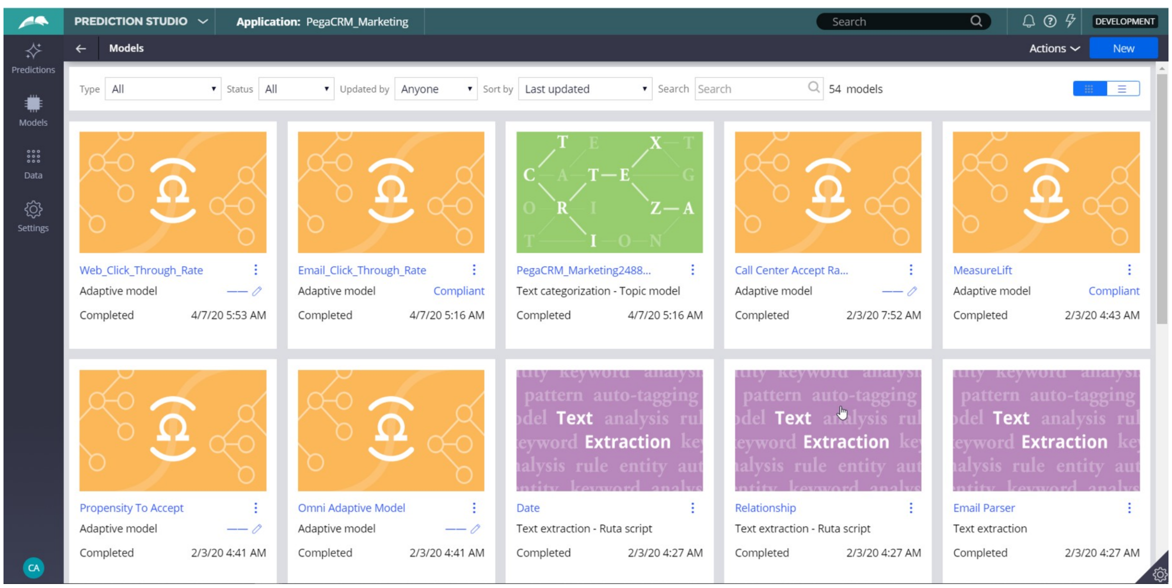
Introduction

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 of the health of adaptive models and their predictors is a regular data scientist task that can be performed in Prediction Studio.

Transcript

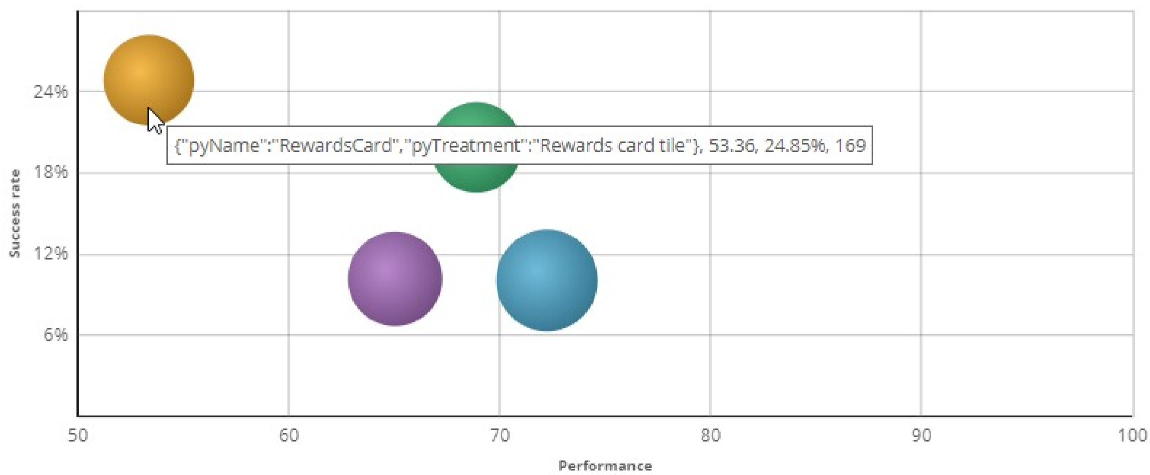
This demo will show how to inspect the health of your adaptive models and their predictors. This is a regular data scientist task.

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.



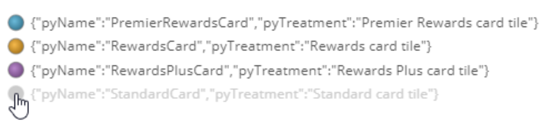
Each bubble represents the model for a specific action.

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, 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.

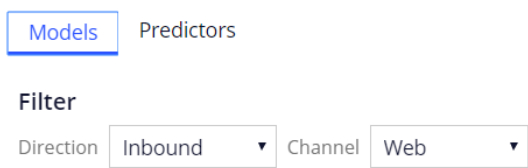
The information you see here is extracted from the Adaptive Data Mart, which is a reporting view of the Adaptive Decision Manager (ADM) server.

The Adaptive Data Mart is built automatically by a process running in the background. This process creates snapshots at regular time intervals.

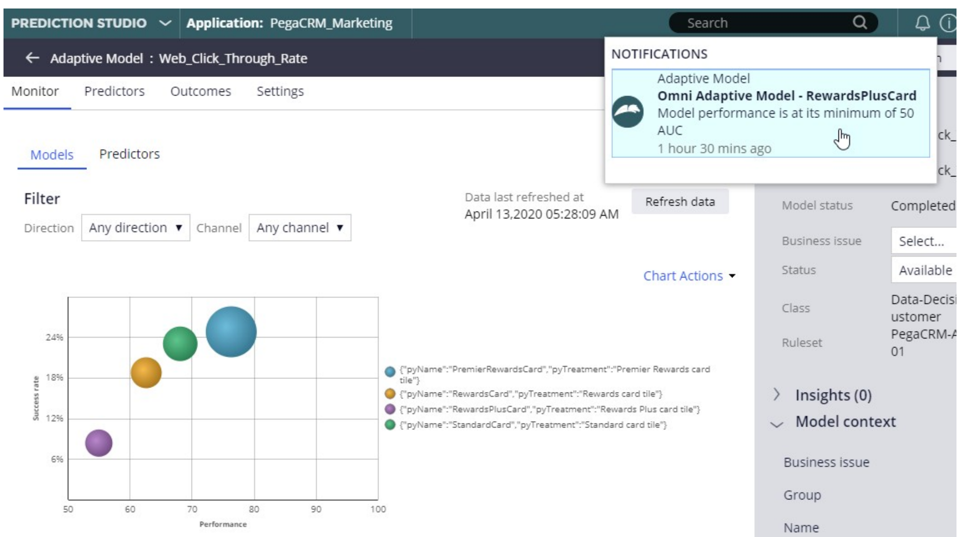
This means that the data as reported may not be the same as the data you see when you view it in real-time. You can refresh the view to synchronize the data.

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

In this case, the Age predictor is used in all four models.

The HasMorgage predictor is active in three models and inactive in one model, where its predictive power is below the threshold.

The default value for this threshold is 52 percent.

Predictor selection

Activate predictors with a performance above

0.52 AUC

The system continuously monitors the predicting power of every predictor. If the predicting power of a predictor drops below the threshold value that predictor is deactivated.

The data that is used to visualize the models in the bubble chart is displayed in a table below the chart.

For each model number of responses, success rate and performance are shown.

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

Predictors | Score distribution | Trend

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

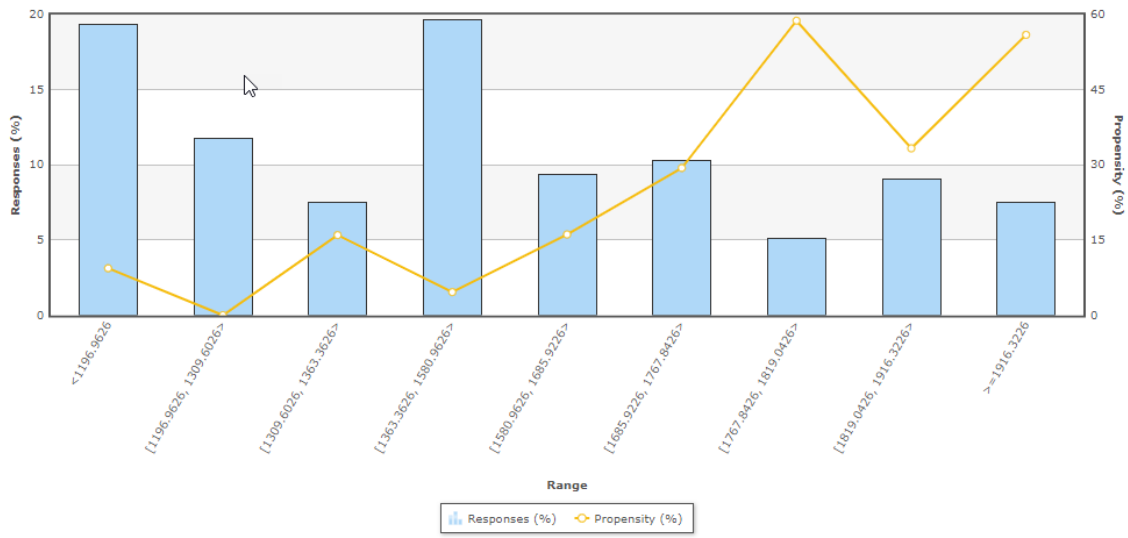
Predictors	Status	Type	Performance (AUC)	Range/Symbols(#)	Bins(#)
Customer.AverageSpent	Active	Numeric	80.86	[1001.33; 1997.33]	9
Customer.Age	Active	Numeric	75.54	[19.0; 80.0]	9
Customer.InteractionContext.PreviousWebpage	Active	Symbolic	74.22	4.00	4
Customer.AverageBalance	Active	Numeric	73.51	[506.21;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

There are three tabs, reporting on predictors, the model score distribution and the trend. In the predictors report, you can examine the performance of individual predictors.

Let’s examine a couple of them.

In this case, the best performing predictor is AverageSpent. This a predictor of type numeric.

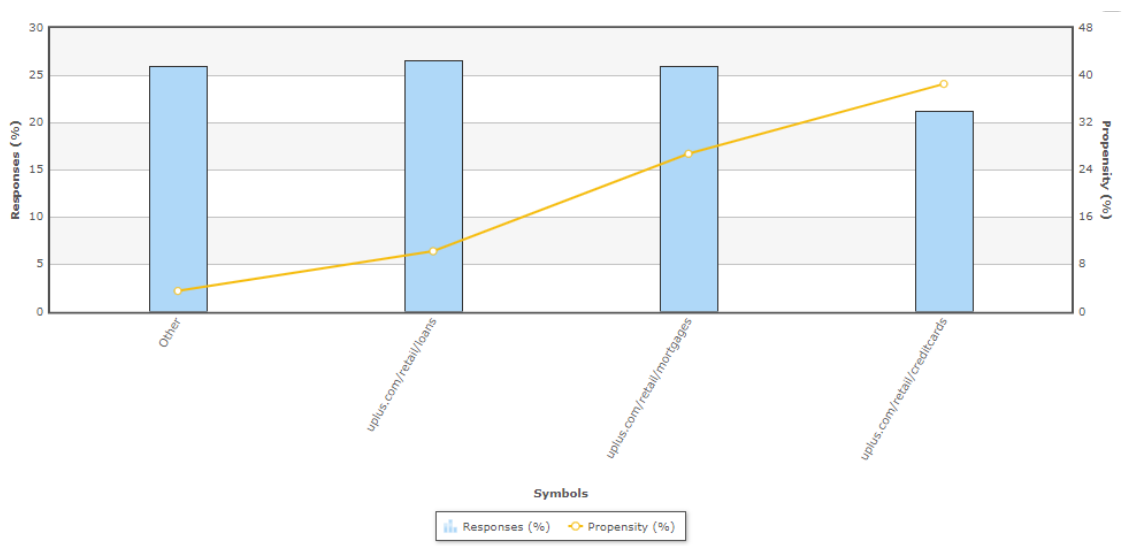
The system split the AverageSpent predictor into 9 bins. Each bin has its own offer propensity.



Propensity is the likelihood of positive customer behavior, which in this example is clicking on a web banner.

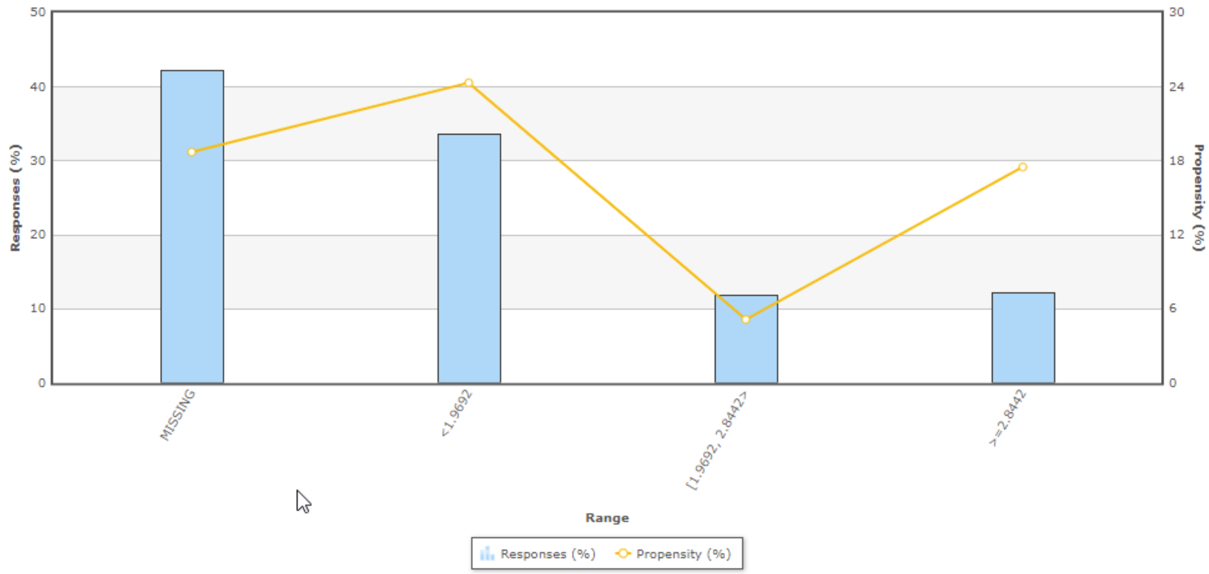
Now, let’s examine the symbolic PreviousWebpage predictor.

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.

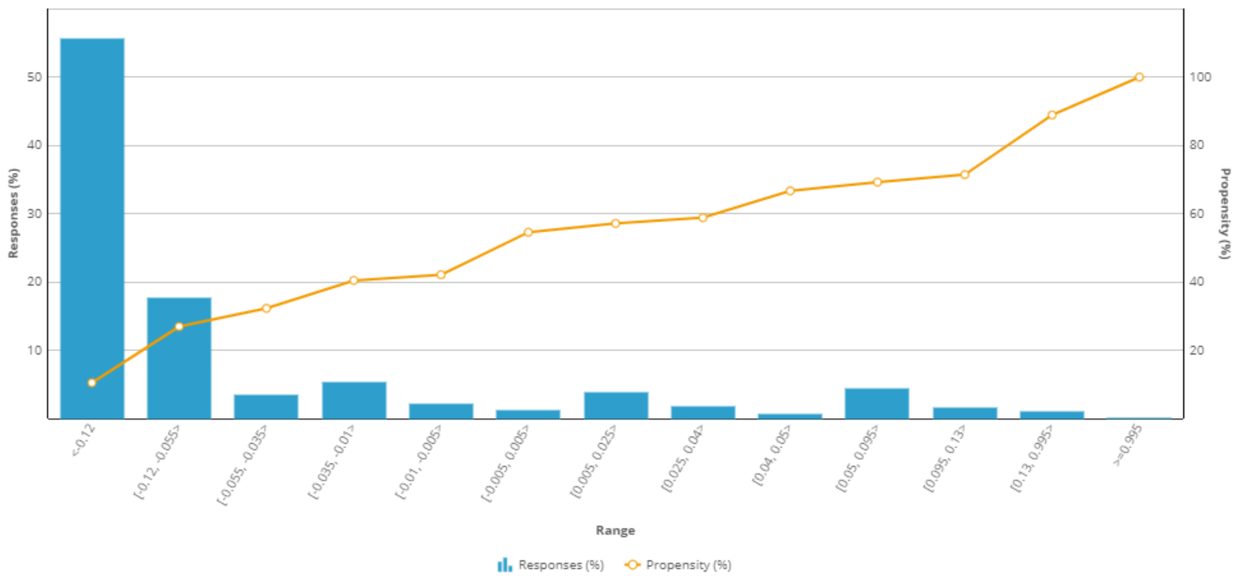
In this example, the adaptive system established that the number of days since the offer was accepted is a well-performing predictor.



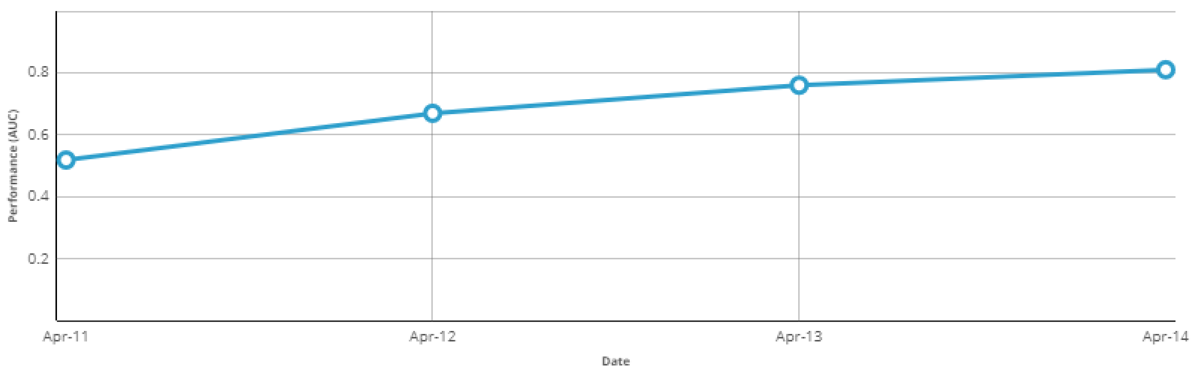
The system grouped three predictors that are correlated. It then marked two of them as inactive. Inactive predictors are not used in the propensity calculation.

▼ 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 historical data

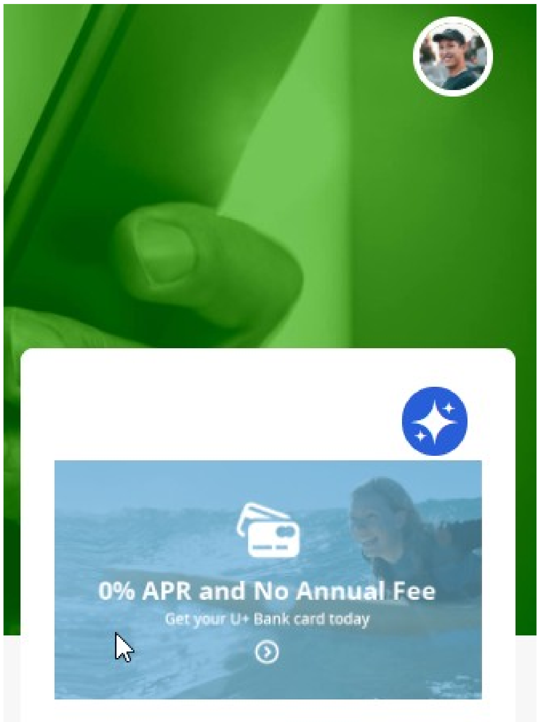
Introduction

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

Transcript

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

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



When a customer is eligible for multiple credit cards, adaptive models decide which card to show.

When the customer ignores the banner, the adaptive model that drives the decision regards this as negative behavior.

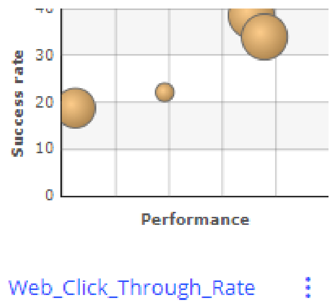
When a customer clicks on the banner, the model regards this as a positive behavior.

As a data scientist, you may want to inspect the raw predictor data used by an adaptive model and the customer interaction outcome to validate data assumptions and check for concept drift.

You can also use the data to build various predictive models externally.

All models are managed in Prediction Studio.

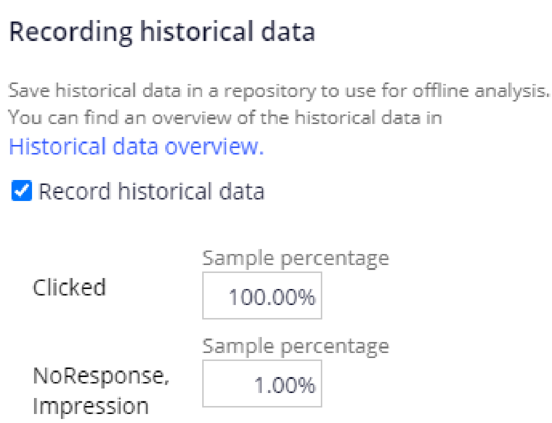
The adaptive model that drives the decision over which banner to display is the **Web Click Through Rate** model.



To extract the data, you enable the recording of historical data for a selected adaptive model.

A web banner typically has a low click-through rate and a significantly lower number of positive responses than negative responses.

In such cases, you can sample all positive outcomes and just one percent of the negative outcomes to limit the storage space needed.



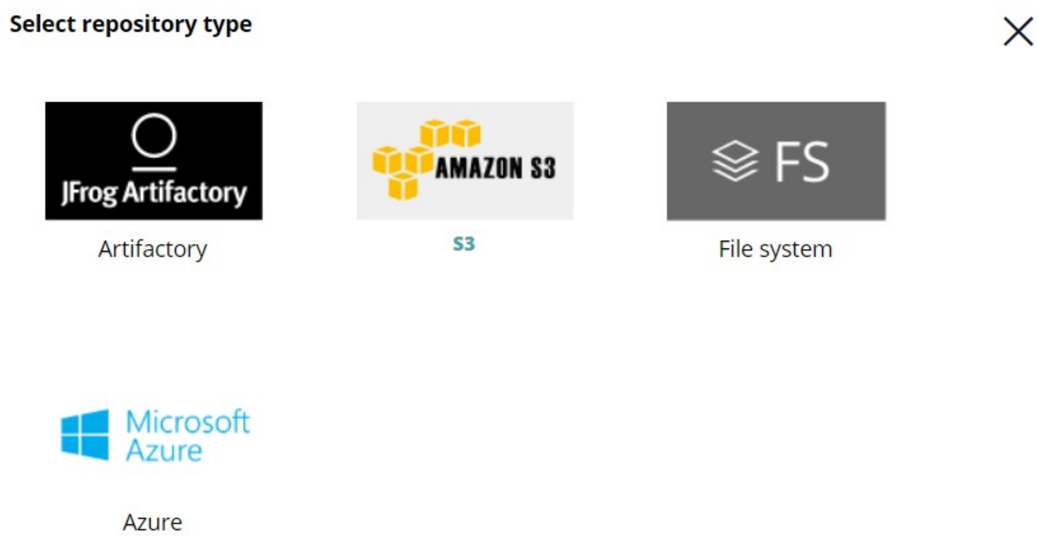
The sample percentages determine the likelihood that a customer response is recorded.

The system stores the predictor data and outcome as a JSON file in a repository of your choice.

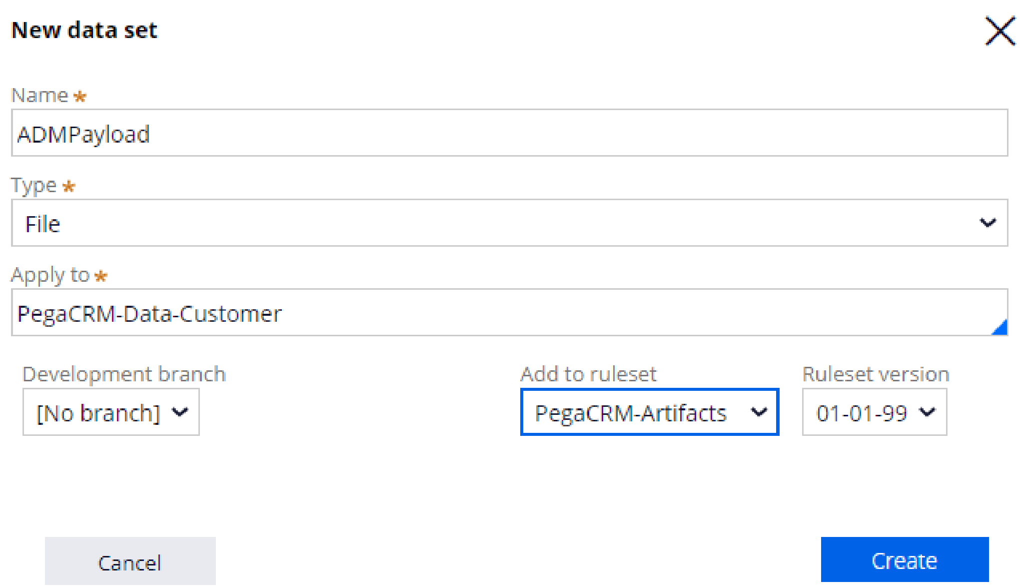
By default, the data is stored for 30 days in the **defaultstore** repository.

However, this repository points to a temporary directory, and a system architect should switch to a resilient repository to avoid data loss.

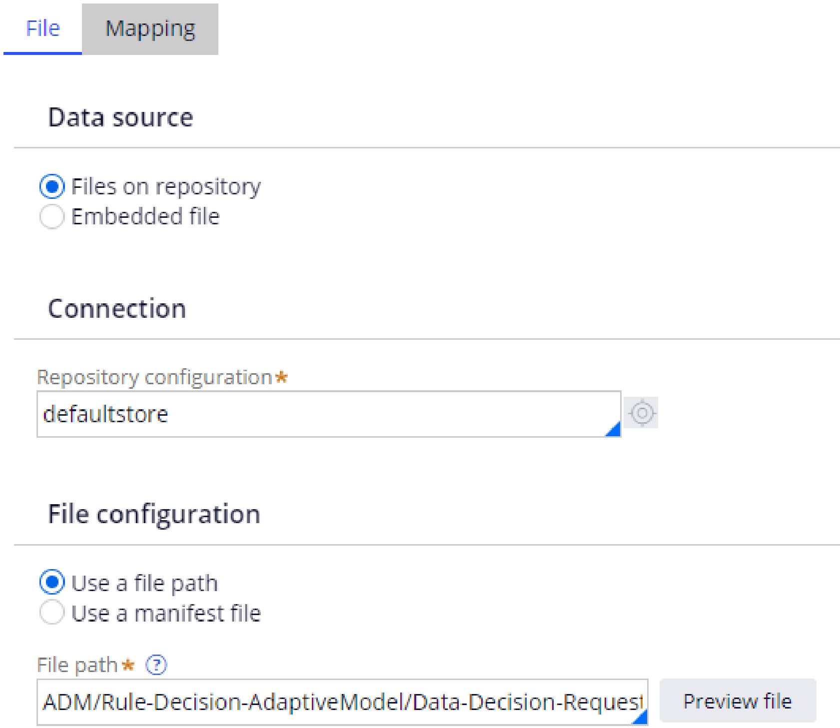
Supported repository types include Microsoft Azure and Amazon S3.



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

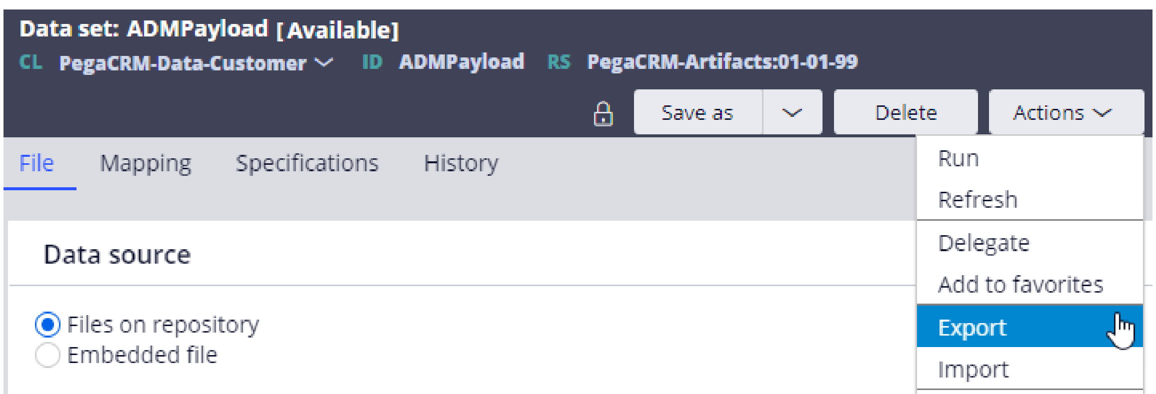


The data set is mapped to the file that contains the recorded historical data.



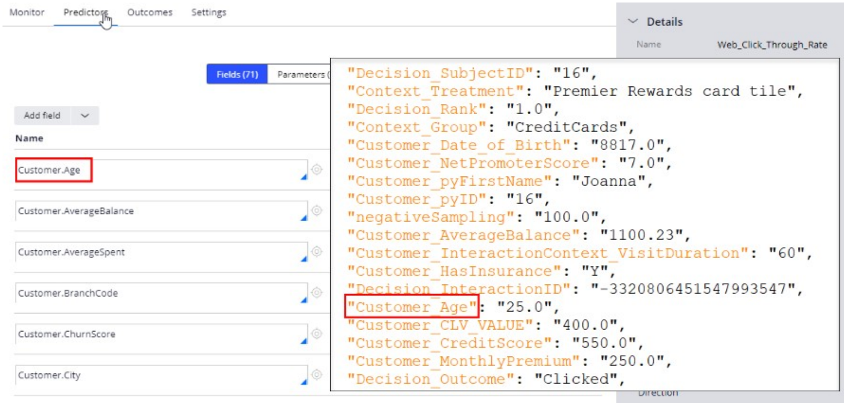
With these settings in place, the input data used for the prediction and associated outcome are stored in the configured data set when customers see an offer and click on an offer.

The system architect can download the data set in DEV Studio.



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.



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.

The impact of machine learning

Description

The boost in the success rate, also known as lift, that artificial intelligence (AI) achieves is an important business metric. To report on this metric, a data scientist can use predictions. Predictions add best practices to predictive models and use a control group as a benchmark to measure lift. Customers in the control group receive a random offer instead of the one selected by AI. The use of a control group also adds a degree of exploration to the exploitation of the models.

Learning objectives

- Describe how predictions add best practices to predictive models
- Explain how the use of a model control group allows the measurement of lift
- Explain how the use of a model control group adds exploration to the exploitation of the models

Measuring lift using a control group

Introduction

The boost in success rate achieved with adaptive models can be measured using a control group in Predictions. Predictions are strategies that add best practices to predictive models. Customers in the control group will receive a random offer instead of the one recommended by the AI. This allows a comparison between the control group and the rest of the audience. It also enables the models to explore alternative outcomes.

Transcript

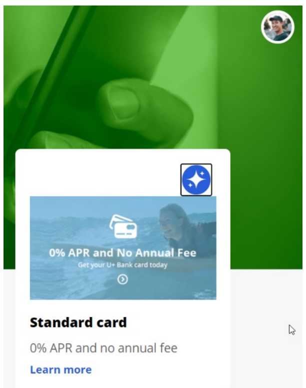
This demo shows you how to use a control group to measure the boost in success rate the adaptive models achieve.

U+ Bank is a small retail bank. When customers log in to the U+ Bank website, they see the credit card offers for which they qualify based on the engagement policy defined by the business.

When customers qualify for multiple credit card offers, adaptive models decide which is the best offer to show.

Adaptive models are self-learning and will automatically learn from customer interactions.

The models interpret a click on the offer banner as positive behavior. When a customer ignores the banner, this is interpreted as negative behavior. Models are updated frequently.

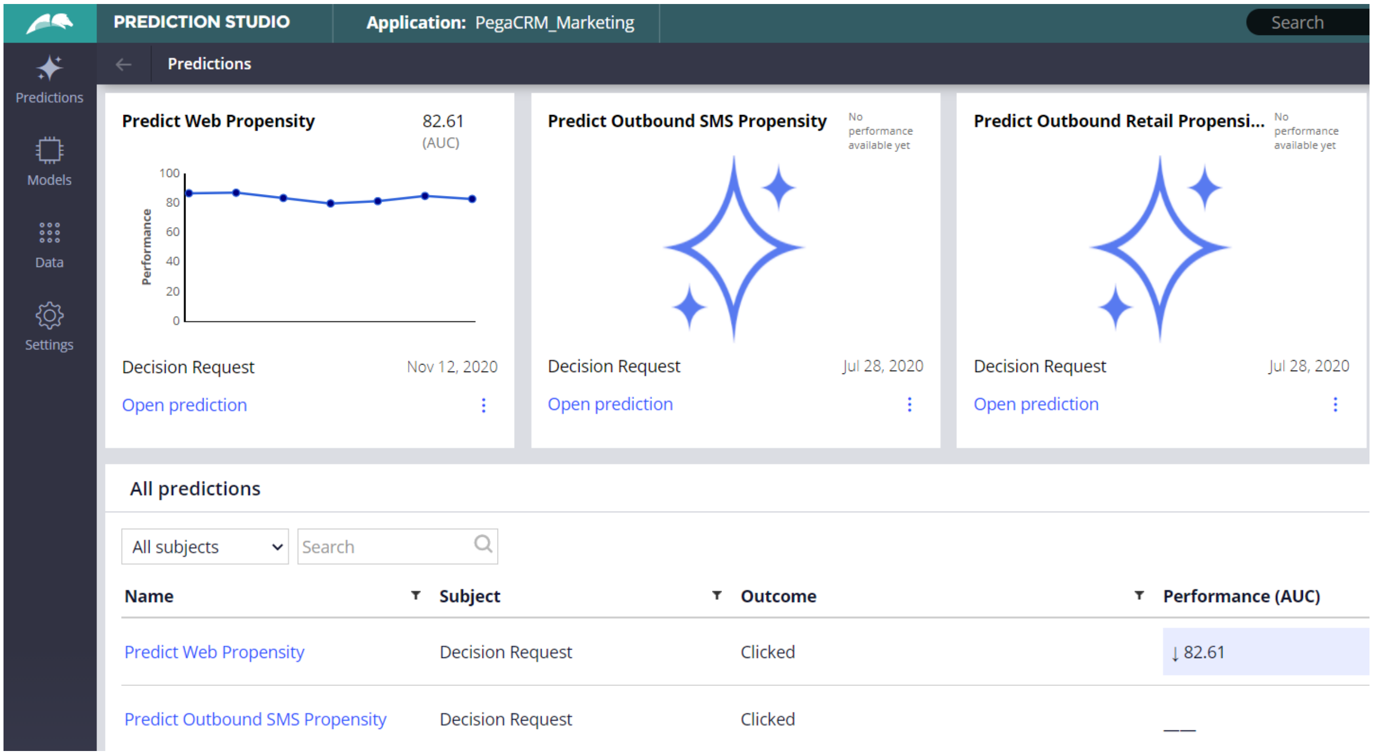


An important business metric is the lift in success rate that the models generate.

To measure and report on this metric, a data scientist can use Predictions. Predictions add best data science practices to predictive models.

One of these is to use a control group as a benchmark for measuring the lift that the models achieve.

All out-of-the-box predictions available are listed on the Predictions landing page in Prediction Studio.



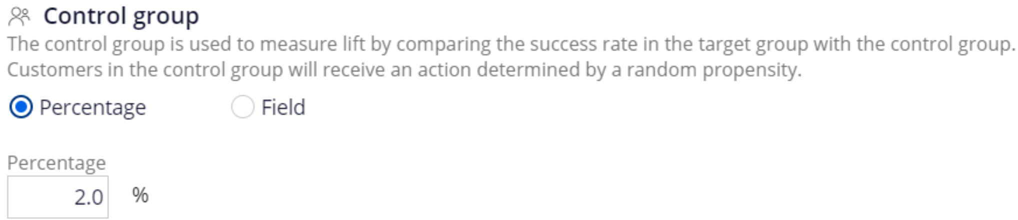
The **Predict Web Propensity** prediction monitors the responses from the U+ Bank website.

The outcome of this prediction is the propensity to click on a web banner.

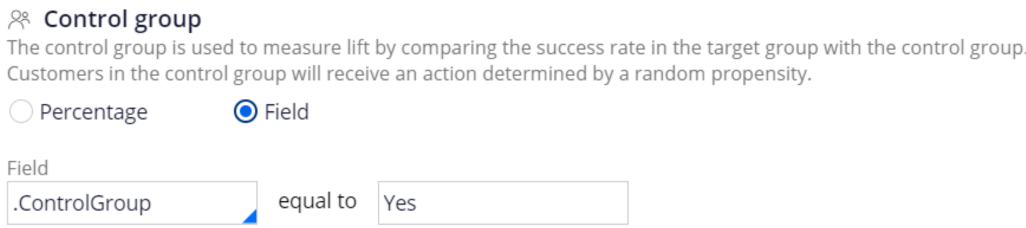
A percentage of the total number of customers is randomly reserved for the control group.

The customers in the control group receive a random action instead of the action that the AI recommends.

A small percentage is sufficient to measure lift. By default, the control group is set to reflect 2% of the population, but you can change that value if desired.

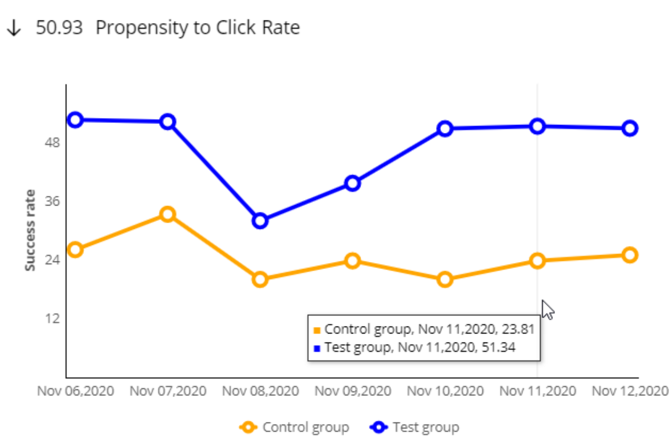


Alternatively, customers can be appointed to the control group based on a customer attribute.



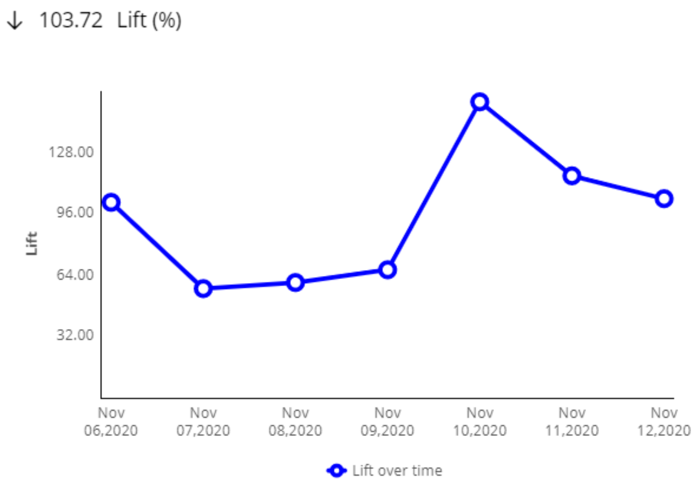
In the first chart, the yellow line shows the success rate, such as the accept rate, or, in the case of the **Predict Web Propensity** prediction, the click rate for the control group.

The blue line shows the success rate for all other customers, referred to as the test group.



As customers in the control group are presented with a random offer, the success rate is expected to be lower than for the test group, for whom the offer is based on the propensity to click.

The second chart shows the lift over time, which is the difference between the blue and yellow lines, expressed as a percentage.



The use of a control group allows the measurement and monitoring over time of the efficiency of the machine learning process.

The control group also enables another data science best practice: mixing some exploration into the Prediction exploitation.

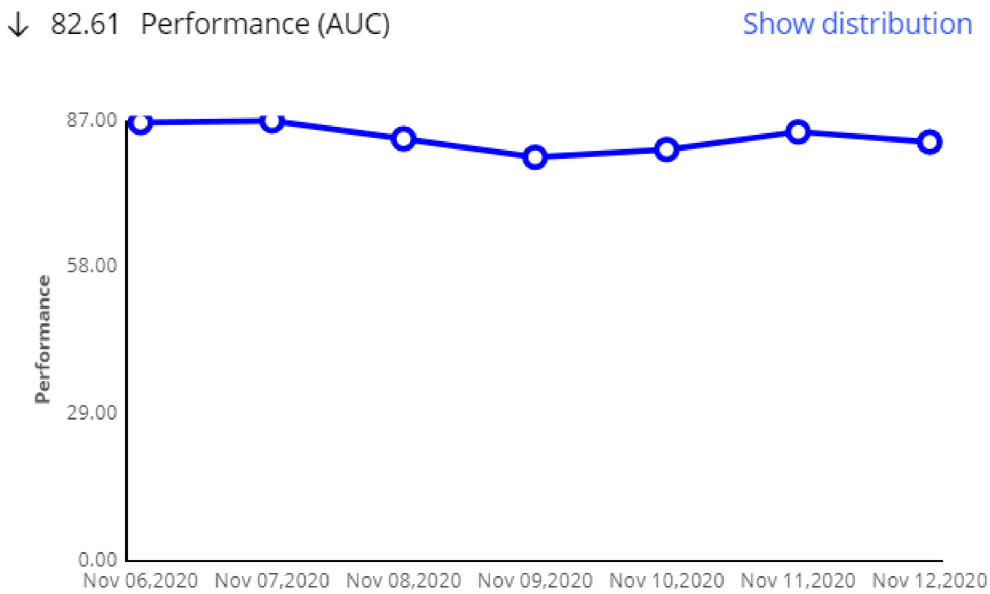
As the random offers made to the customers in the control group are not based on a particular customer profile, the models have a degree of freedom to explore.

In the notification section, two actionable insights related to lift are provided when applicable: absence of lift ...

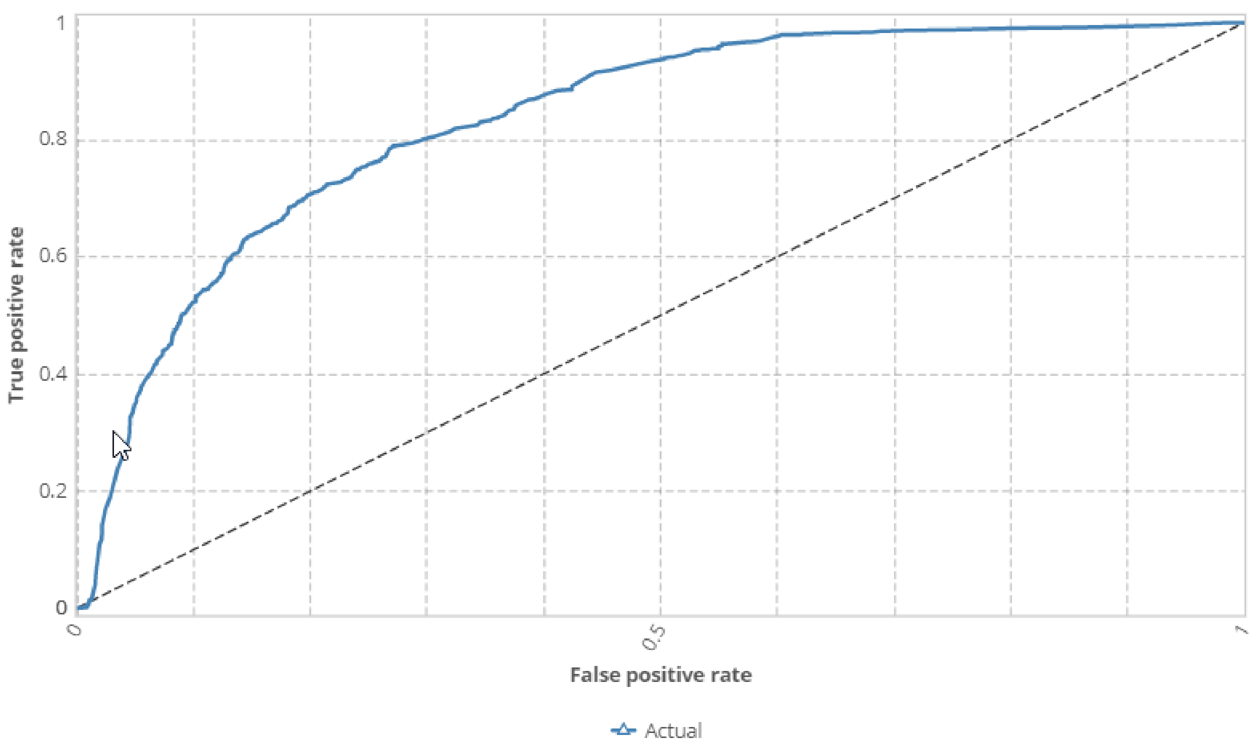
... and a lift that is significantly lower than in the previous week.

Both notifications prompt for an inspection of the model that drives the prediction as well as its predictors.

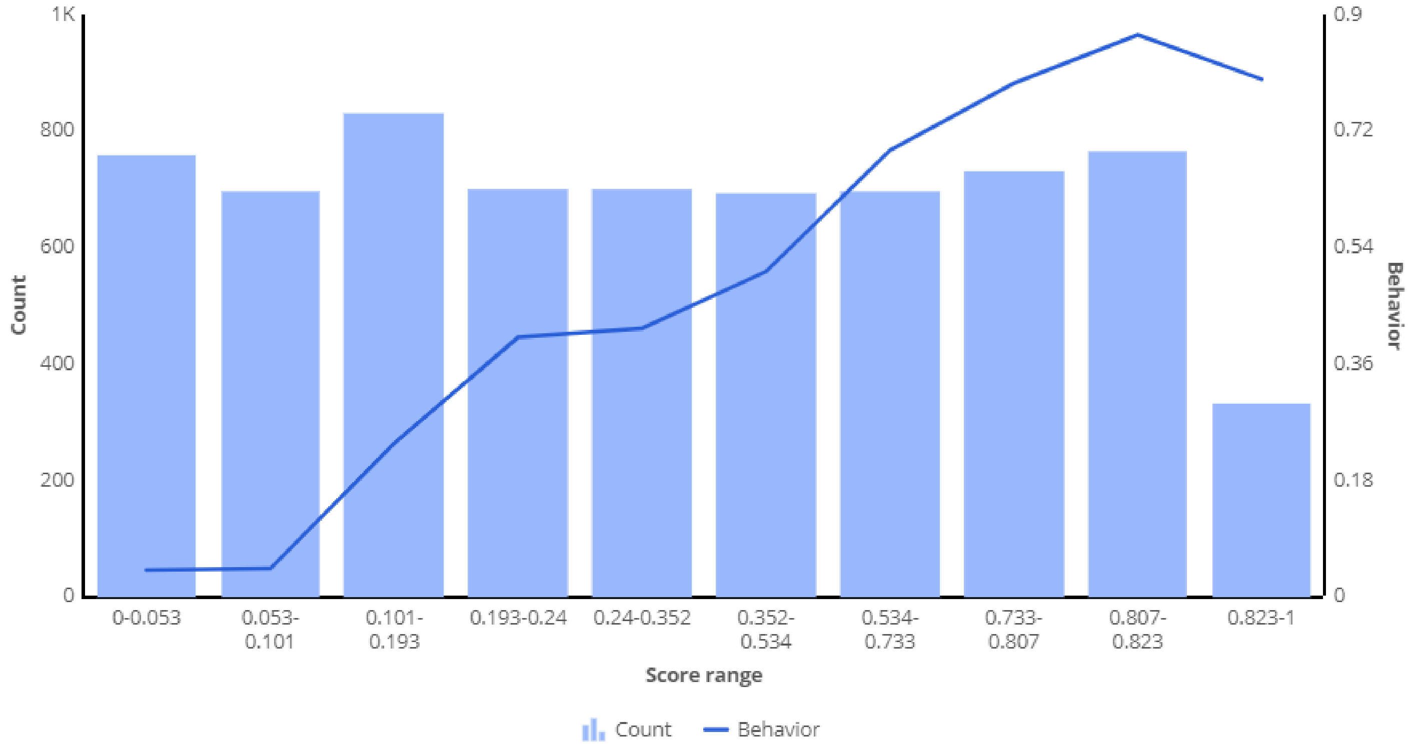
The third graph shows the performance of the model over time.



The ROC curve shows the true positive rate, or sensitivity, versus the false positive rate, or 1 minus the specificity, of the prediction.



Also, a decile distribution graph is provided for further inspection.



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

- How predictions add best practices to predictive models.
- How the use of a model control group allows the measurement of lift.
- How the use of a model control group adds exploration to the exploitation of the models.

Pega Process AI overview

Description

Gain a greater understanding of the key features, capabilities, and benefits of Prediction Studio. Prediction Studio is the dedicated workspace for data scientists to control the life cycles of predictions and the predictive models that drive them. Configure the predictions that are deployed in Pega Process AI™ to increase efficiency and effectiveness in case management.

Be aware that the content of the Pega Process AI mission is partly based on Pega Customer Decision Hub™ use cases. Although a data scientist working with Pega Process AI does not use the Customer Decision Hub portal, the material has a generic value.

Learning objectives

- Describe the use of Pega Process AI in case management
- Explain the types of predictions that are available in Prediction Studio

Pega Process AI overview

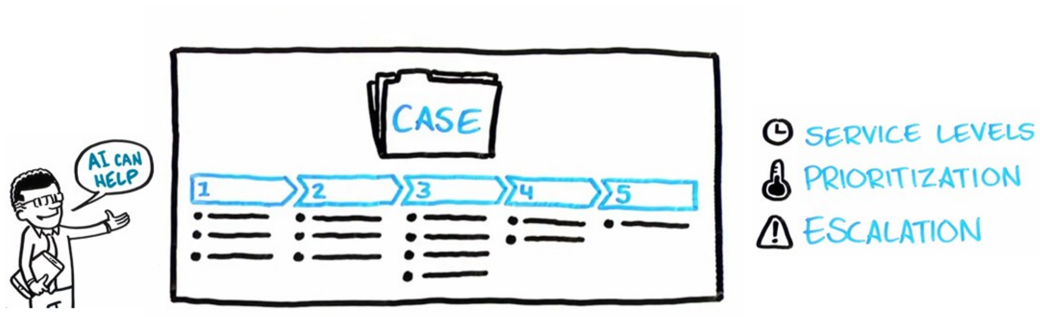
Introduction

In recent years, artificial intelligence has moved out of the labs and helped enterprises generate proven business value. At the same time, operationalizing AI can be a bottleneck. Pega Process AI™ tackles this problem by using AI to self-optimize processes and applying your own AI in Pega case management.

Transcript

This video provides an overview of the Pega Process AI capabilities in intelligent automation.

Process management aims to optimize business processes by increasing efficiency, consistency, and transparency, which decreases costs and improves quality.



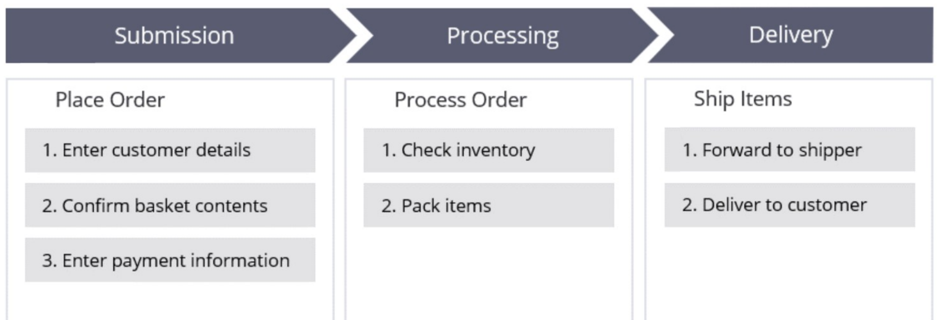
For example, consider an online order process. The customer submits an order, and the company processes and then delivers the order.

An Pega Platform application that models the online order process follows the same sequence as a series of stages. A **case type** is the abstract model of that process.

Case types model repeatable business transactions that might refer to a customer, or another entity, such as a machine in a maintenance case type.

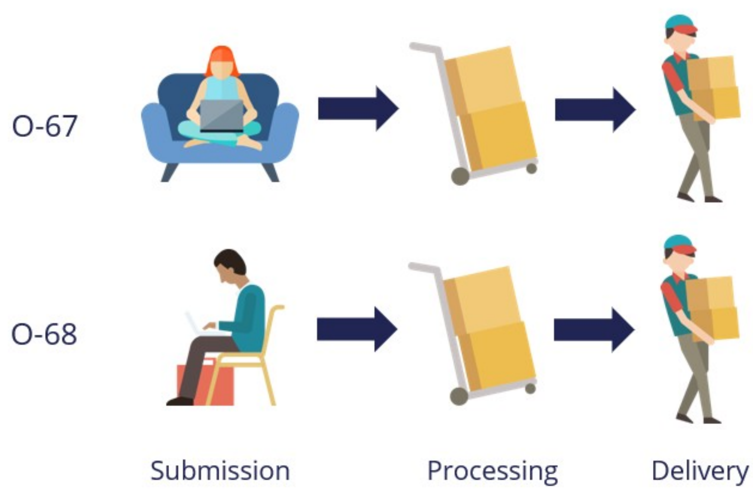
The **case life cycle** for a case type helps to visualize the work to complete as part of a business transaction.

Each stage in the life cycle contains the steps required to complete it and move to the next stage.



A **case** is a specific transaction instance of the case type.

Each time a user submits an online order, Pega Platform creates an order case and assigns the case a unique identifier.

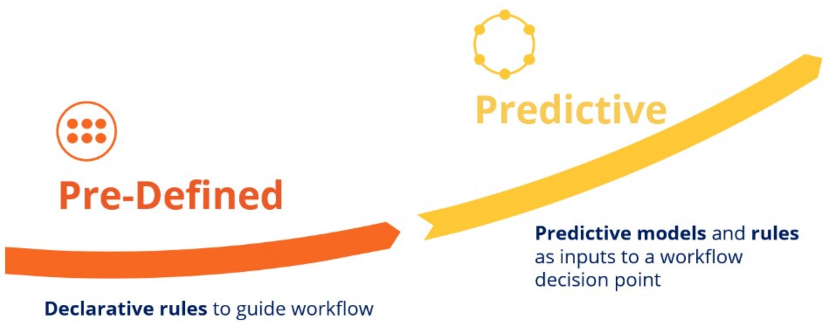


A case type can use declarative rules to manage the workflow, for example, to confirm that the order contains a valid shipping address or the order amount threshold to qualify for free shipping.

Pega Process AI can improve the quality of the decisions in the workflow by weighting in predictions, driven by predictive models.

The first approach is to operationalize existing predictive models that have proven their efficiency, to support the decisions that benefit from predictions, such as credit risk in a sales case or fraud risk in a claims case.

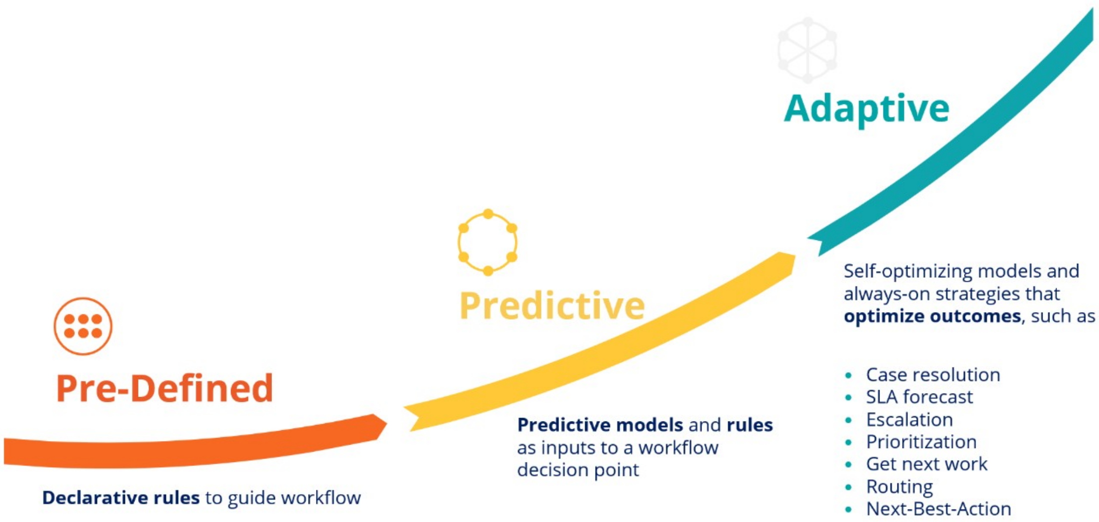
For example, the decision not to process an order can be based on a high credit risk score, and then the application can route the dubious claim for closer inspection.



The inputs for such a predictive model can be attributes of the case itself, such as the claimed amount in a claims case type, but they can also include data such as the number of claims submitted recently by the same customer.

You can build predictive models in Prediction Studio, import the models in the PMML and H2O formats, or run externally on the Amazon SageMaker and Google ML platforms to drive a prediction.

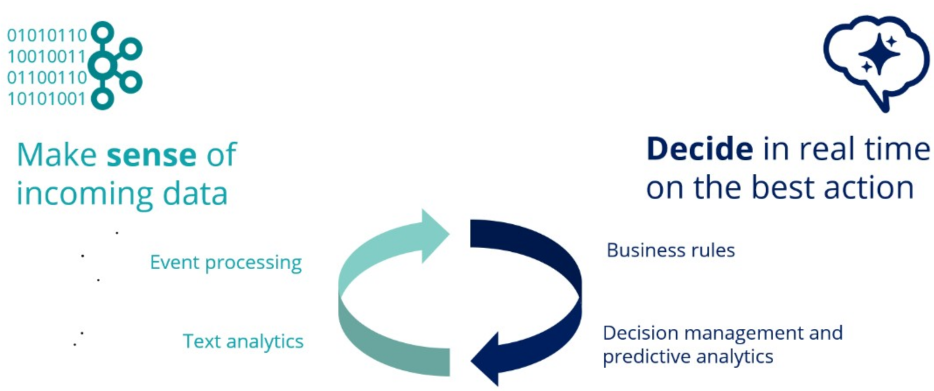
To optimize case outcomes, use adaptive models that can predict outcomes, such as case resolution, or intelligently prioritize and route cases to optimize business value and customer experience.



Adaptive models self-optimize by learning from the previous case outcomes that they capture.

The objective of Pega Process AI is to make sense of the incoming data and then decide on the best action to take in a specific stage of the case.

You can enhance the incoming data analysis by event processing to detect patterns of interest in real-time data streams and by natural language processing of incoming text.

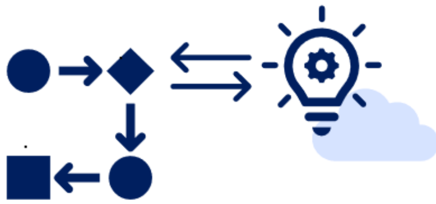


The decision is based on the business rules and supported by predictive analytics. This process is repeated every time that the case requests a decision.

As the number of processed cases increases and model evidence accumulates, the predictive power of the models increases over time.

To summarize, Pega Process AI uses artificial intelligence in case management to produce better business outcomes.

You can use real-time, adaptive case outcome predictions and your own AI models in custom predictions.



Custom AI predictions

Real-time, adaptive **case outcome predictions**

Predicting fraud

Description

Occasionally, an insurance claim might be erroneous or even fraudulent. To detect fraud and optimize the way in which the application routes work and meets business goals, learn how to use your own predictive models in case management.

Learning objectives

- Create a prediction to detect fraud
- Use the new prediction in a case type

Predicting fraud

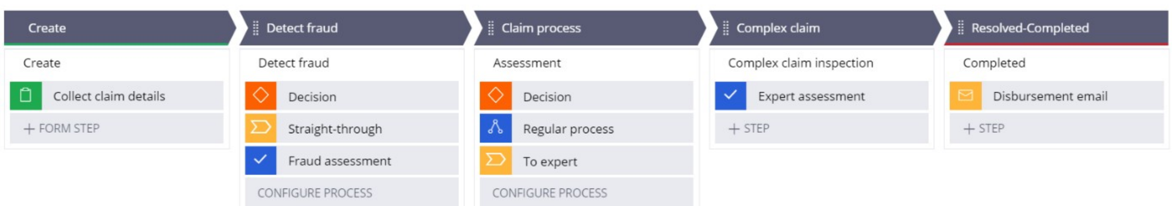
Introduction

Pega Process AI™ lets you bring your own predictive models to Pega. Use predictions in case types to optimize the way in which your application processes work and to meet your business goals. Learn how to use a predictive fraud model to effectively route suspicious claims for closer inspection.

Transcript

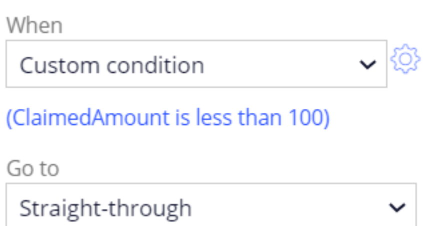
This demo will show you how to use a predictive fraud model in a case type to route suspicious claims to an expert.

U+ Insurance uses Pega Platform™ for case management. The life cycle of the case type that processes incoming car insurance claims contains a fraud detection stage, a regular process stage, and a complex claim process stage.

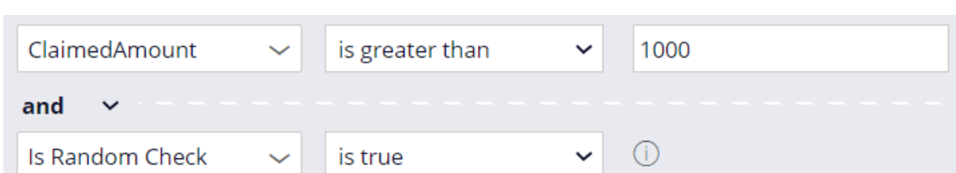


When the case is resolved, the claimant receives an email that communicates the decision.

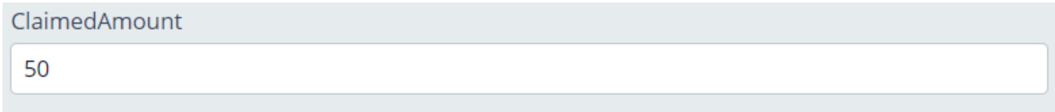
The decision step in the **Detect fraud** stage routes cases with a low claimed amount for straight-through processing.



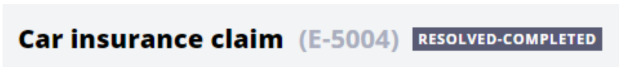
A set percentage of claims with a high claimed amount is routed to an expert for fraud assessment.

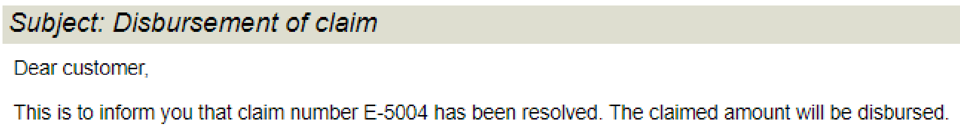


Consider this car insurance claim. The claimed amount is 50.

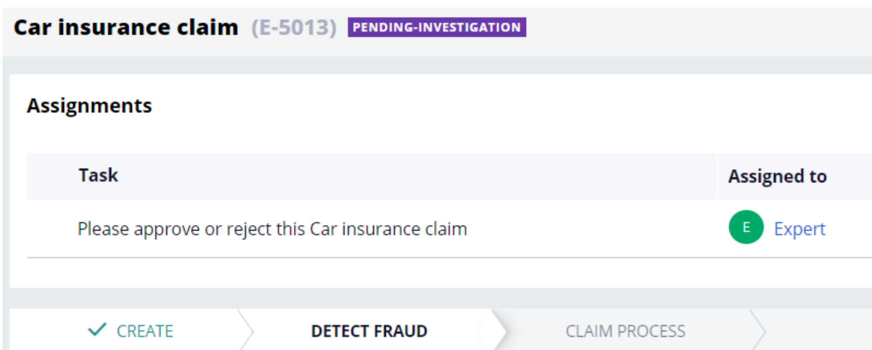


The claim qualifies for straight-through processing as the claimed amount is below the threshold. The case is automatically resolved, and the claimant receives an email that states that the claimed amount will be disbursed.

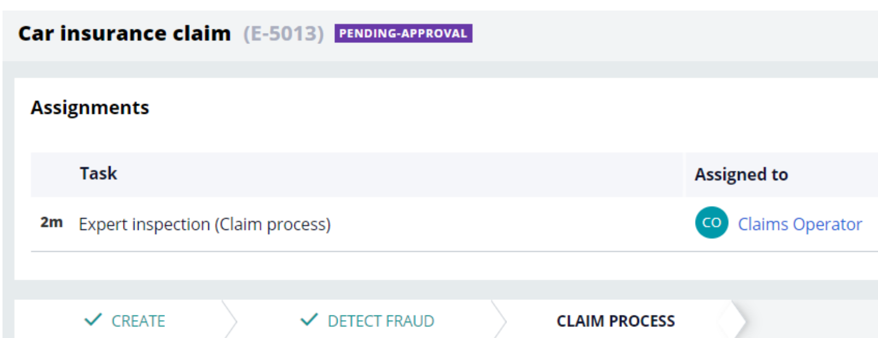




A fraud expert inspects a set percentage of cases with a high claimed amount.



After approval, the system routes the case to the regular claim process.



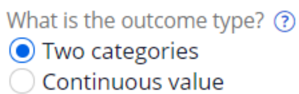
U+ Insurance wants to improve the effectiveness of fraud detection by using a predictive model that calculates the fraud risk of each claim.

The business requirements are that claims only qualify for straight-through processing if the fraud risk score is very low, while all claims with a high fraud risk score are inspected by the fraud expert. The routing of randomly selected cases to the fraud expert must remain in place to create a control group.

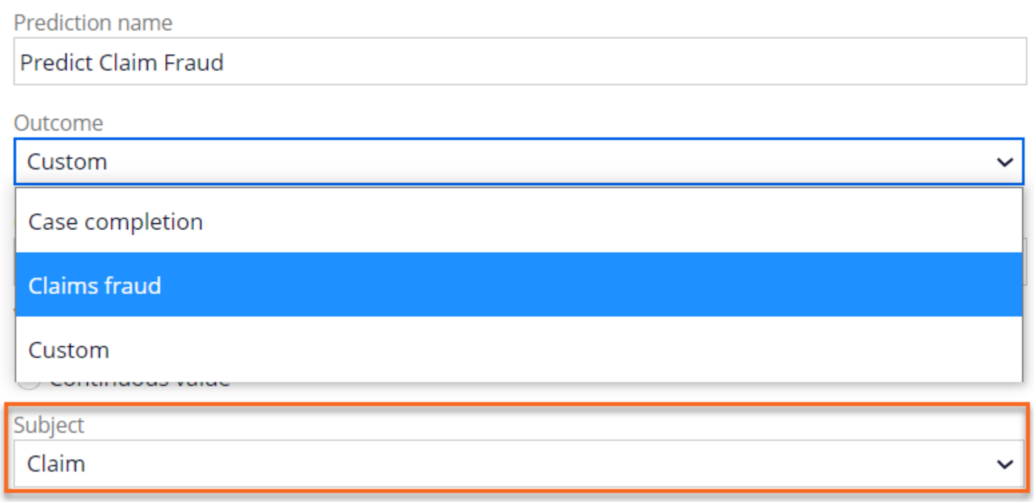
The data scientist team of U+ Insurance has developed a fraud model on the H2O.ai platform and has validated the model against historical data that the company captured.

The system qualifies a claim as abnormal if the probability of fraud exceeds the threshold; otherwise, the system classifies the case as normal.

To implement the fraud model, you create a new case management prediction. You can create a custom prediction that can forecast binary or numerical outcomes.



For fraud detection, Process AI provides an out-of-the-box template. The claim is the subject of the prediction.

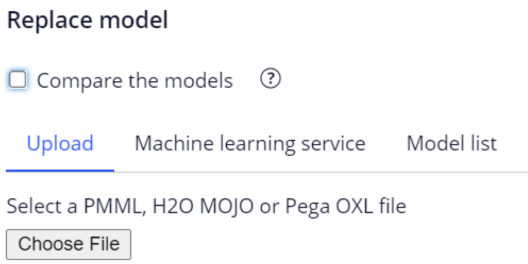


A placeholder scorecard initially drives the prediction.

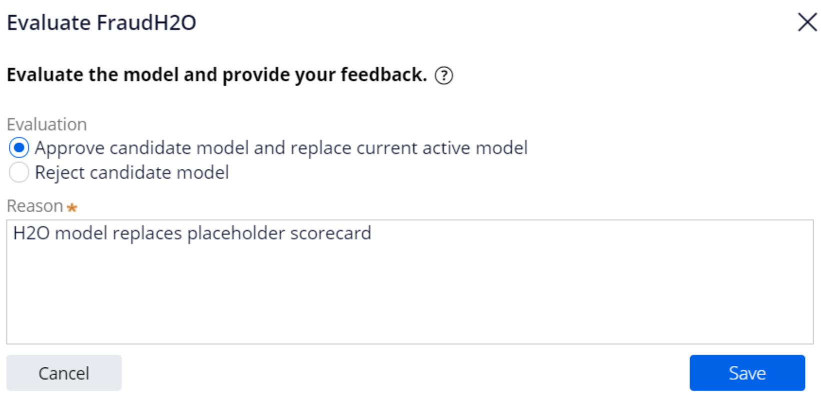
Claims fraud

Name	Type	Performance	Status
Predict Fraud	Scorecard	——	ACTIVE

When the predictive fraud model replaces the scorecard, the prediction is ready for implementation in the Car insurance claim case type. You replace the placeholder with a machine learning model, a scorecard, or a field that contains a precalculated score. You can upload a machine learning model as a PMML or H2O file. Alternatively, you can connect to online machine learning services.



You can select predictive models that are available in the application in the model list. When the model is ready for review, approve the model to replace the scorecard.

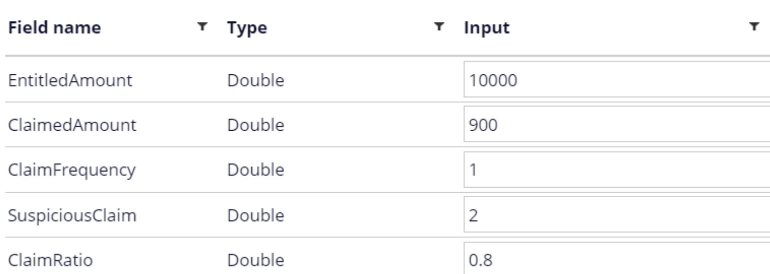


The fraud model now drives the prediction.

Claims fraud

Name	Type	Performance	Status
FraudH2O	Predictive model	——	ACTIVE

When you run the model with these input values, the model qualifies the claim as abnormal.



Field name	Type	Input
EntitledAmount	Double	10000
ClaimedAmount	Double	900
ClaimFrequency	Double	1
SuspiciousClaim	Double	2
ClaimRatio	Double	0.8

The model predicts the claim to be abnormal because the propensity value is above the threshold.

Outputs

Results

Result
abnormal

Propensity
0.8375238099694252

Predictors of the model include the claim data, such as location and claimed amount, but can also cover customer behavior data, such as the number of recent claims.

As an application developer, you can implement the fraud prediction to route claims based on the fraud risk calculated by the model. To use your fraud prediction, add the prediction to the case type.

Predictions
Manage predictions and associated objectives

Prediction	Objective	Data object
Predict Fraud Risk	Claims fraud	Claim

+ Add prediction

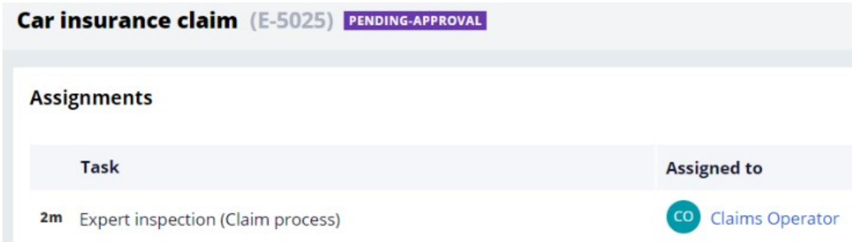
Next, in the **Decision** step in the **Detect fraud** stage of the life cycle, implement the prediction. Add the condition that only claims with a very low predicted fraud risk qualify for straight-through processing.

ClaimedAmount	is less than	100
and		
Probability	is less than	10

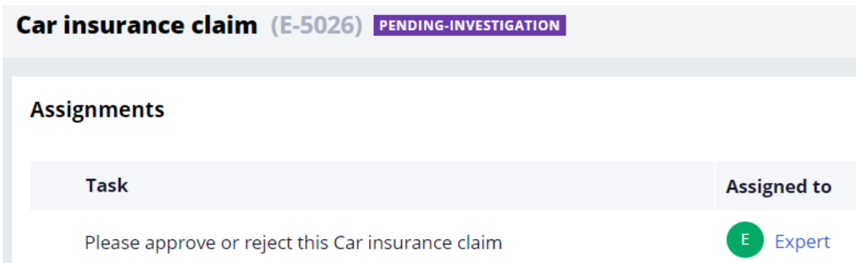
Replace the condition that routes a claim to a fraud expert based on the claimed amount with a condition that is based on the outcome of the fraud model and change the logical operator to generate the control group.

Probability	is greater than	50
or		
Is Random Check	is true	

When you run the same claim that previously qualified for straight-through the claim now disqualifies because the condition that fraud risk is very low is not met and the system consequently routes the case for regular processing.



When a claim with the same predictor values as previously tested in Prediction Studio is run, the system routes the case to the fraud expert.



This demo has concluded. What did it show you?

- How to create a case management prediction driven by a predictive model.
- How to use a prediction in a case type.

Predicting case completion

Description

Pega Process AI™ can help to distinguish regular from complex claims. Complex claims often escalate into a lengthy process, which is not only costly, but also leads to poor customer experiences.

Learn how to use Process AI to create an adaptive model to route complex cases to an experienced handler and leave many of the claims for straight-through processing. As the adaptive model learns from the outcome of each case, it becomes more accurate at predicting which claims to escalate, and in that way to self-optimize the process.

Learning objectives

- Create a prediction that predicts case outcomes
- Use the new prediction to route complex cases to an expert

Predicting case completion

Introduction

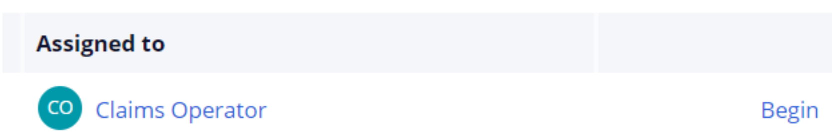
Pega Process AI™ can help to distinguish regular from complex claims. Complex claims often escalate into a lengthy process, which is costly and leads to a bad customer experience. The distinction lets you detect these claims early and address them at once.

Learn how to create a prediction that aims to identify cases that are likely to miss their deadlines and route them to a senior employee to handle them more efficiently and improve the customer experience.

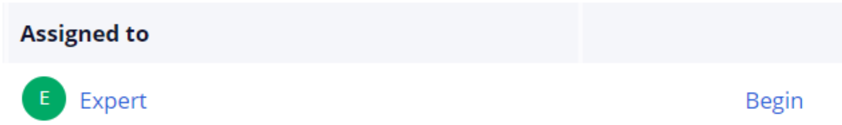
Transcript

This demo shows you how to use adaptive models to predict successful case completion.

U+ Insurance uses Pega Platform™ for case management. An incoming car insurance claim is routed to a claims operator, who approves or rejects the claim to resolve the case.



A case is escalated to an expert when the claim is not completed in the allotted time for regular processing.

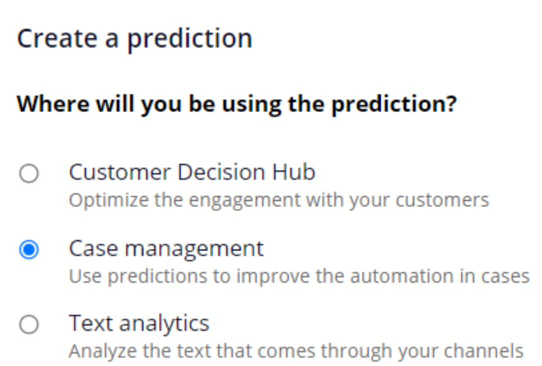


Claims that involve casualties can be very complex and are always routed to an expert instead of a regular claims operator as a precaution.

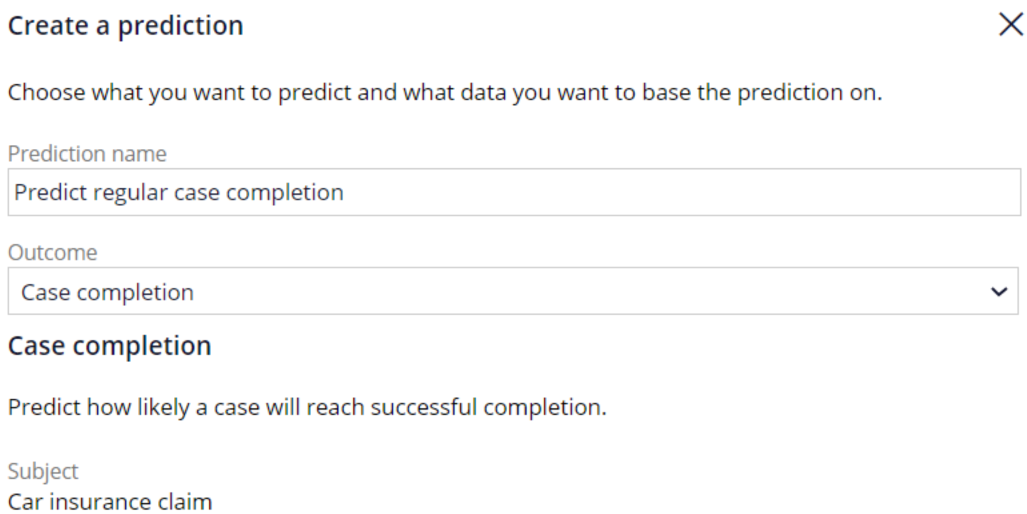
However, claims that involve casualties can often be resolved on time in the regular claims process. The experts consequently spend valuable time on relatively simple claims.

The primary stages of the Car Insurance claim case type comprise the regular claims process that leads to disbursement of the claimed amount. The alternative stages represent other process resolutions, such as the rejection of the claim.

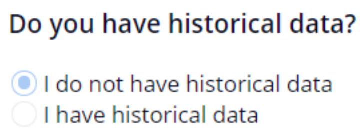
Process AI can help optimize the process by predicting the likelihood that a case is resolved before the deadline in the regular workflow and otherwise, route it to an expert irrespective of the cause of the complexity of the claim. This optimization requires a data scientist to create a case management prediction that calculates the propensity of whether the case is complex.



Process AI offers a wizard to create case completion predictions.

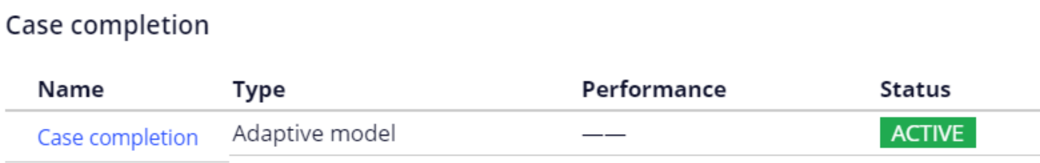


The subject of the prediction is the insurance claim. You do not need historical data, as the prediction is self-learning and uses adaptive models.

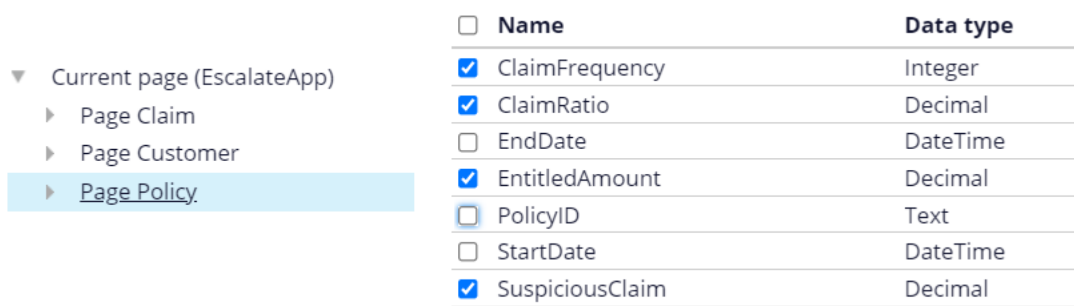


The target response label for the prediction is **Resolve**. It denotes a claim that is approved or rejected in the regular process. The alternative label, **Fail**, maps to resolution of a claim by an expert. The *CX* in the outcome names denotes a complex claim.

A best practice is to deselect fields that have no predictive value or are not allowed. Also, use a date field only if it reflects a time interval and not a definite date. An adaptive model drives the prediction.

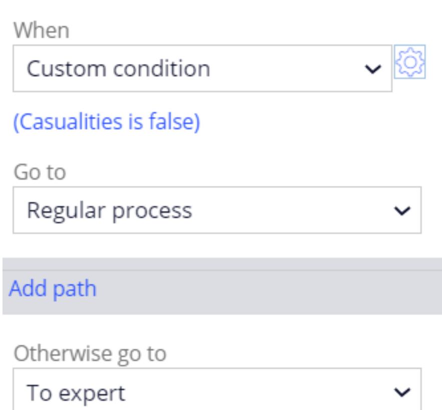


A best practice is to include many unrelated fields, including the claim properties. **Casualties** is one of the fields, and AI determines how well this predictor performs in predicting a complex case. Also include customer properties and behavior data such as the claim frequency of the customer.



The adaptive model learns from previous cases and automatically activates predictors that perform above a threshold and deactivates predictors when their performance drops over time. The prediction is ready to be implemented in the Car Insurance claim case type by an application developer.

In the **Claims process** stage of the case type, a decision component routes cases to a claims operator or, when the claim involved casualties, to an expert.

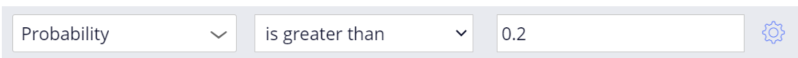


The condition requires an edit to meet the business requirement that the routing decision is based on the propensity that is calculated by the case outcome prediction. To use a prediction in a case type, add it to the settings.

Prediction	Objective
Predict regular case completion	Case completion

+ Add prediction

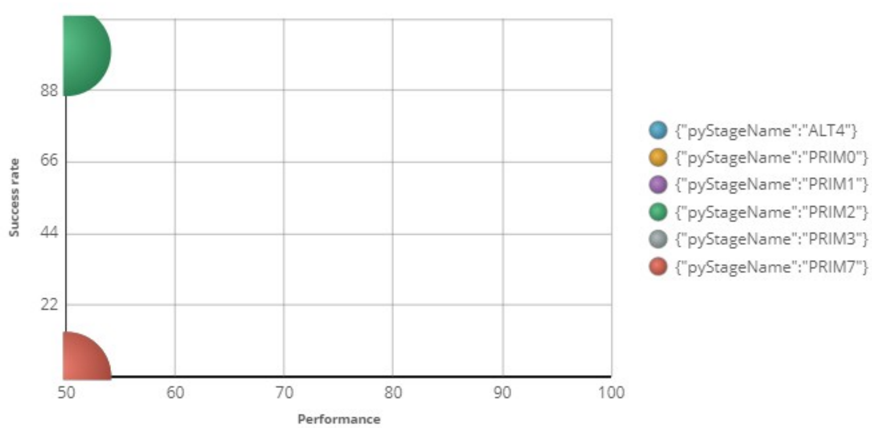
To qualify for the regular claims process, the propensity to resolve the claim without the involvement of an expert exceeds a threshold.



When a claims operator handles a claim the case status is **Resolved-Completed** or **Resolved-Rejected**, and the outcome of the case maps to the target label for the prediction. When a complex claim misses the deadline and is reassigned to an expert the outcome of the case maps to the alternative label for the prediction.

A claim with a low propensity to be resolved successfully in the regular workflow is immediately routed to an expert. The claim is routed to the regular workflow when the expert assesses the claim and does not consider it a complex case. This reassignment allows the adaptive model to learn from cases that are incorrectly routed to the expert.

An adaptive model is created for each primary and alternative stage in the case type. A decision request in a stage uses the model that is specific to that stage to calculate the propensity.



The models have no predictive power yet but self-optimize as more case outcomes are captured over time.

This demo has concluded. What did it show you?

- How to create a case completion prediction.
- How to implement a case completion prediction to improve efficiency.

Text analytics for email routing

Description

Humans can effortlessly interpret a single tweet but are unable to parse a large volume of information efficiently. Businesses are exploring ways to use machine learning to extract meaningful information from a large quantity of text messages. These insights help improve business performance and customer experience.

Learning objectives

- Explain text analytics
- Name a few practical applications where text analytics can be used
- Describe the role of artificial intelligence (AI) in text analytics
- Explain how text models for text categorization are trained

Text analytics

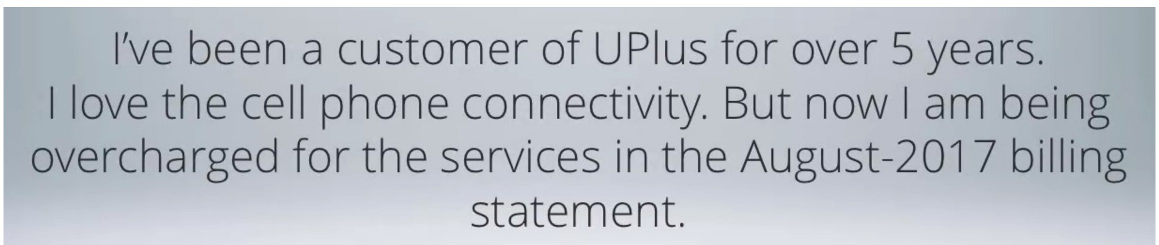
Introduction

A company wants all incoming emails to be automatically routed to the correct department. The company must find a way to process the text and extract the information for which it is looking. Text analytics is the most efficient and effective way to obtain information.

Transcript

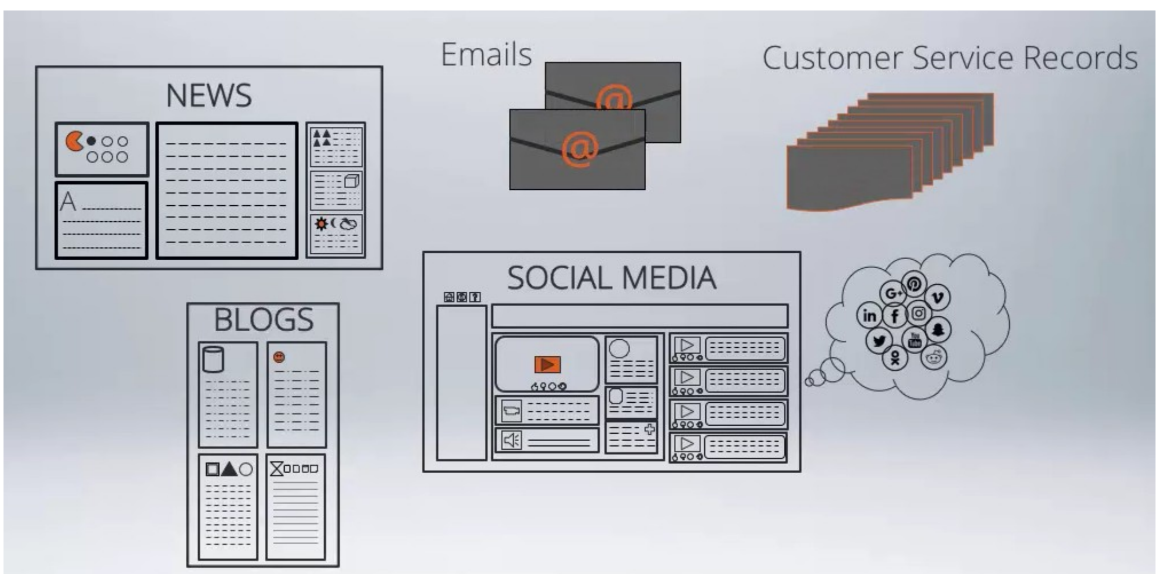
In this lesson you are going to learn what text analytics is and how it is used to help understand customer sentiments and needs.

Consider this message from a customer.



Humans have the ability to understand the implications of a piece of text. We can summarize it or elaborate on it or even re-phrase it and still preserve the original meaning. In fact, we do this very well. But can we do it on a large scale?

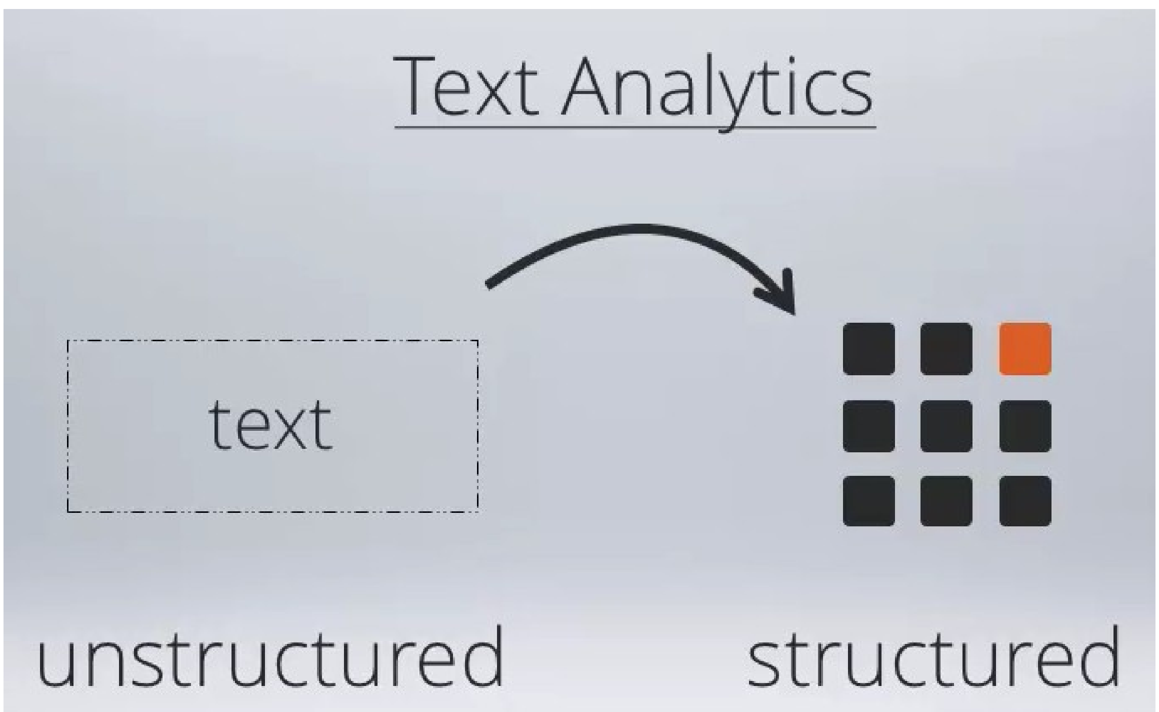
Today, digital sources such as news, blogs, social media and emails are generating more text than ever before.



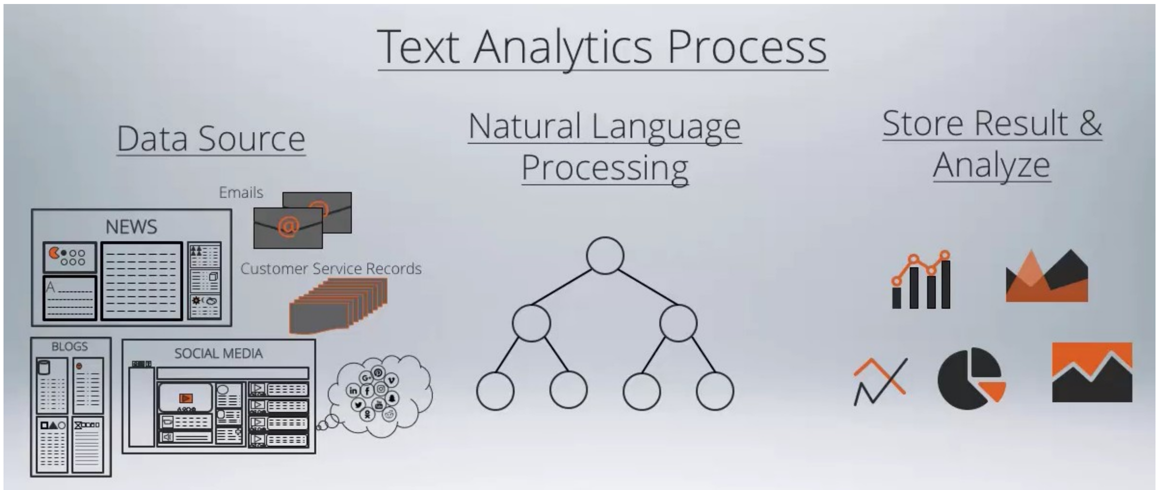
It’s not possible for humans alone to analyze and extract valuable information from such a large volume of text. Consider a company that has tens of thousands of customer service records in text form.

Let’s say the company wants all of the complaints to be automatically routed to the right department. Then it wants a report that breaks down the complaints by department. The company also wants to use these records to understand how satisfied its customers are about various products and services. The company must find a way to process the text so that the information it’s looking for can be extracted. Text analytics is the most efficient and effective way to do this.

Text Analytics is the process of deriving high-value information from text.

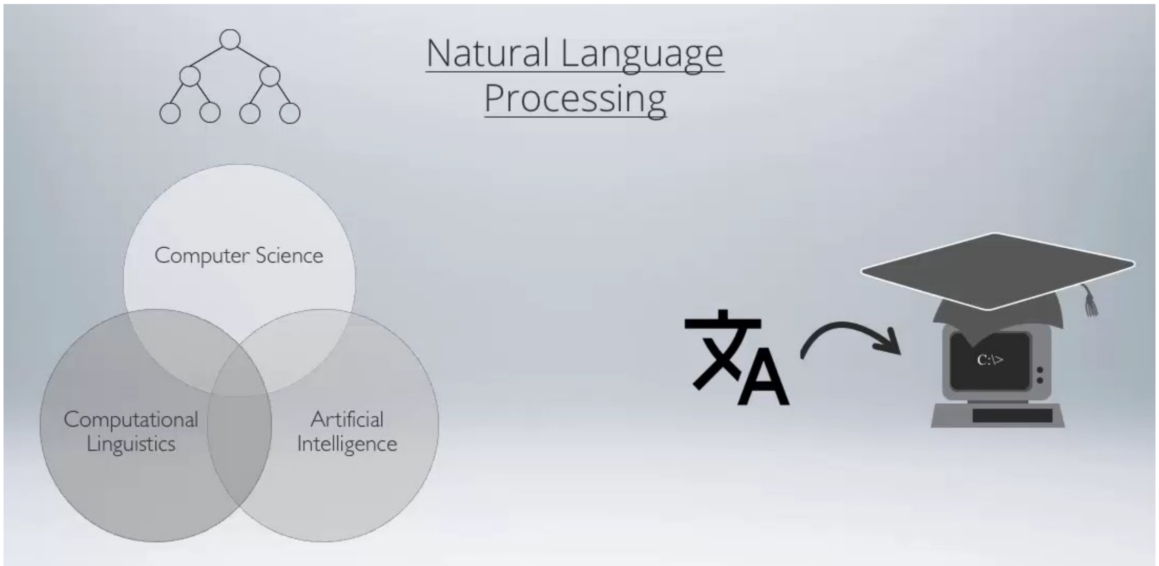


It involves converting unstructured text into structured data that can be analyzed using statistical methods. There are three main steps involved in text analytics.

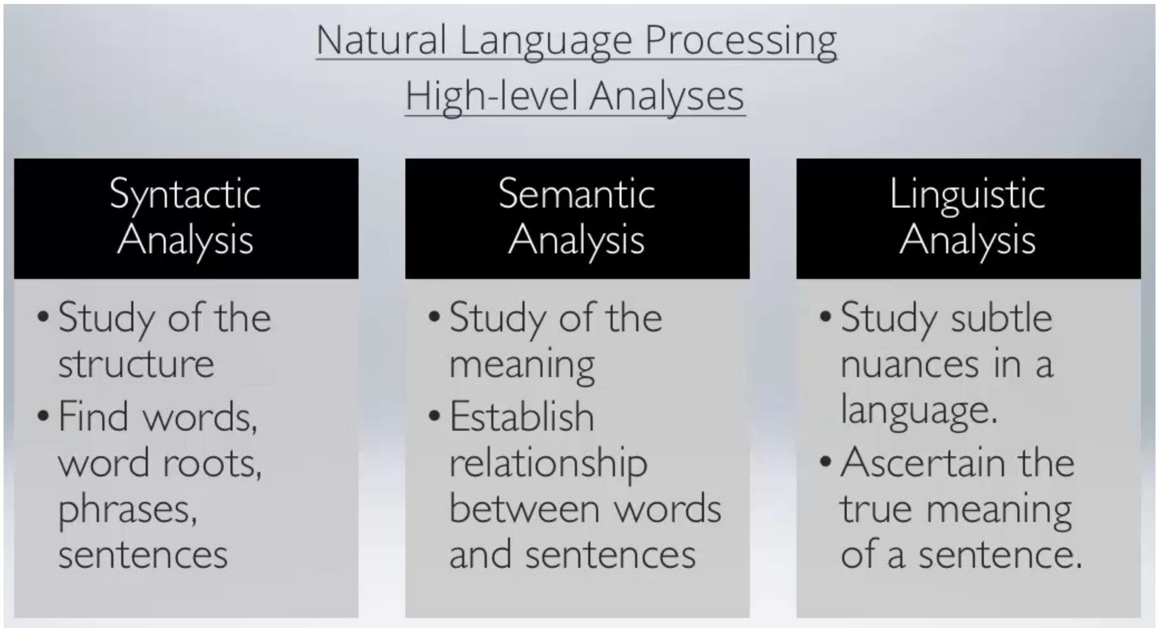


First, data is fetched from a source, such as social media, emails or customer service records. Second, natural language processing techniques are applied to extract certain attributes from the text and present them as structured data. And third, the extracted structured data is stored and analyzed.

Natural language processing is a field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and human language.

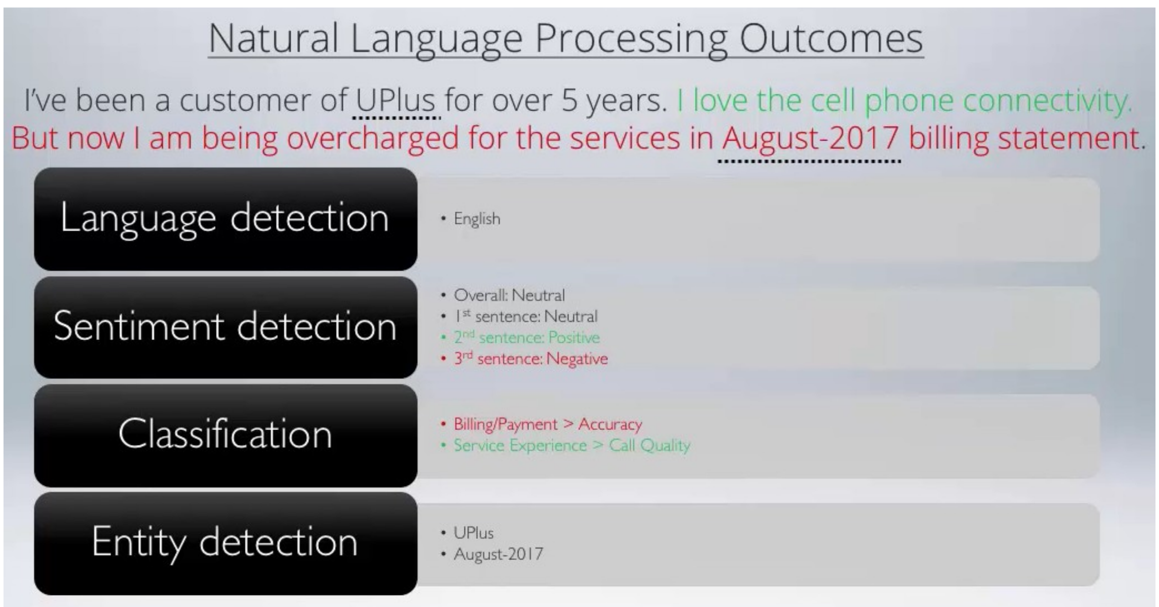


In particular, it is concerned with programming computers to fruitfully process large volumes of natural language text. At a high level, the following analyses occur during natural language processing.



Syntactic analysis analyzes the structure of text and recognizes the parts of speech, such as nouns, verbs, adjectives, etc. Semantic analysis establishes the meaning of a piece of text. So the relationship between words and sentences is analyzed. In the linguistic analysis phase, the nuances of a language are considered to ascertain the exact meaning of the text.

Here are the outcomes of natural language processing.



The language of the text is detected. The sentiment is detected as positive, negative or neutral. Sentiment is the general attitude expressed towards a subject. The overall sentiment as well as the sentiment per sentence is reported.

The text is classified into one or more pre-defined categories, such as the business functions in a company. For each category detected, the sentiment of the corresponding sentence is also reported. Entities refer to the proper nouns found in the text such as names of people, places, dates and times, organizations, etc. This helps in establishing the subject of the text.

You have now reached the end of this lesson which provided an overview of text analytics and how it works.

Email routing

Introduction

A company has an email channel where its customers can send any email — from service requests to compliments to sales inquiries — to the product support team. This email variety makes it difficult to provide each customer with a prompt and personalized response. Pega Infinity™ uses AI-powered text analytics to perform intelligent email routing.

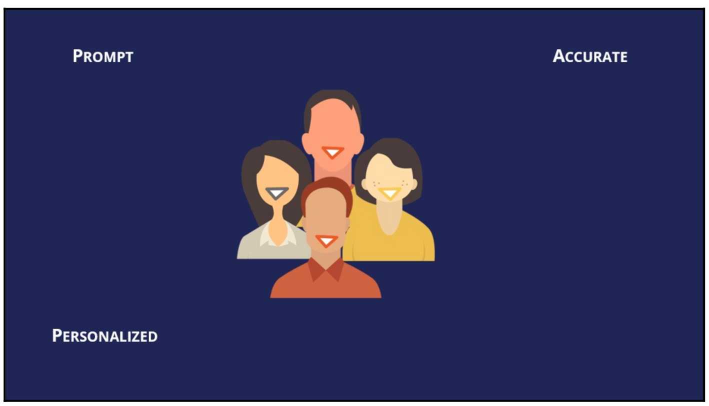
Transcript

In this video, you will learn how Pega Infinity™ uses AI-powered text analytics to do intelligent email routing.

Customer satisfaction is a reflection of what a customer expects from a company vs. what they experience from the company.

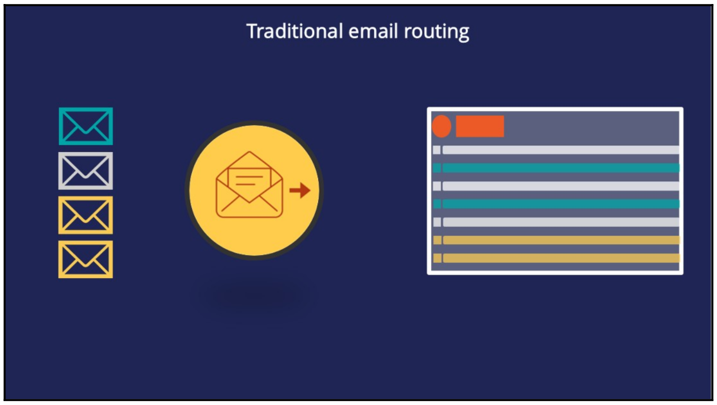
Meeting, or even better, exceeding customer expectations means addressing their service requests and complaints promptly, accurately, and with personalized solutions.

Doing this will ensure the customer has a great experience.



Assume a company has an email channel in which its customers can send any type of email —from service requests to compliments to sales inquiries—to the product support team.

These emails are often routed to the same container and are uncategorized. This makes it difficult to provide each customer with a prompt and personalized response.



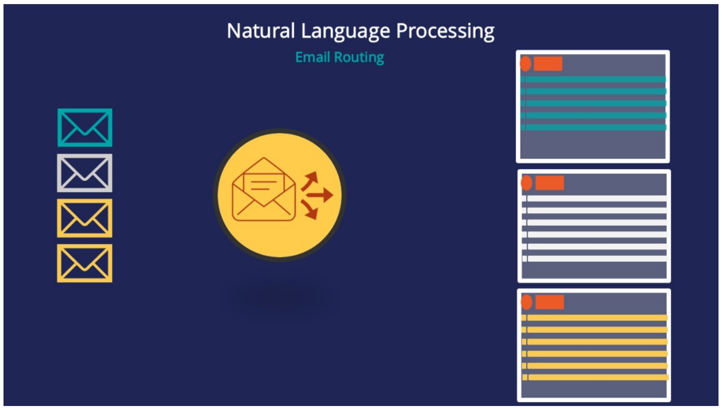
In Pega Infinity, this issue can be addressed by routing the emails using artificial intelligence.

Pega Infinity uses AI-powered Natural Language Processing to detect the topic of an email and route the email to the appropriate container.

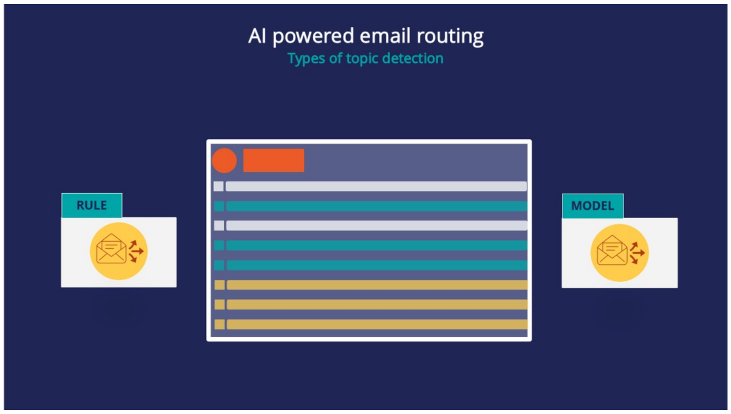
Consider emails from customers requesting an account address change, making a compliment, or requesting a new credit card.

With the help of AI-powered text analytics, Pega Infinity is able to read and understand the content of each email and route it appropriately.

This means customer service representatives can be alerted to any account-related service requests and resolve them quickly.



Email routing is done using the topic detection mechanism. The two types of topic detection are rule-based and model-based.



In rule-based topic detection the routing is based on the rules configured in the email channel.

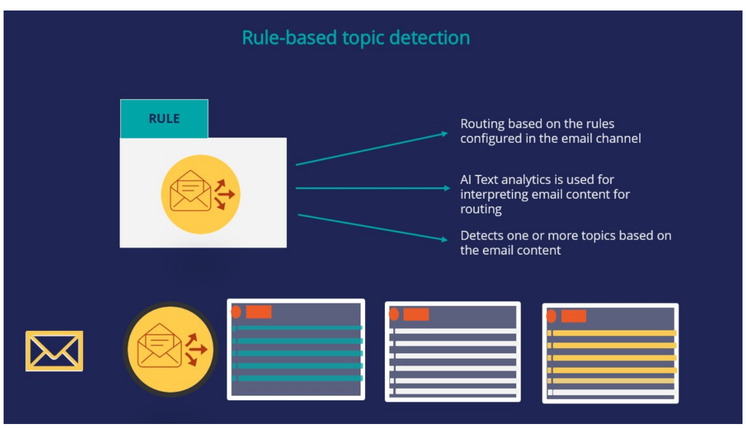
AI-powered text analytics is used to detect the topic of the email, and the channel rules route it to the right container.

This type of topic detection may detect one or more topics if the email contains words associated with more than one topic.

Let's consider a service request email from a customer. The email content is analyzed and routed to the right container.

If an email from another customer contains words that are associated with two topics, the rule-based topic detection detects both topics.

The email can then be routed to two different containers depending on how the channel rules are configured.



In model-based topic detection the routing is based on AI models built by a data scientist using machine learning.

AI powered text analytics is used to detect the topic of the email. Building these models requires a training data set and a test data set.

The data sets consist of a list of emails and the associated topic for each email.

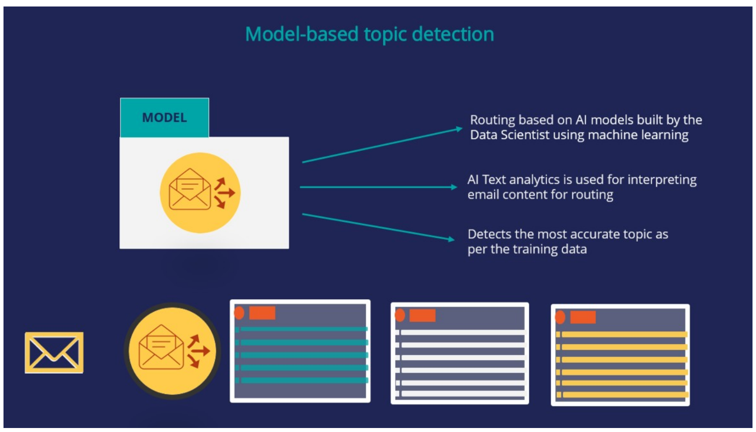
This type of topic detection identifies the most accurate topic based on the AI model and training set used by the data scientist.

Let's consider the same service request email from the customer. The email's content is analyzed and routed to the right container.

If an email from another customer contains words associated with two topics during the training of the models ...

...the model-based topic detection detects both topics but typically with a different accuracy factor.

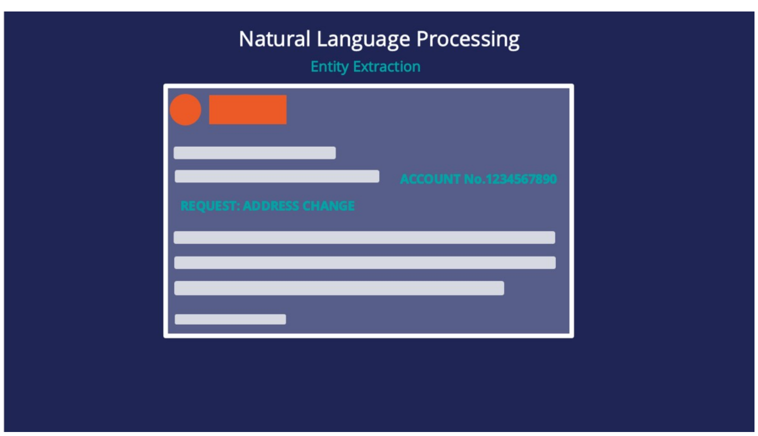
In this case, the topic with the highest accuracy factor is chosen.



Pega Infinity also enables you to extract entities from an email.

This means that when an email is sent, certain entities such as account number, email address, street address, etc. can be automatically detected and extracted.

This allows certain emails to be automatically processed or given priority.



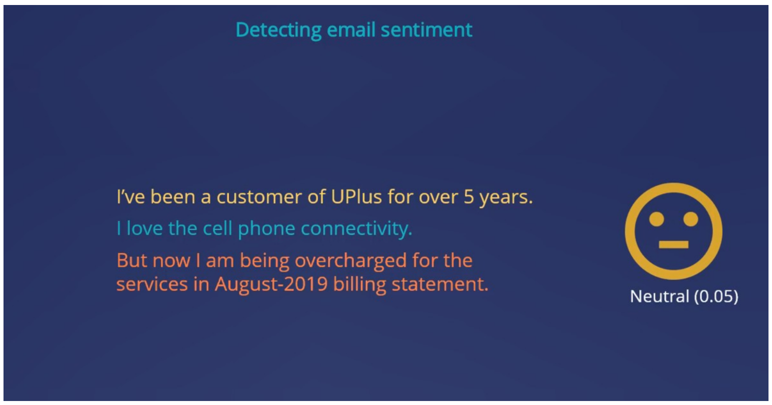
Besides Topic detection and Entity Extraction, Pega Infinity uses its AI-powered text analytics to enable you to detect the sentiment of an email based on its content.

Suppose a customer sends the following email to customer service. The email's content is a combination of different types of sentences.

The first sentence has a neutral sentiment. In the second sentence, the customer expresses his satisfaction with the connectivity, so it has a positive sentiment.

The third sentence is negative, as the customer is complaining about an overcharge.

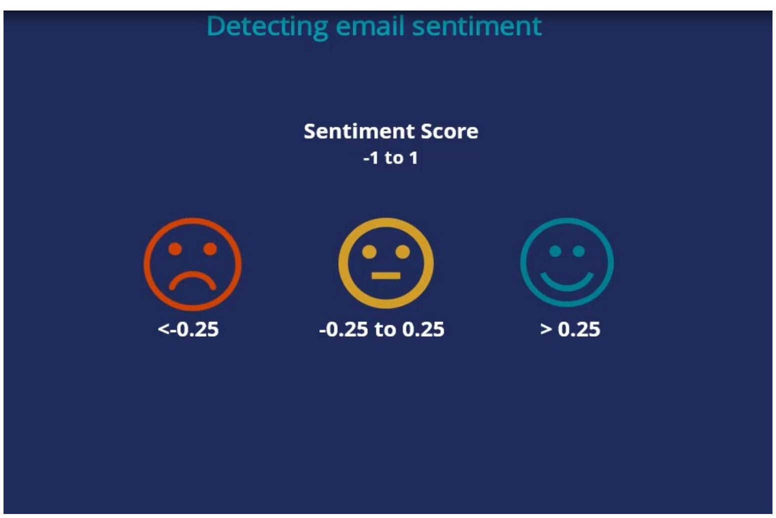
The overall sentiment of the email is determined based on the weight of the content sentiments. In this specific example the overall sentiment is neutral with a sentiment score of 0.05.



The sentiment score is a value between -1 and 1.

In the out-of-the-box configuration, a sentiment score <-0.25 results in a Negative sentiment, a sentiment score between -0.25 and 0.25 results in a Neutral sentiment ...

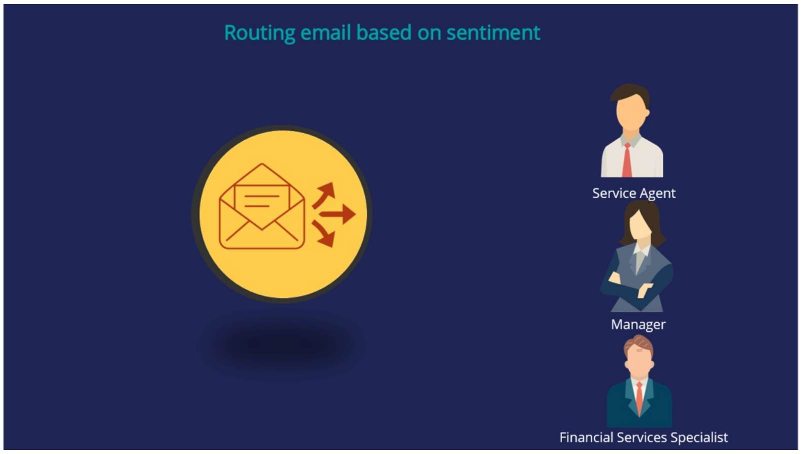
... and a sentiment score above 0.25 results in a positive sentiment.



Once the email sentiment is detected, you can configure the email channel to route a specific topic with a specific sentiment to a specialized agent for a quick and personalized response.

For example, you could route an address change with a neutral sentiment to a Service Agent, a complaint email with a negative sentiment to a Manager ...

... and a credit card inquiry with a negative sentiment to a Financial Services Specialist.



In summary, Pega Infinity's AI-based email routing capability enables customer service representatives to be more productive, reduces request processing time ...

... and improves the customer experience by providing prompt and personalized service.

Training a topic model to improve email routing

Introduction

U+ Bank uses Pega Customer Service™ to route incoming emails to the appropriate department based on the topic of the email. For several use cases (for example, an address change), emails are routed based on keywords that are detected in the message. To improve the email routing, learn how to train the text prediction with a data set that contains classified messages.

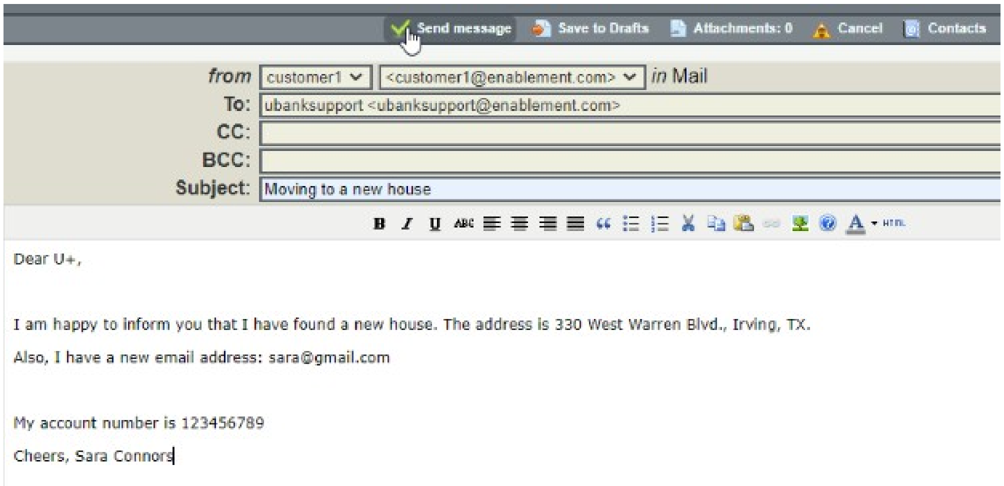
Transcript

This demo shows you how to train a text prediction to improve email routing.

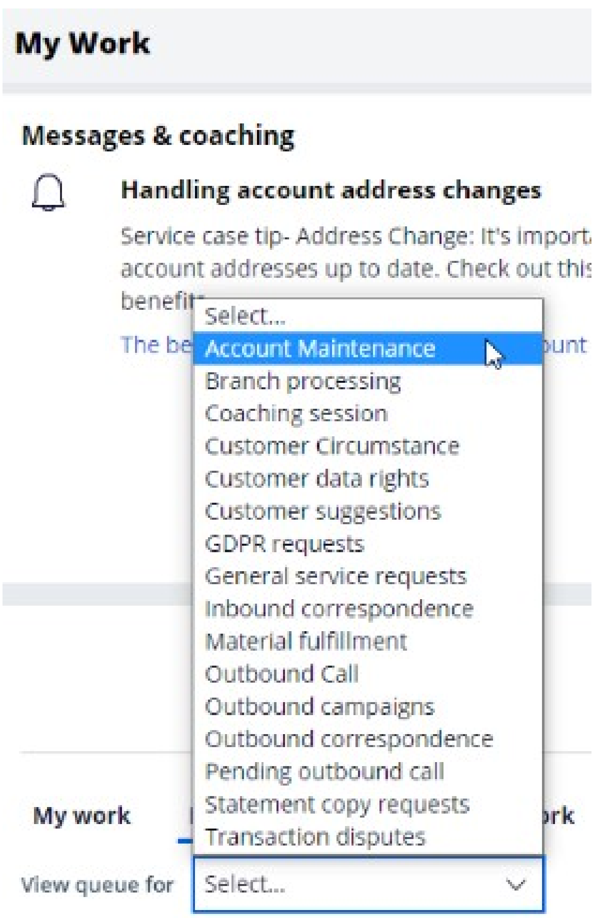
U+ Bank uses machine learning to route inbound messages in the email channel to the appropriate department based on the topic of the email.

A text prediction that aims to detect the topic of the message drives the routing.

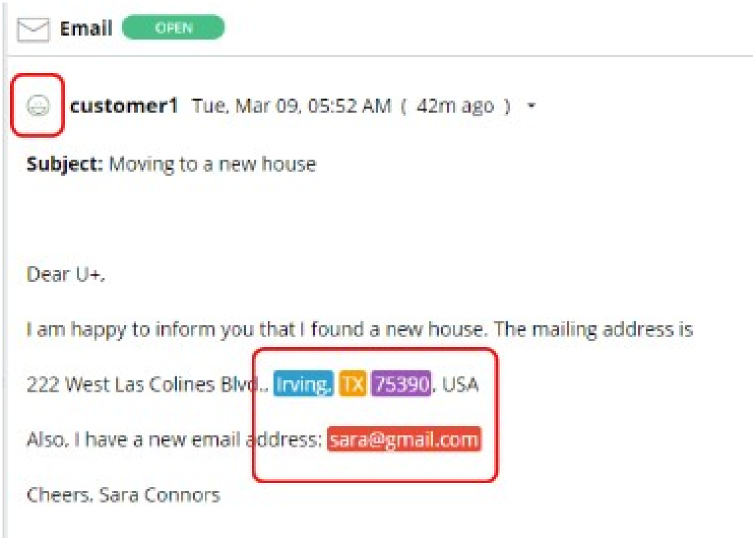
For example, when Sara writes an email to inform U+ Bank that she has moved to a new house, the text prediction detects an address change as the topic.



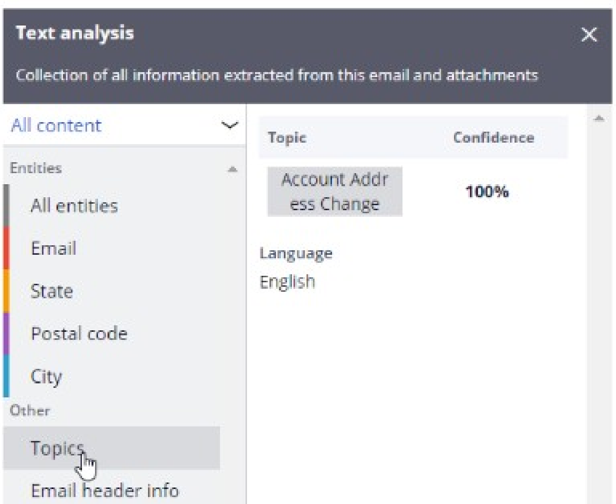
The **Account maintenance** department receives all emails where an address change is detected as the topic of the message.



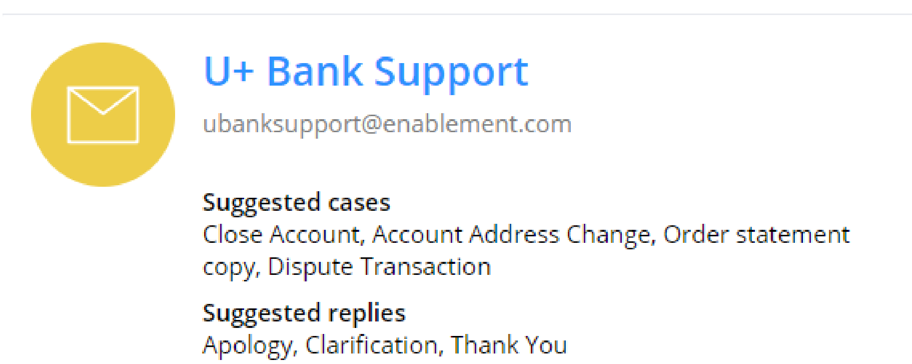
The text prediction also detects entities such as a ZIP Code and an email address and the overall sentiment of the message.



Notice that the topic Account Address Change is detected with 100% confidence.



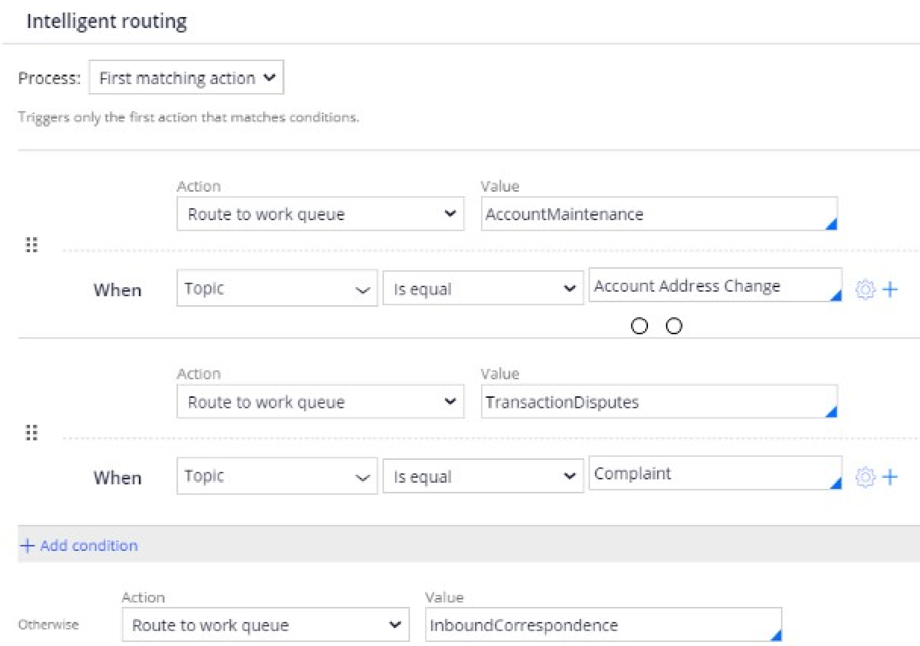
The intelligent routing is set up in the email channel, in App Studio.



An email is routed to the work queue of the Account maintenance department when the detected topic is an address change.

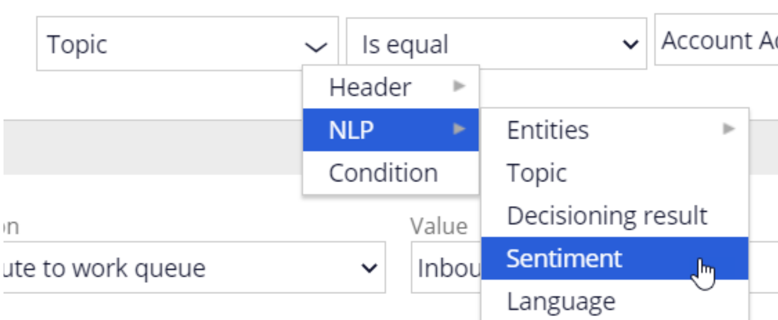
Complaints are routed to the **Transaction Disputes** department.

If the address change and complaint topics are not detected, the email is routed to a default work queue.

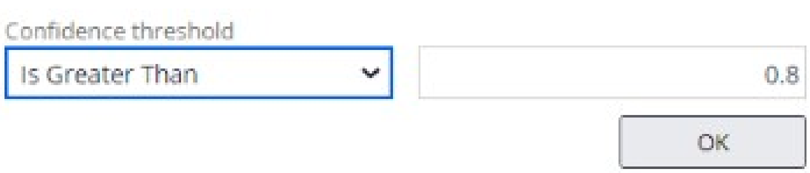


Note that in the **When** condition of the email routing, you use the outcome of the text prediction.

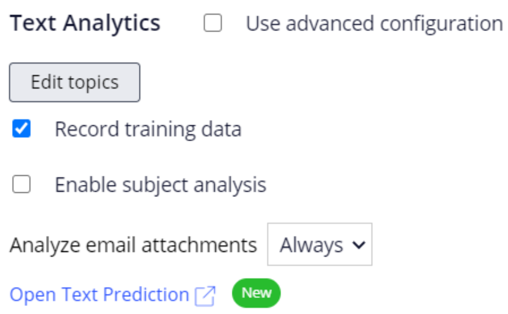
Optionally, you can also use the sentiment of the email in the routing conditions.



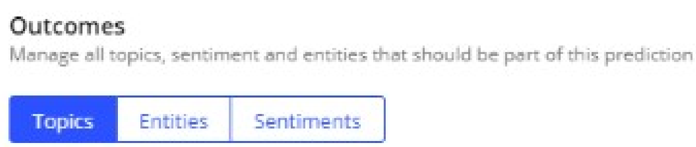
You can set a threshold for the confidence with which a topic must be detected to trigger routing.



Every channel is associated with a text prediction.



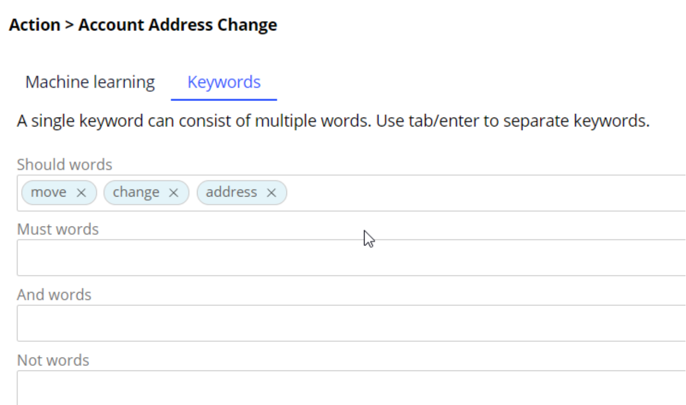
A text prediction is driven by predictive models that detect topics, entities, and sentiments.



Many entity extraction models, and a sentiment model, are available out of the box.

For topic detection, Prediction Studio supports keyword models as well as machine learning models based primarily on training data.

A keyword model uses **Should words**, **Must words**, **And words**, and **Not words**.



If any of the **Should words** appear in a piece of text, topic detection assigns that text to the corresponding topic. To achieve accurate results, create an exhaustive list of **Should words**.

Only if all **Must words** appear in a piece of text, will the topic detection assign that text to the corresponding topic.

Use **And words** to distinguish between similar topics while using identical **Must words**.

If a **Not word** appears in a piece of text, the text is not assigned to the corresponding topic.

You can test the output of a prediction on a sample message.

Two topics are correctly detected: an address change and a complaint.

Notice that the confidence score for both topics is **1**. The keyword model performs a Boolean match based on the presence or absence of words and detects the topic with absolute certainty.

Topic detection

Granularity
Document

Analysis
Rule

Topic	Sentiment	Sentiment score	Model name	Model type	Confidence score
Action > Account Address Change	Neutral	0	pyInteractionTaxonomy	Pega NLP	1.00

It is highly recommended to use a machine learning model for topic detection. It determines the confidence score based on evidence.

A machine learning model is based on training data that consists of categorized texts.

You can get training data from two sources. The first is training data accumulated through the application that is running in production.

When a customer service representative corrects the topic of an incoming email, this change is added to the training data.

After the messages are reviewed and approved, they are used to rebuild the topic model.

Also, you can choose to import data, provided you have accumulated data in the past.

The file must contain the message and the associated topic.

	A	B	C
1	**content**	**result**	**type**
20	Effective 11/28/17 please change mailing address to	Action > Account Address Change	Test
21	Please amend the mailing/billing address to the follov	Action > Account Address Change	Test
22	Confirm - mailing address: 1430 E 33rd St., Signal Hill	Action > Account Address Change	Test
23	Effective 8/28/17, please change the insured's mailin	Action > Account Address Change	Test
24	Per our insureds request, please issue the following c	Action > Account Address Change	
25	Effective 8/25/17, please correct the mailing address	Action > Account Address Change	
26	Effective 8-16-17 please amend the mailing address a	Action > Account Address Change	
27	Please amend the mailing address.	Action > Account Address Change	
28	Effective 8/28/17 please change the mailing and billir	Action > Account Address Change	
29	AMEND insured's MAILING ADDRESS to read:	Action > Account Address Change	
30	Effective: 09/01/2017 Please amend mailing address	Action > Account Address Change	
31	Please update mailing address per the below:	Action > Account Address Change	
32	We have received a request to change your mailing a	Action > Account Address Change	
33	Please change mailing address to read: 197 Warren A	Action > Account Address Change	
34	Please change the mailing address as per the attache	Action > Account Address Change	
35	Please amend mailing & billing address to 3253 Phillip	Action > Account Address Change	

Notice that training data for the address change and the complaint topics are in the pending state.

You can now rebuild the models.

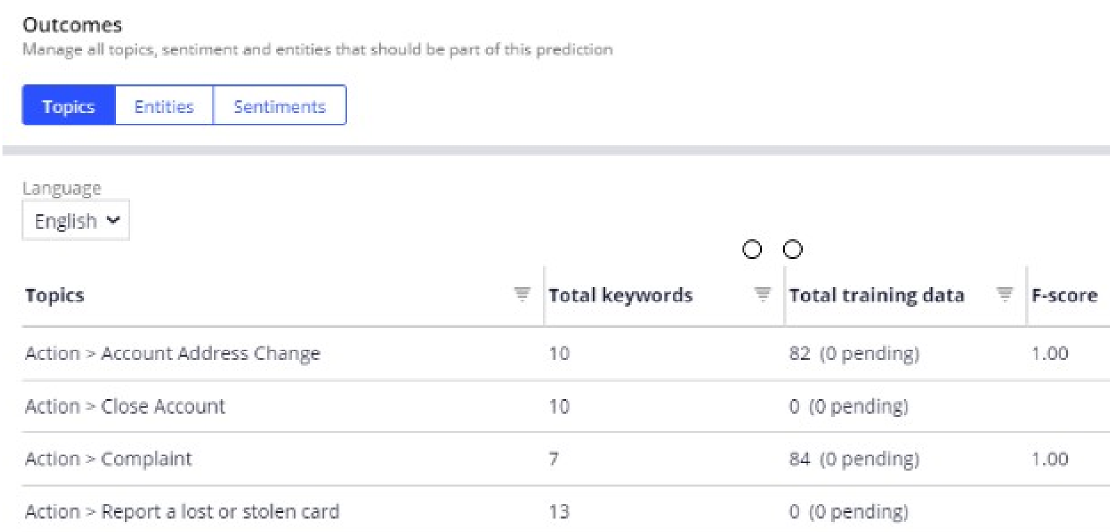
You can select individual models or rebuild all models across model types and languages.

While training is in progress, you have the option to cancel.

When the process completes, you can view the latest model report on the **Models** tab.

The report contains the validation data, the confusion matrix, and the score sheet.

The topic detection now uses machine learning models based on the training data and the keywords provided for the address change and complaint topics.



Topics	Total keywords	Total training data	F-score
Action > Account Address Change	10	82 (0 pending)	1.00
Action > Close Account	10	0 (0 pending)	
Action > Complaint	7	84 (0 pending)	1.00
Action > Report a lost or stolen card	13	0 (0 pending)	

The **Should words** and **Must words** act as positive features for matching text to a topic, while the **Not words** act as negative features.

But the training and testing data have the greatest impact on your machine learning model, while keywords have a smaller impact.

Once the models are rebuilt, you can test the prediction.

Notice that the topics are detected with a confidence score below one.

In this case, the topic message is recognized as a complaint with high confidence. The address change topic is detected, but with a lower confidence score.

When multiple topics are detected in a message, comparison of the confidence scores allows selection of the one with the highest priority.

Topic	Sentiment	Sentiment score	Model name	Model type	Confidence score
Action > Complaint	Neutral	-0.25	U+ Bank customer support	Pega NLP	0.75
Action > Account Address Change	Neutral	-0.25	U+ Bank customer support	Pega NLP	0.25
Action > Report a lost or stolen card	Neutral	-0.25	U+ Bank customer support	Pega NLP	0.00

The other outcomes of the text prediction are the entities detected, such as an email address and the sentiment of the message.

Output

I am happy to inform you that I found a new house. The address: 330 West Warren Blvd., Irving. Also, I have a new email: sara@gmail.com For the third time, I have noticed an invalid transaction. The amount is small, but could you please check?

Sentiment indication

Positive Negative Neutral

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

- How Pega Customer Service routes incoming emails to the appropriate department based on the topic of the email
- How text predictions work to predict the topic and sentiment of a message and detect entities
- How to train a topic model and use machine learning to identify the topic correctly