Pega Process AI essentials

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: Pega Process AI Essentials

Product: Pega Platform™ 8.7

URL: https://academy.pega.com/mission/pega-process-ai-essentials/v2

Date: 22 December 2021

Contents

Pega AI overview....4
    Predictive models drive predictions....5
    Prediction Studio....9
Pega Process AI overview....18
    Pega Process AI overview....19
Predicting fraud....23
    Predicting fraud....24
    Using machine learning services....30
Creating predictive models....37
    Predictive models....38
    Building models with Pega machine learning....42
    Importing predictive models....50
    Using machine learning services....55
    Model transparency....62
MLOps....64
    MLOps process....65
    Placing a predictive model in shadow mode....69
Adaptive analytics overview....72
    Adaptive analytics....73
    Predictors and outcomes of an adaptive model....75
    Advanced settings of an adaptive model....79
    Adaptive model outputs....81
Predicting case completion....82
    Predicting case completion....83
Monitoring adaptive models....88
    Regular monitoring of adaptive models....89
    Inspecting adaptive models....91
    Exporting historical data....98

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.

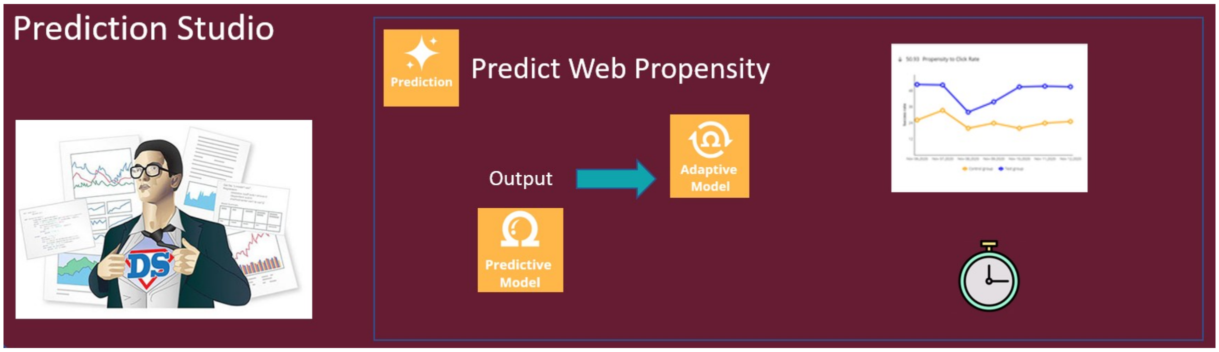
Comparison of the control group and the model propensity-based group allows you to measure the lift in a success rate that the AI generates, an important business metric.

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.

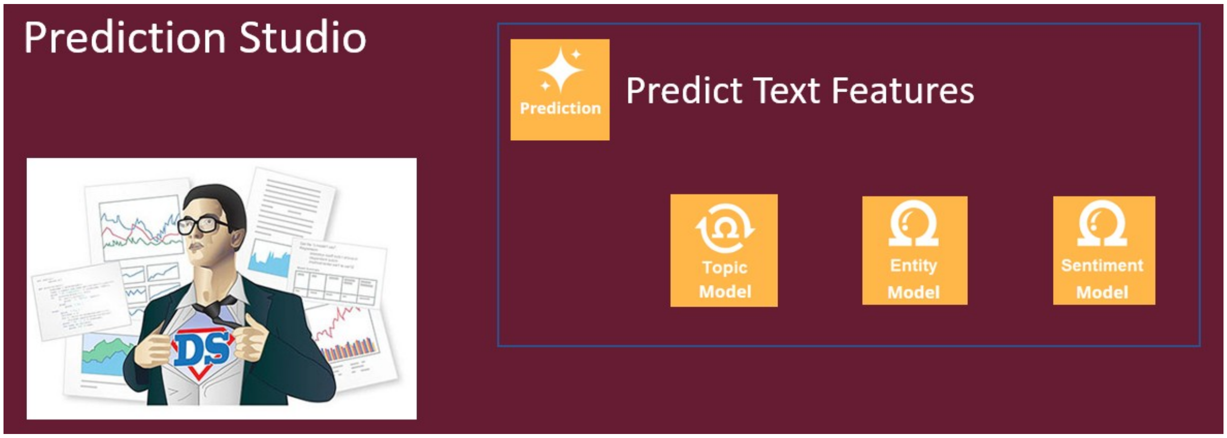
Pega Customer Service uses Text analytics predictions that are distinctly different from both case management predictions and Customer Decision Hub predictions.

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.

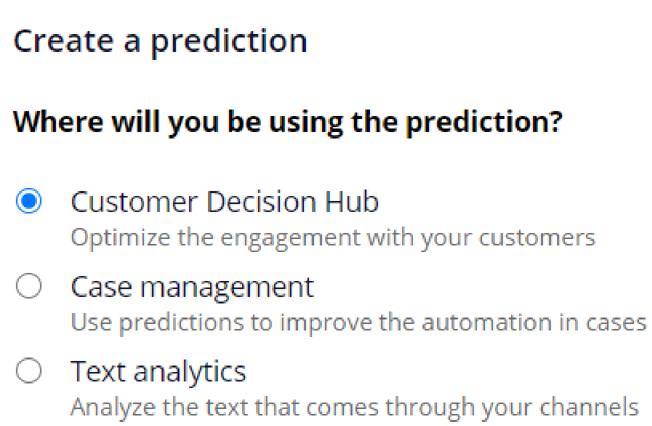
Prediction Studio

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 model.

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

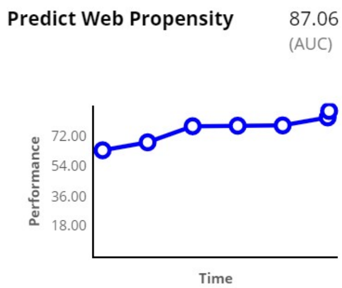
Text categorization models can detect the topic of a message and the sentiment of the author. **Text extraction** models identify entities such as an email address, an account number, or a city.

Text categorization

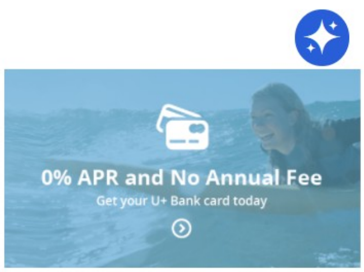
Text extraction

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.



Standard card

0% APR and no annual fee

Learn more

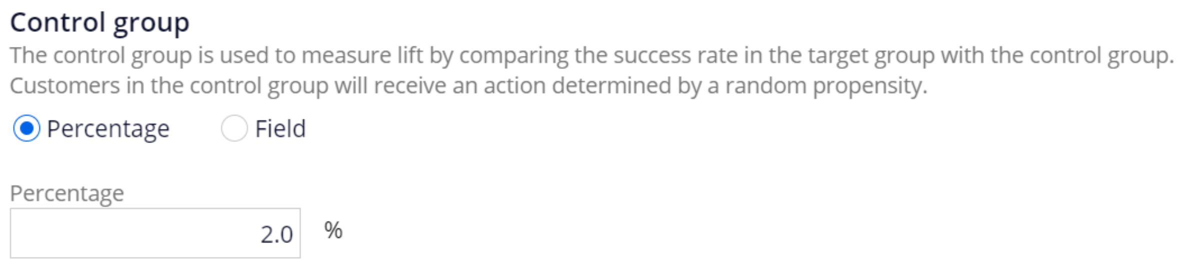
When a customer is eligible for multiple credit cards, the prediction calculates the propensity of receiving a positive response from the customer for each card. Customer Decision Hub decides which credit card to offer based on business rules, interaction context, and predictions.

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 test group.

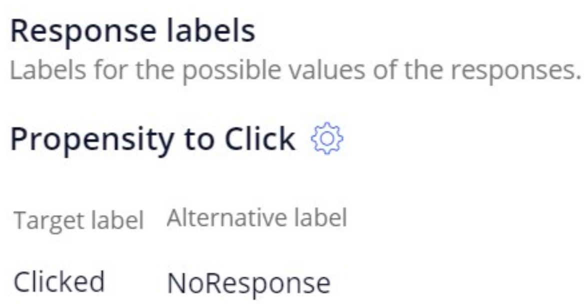


Customers in the test group receive the offers that they are most likely to accept, based on the propensity that the prediction calculates for each customer. The purpose of the control group is to calculate lift by comparing the success rate in the control group with the success rate in the test group. The random offers also allow predictive models to continuously explore all actions.

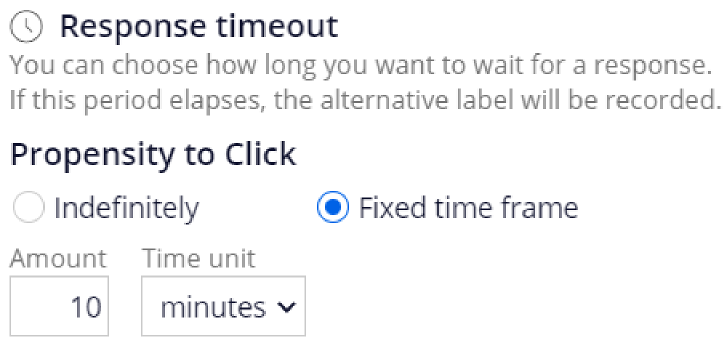
Based on the lift, you can determine the effectiveness of your prediction, for example, in increasing conversion rates. The control group is typically defined as 2% of all customers, but this can be changed.

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**.

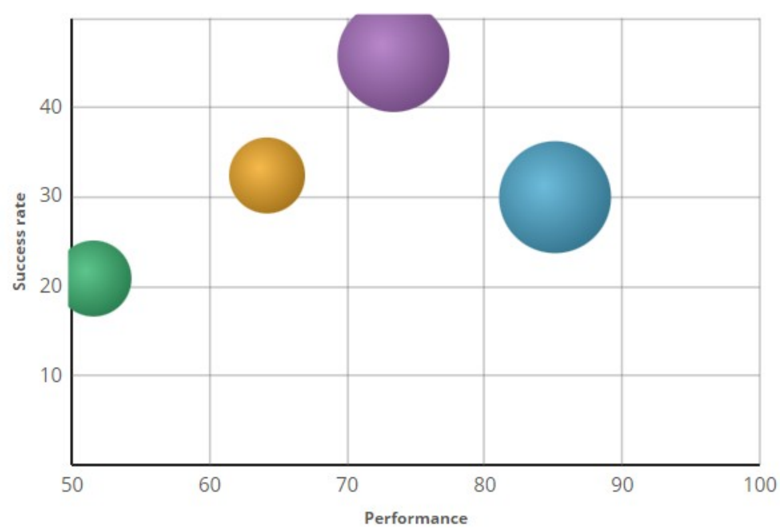


The **NoResponse** response can be captured on request or automatically depending on the response time-out setting. The response time-out defines how long to wait for the customer to respond to your offer. After the specified amount of time elapses, the system automatically records the alternative outcome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



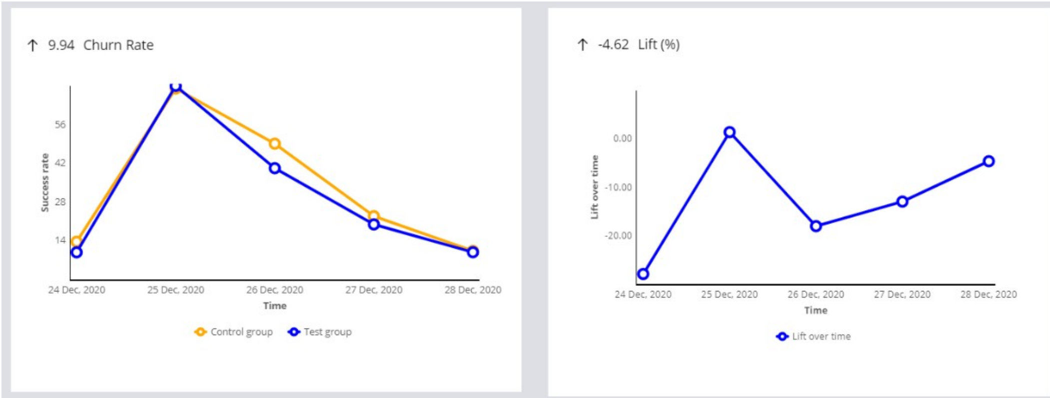
This setting depends on the use case. When predicting a click on a web banner, you typically set it to 30 minutes or less, but in an outbound email offer, a waiting time of several days is more appropriate.

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.



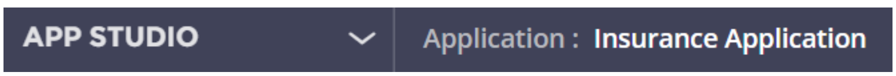
The models are represented by colored circles. The size of the circles indicates the number of responses captured by the model.

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).

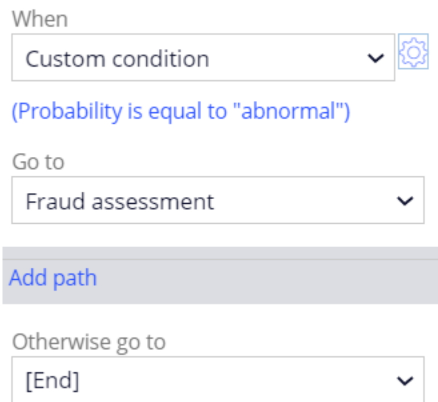
Case management predictions support decisions in a case type.



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.

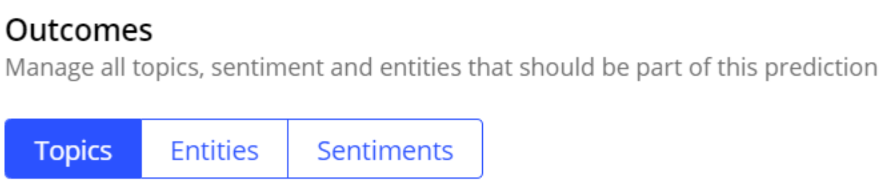
Based on the condition, a case is routed to a fraud expert when the prediction flags the claim as abnormal.



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.

Data

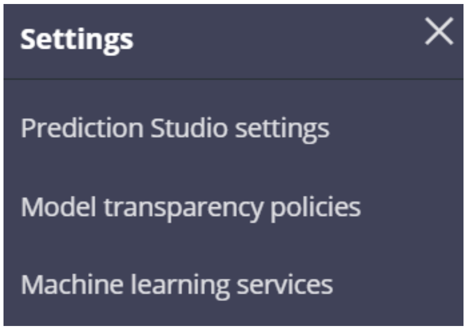
Data sets

Resources

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.

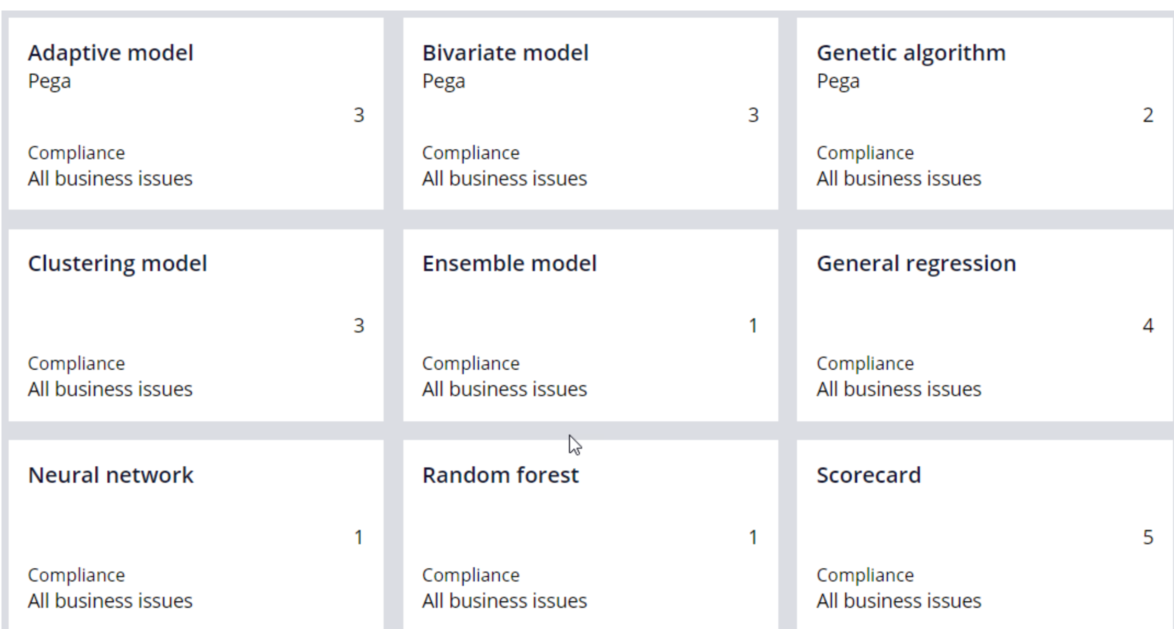


Also, you can review company policies regarding the transparency thresholds for different business issues.

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 setting of the business.

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

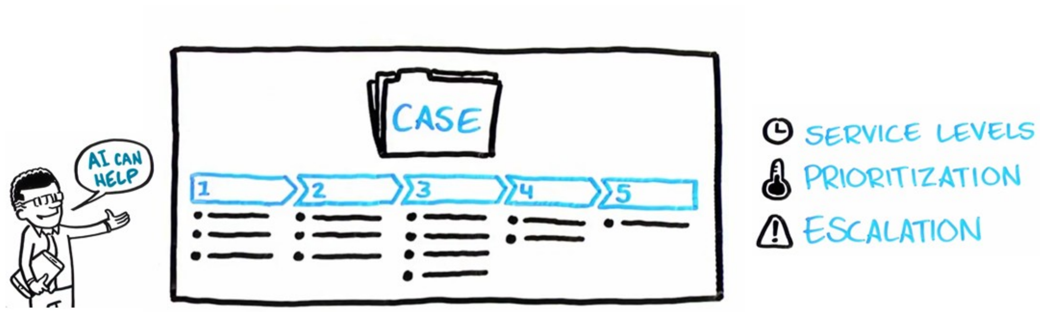
Introduction

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.

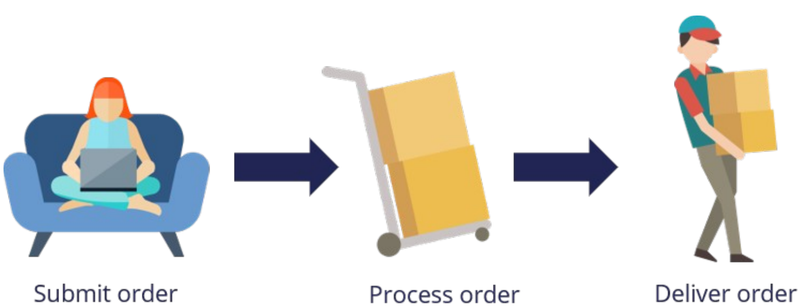
Transcript

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.

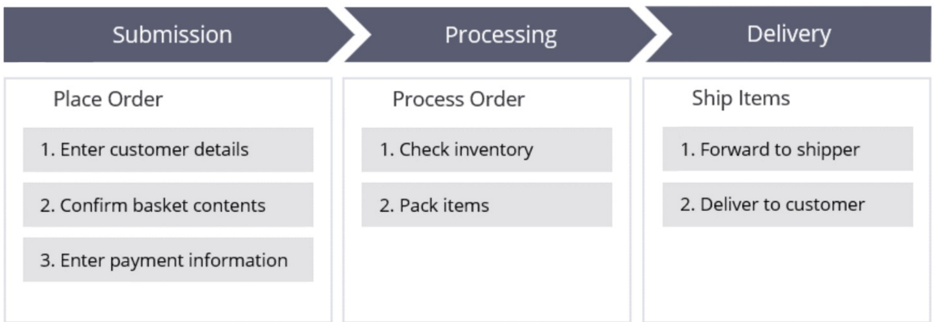


An Pega Platform application that models the online order process follows the same sequence as a series of stages. A **case type** is the abstract model of that process.

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

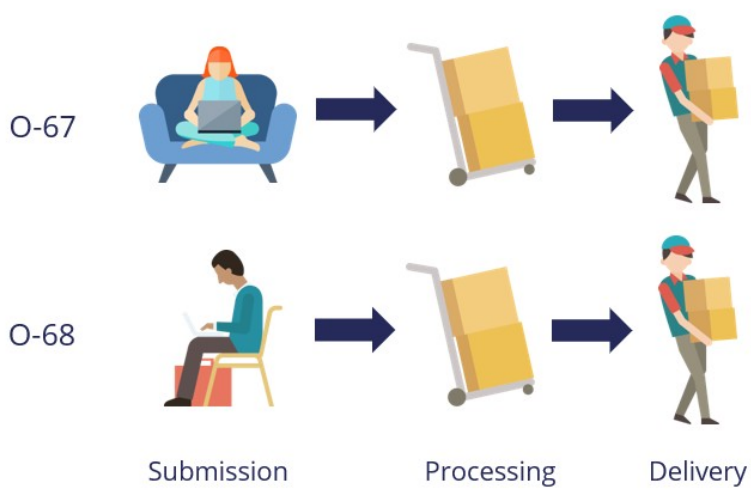
The **case life cycle** for a case type helps to visualize the work to complete as part of a business transaction.

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.

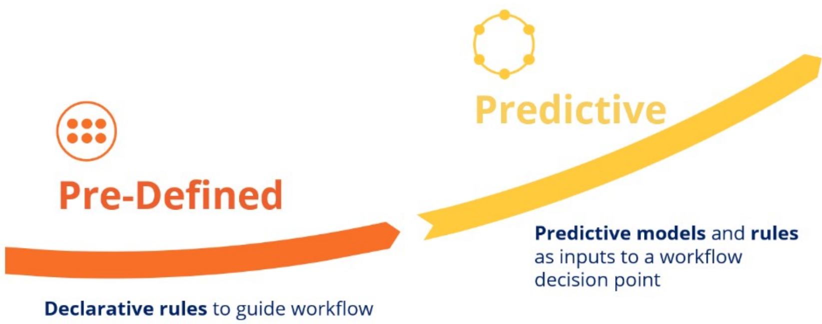


A case type can use declarative rules to manage the workflow, for example, to confirm that the order contains a valid shipping address or the order amount threshold to qualify for free shipping.

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.

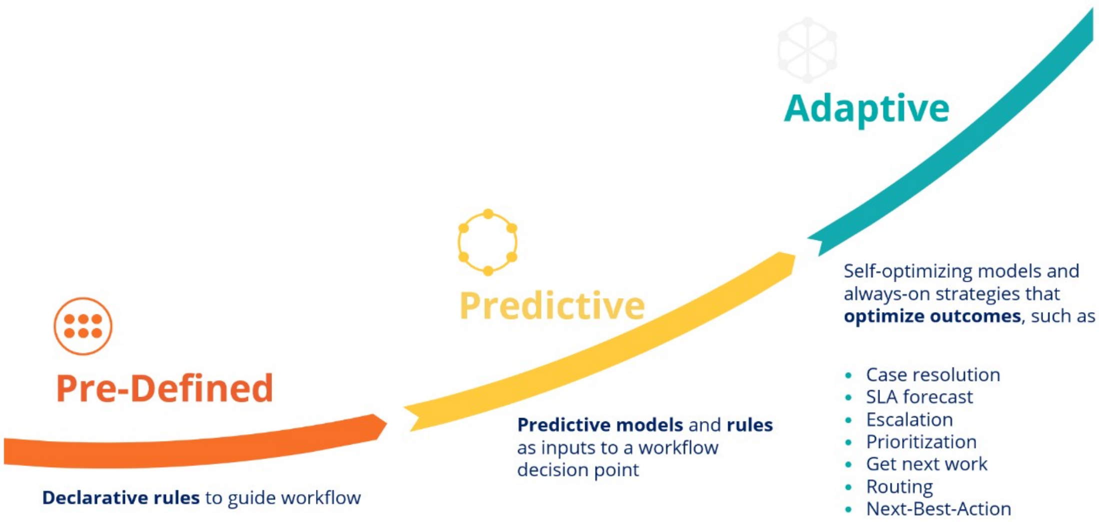
For example, the decision not to process an order can be based on a high credit risk score, and then the application can route the dubious claim for closer inspection.



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.

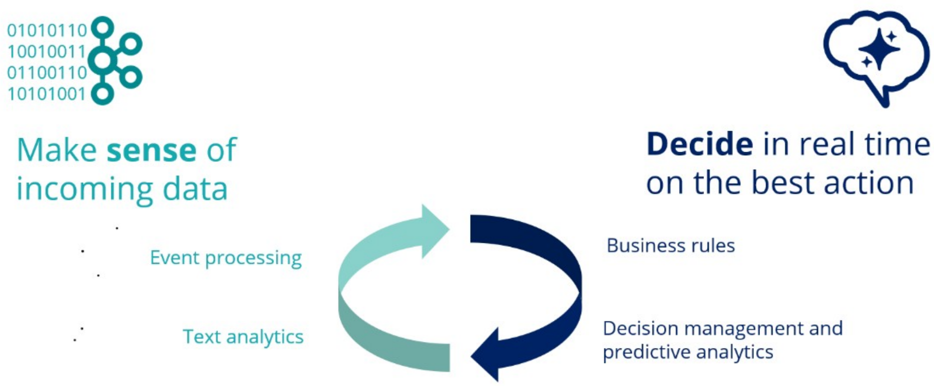
To optimize case outcomes, use adaptive models that can predict outcomes, such as case resolution, or intelligently prioritize and route cases to optimize business value and customer experience.



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.

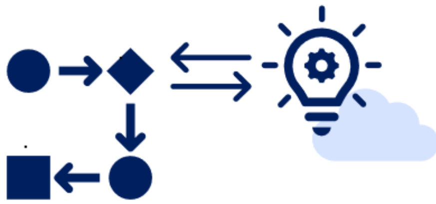


The decision is based on the business rules and supported by predictive analytics. This process is repeated every time that the case requests a decision.

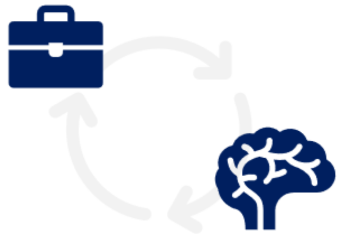
As the number of processed cases increases and model evidence accumulates, the predictive power of the models increases over time.

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

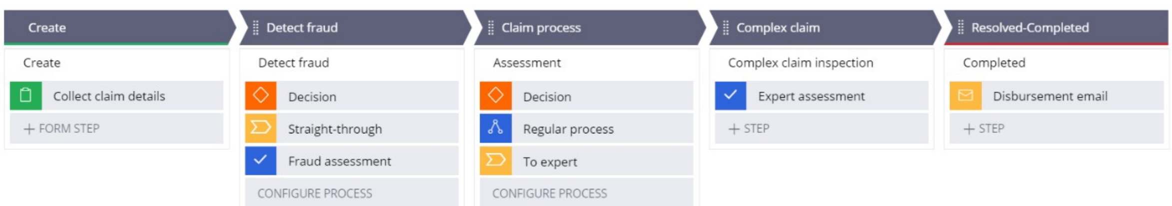
Introduction

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

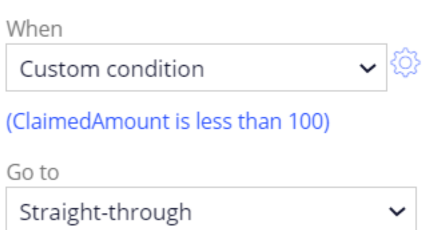
This demo will show you how to use a predictive fraud model in a case type to route suspicious claims to an expert.

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.

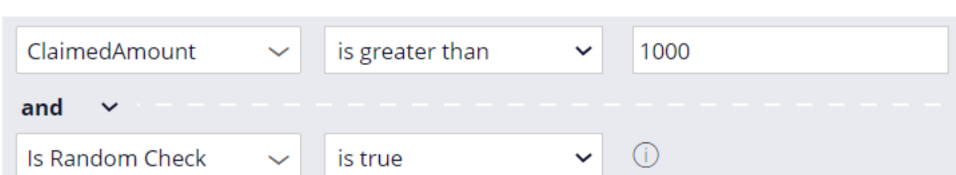


When the case is resolved, the claimant receives an email that communicates the decision.

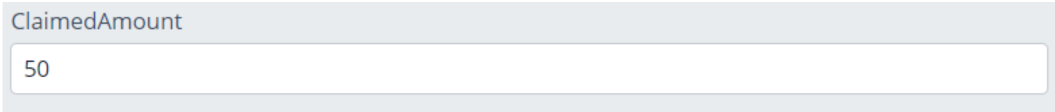
The decision step in the **Detect fraud** stage routes cases with a low claimed amount for straight-through processing.



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

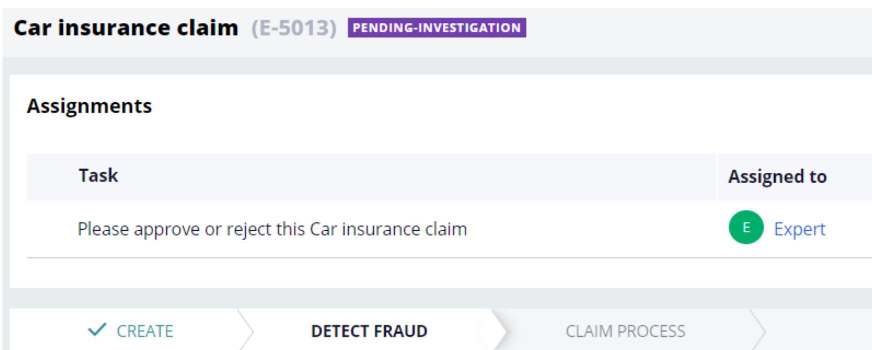


Consider this car insurance claim. The claimed amount is 50.

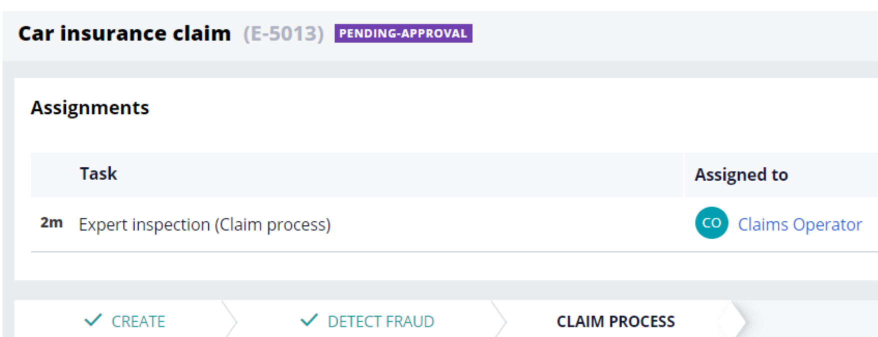


The claim qualifies for straight-through processing as the claimed amount is below the threshold. The case is automatically resolved, and the claimant receives an email that states that the claimed amount will be disbursed.

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.



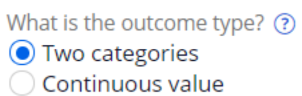
U+ Insurance wants to improve the effectiveness of fraud detection by using a predictive model that calculates the fraud risk of each claim.

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.

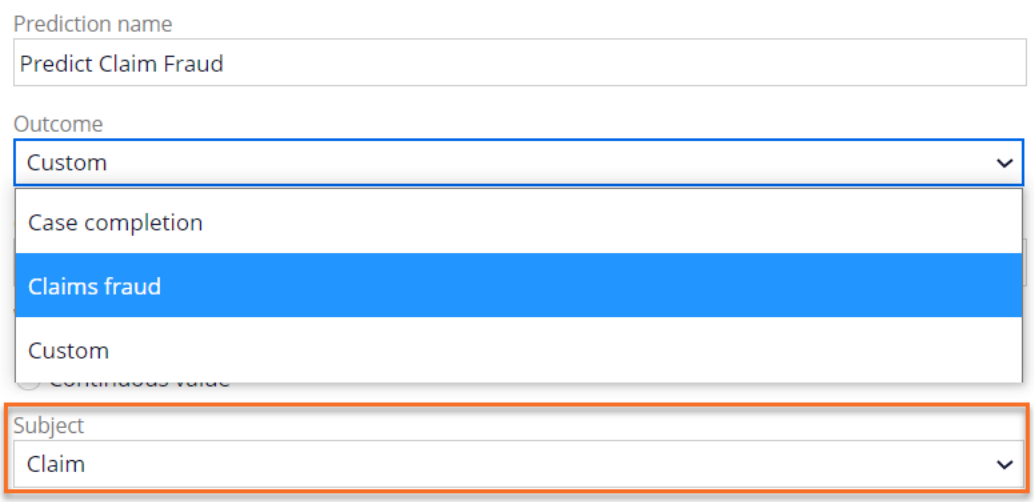
The data scientist team of U+ Insurance has developed a fraud model on the H2O.ai platform and has validated the model against historical data that the company captured.

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.

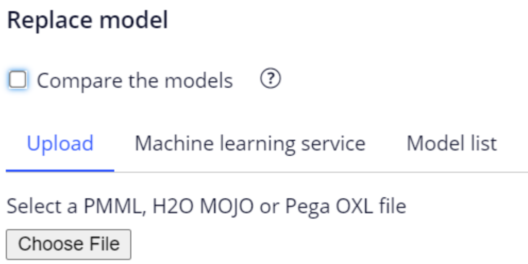


A placeholder scorecard initially drives the prediction.

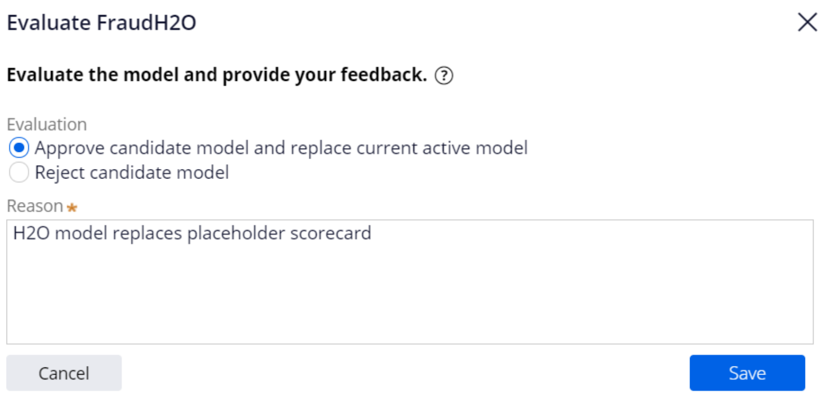

Claims fraud

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.

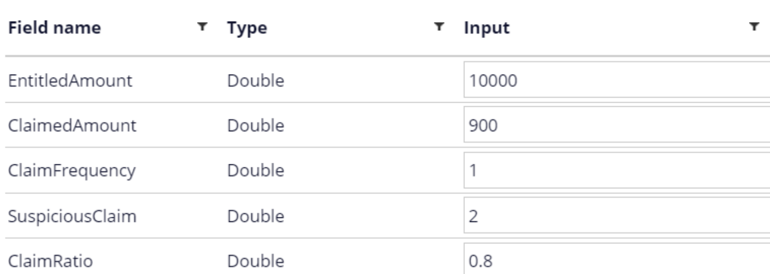


You can select predictive models that are available in the application in the model list. When the model is ready for review, approve the model to replace the scorecard.



The fraud model now drives the prediction.

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



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.

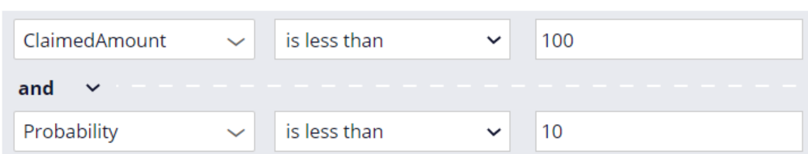
Predictions

Manage predictions and associated objectives

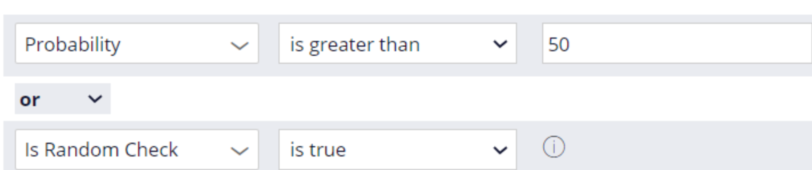
Prediction	Objective	Data object
Predict Fraud Risk	Claims fraud	Claim

+ Add prediction

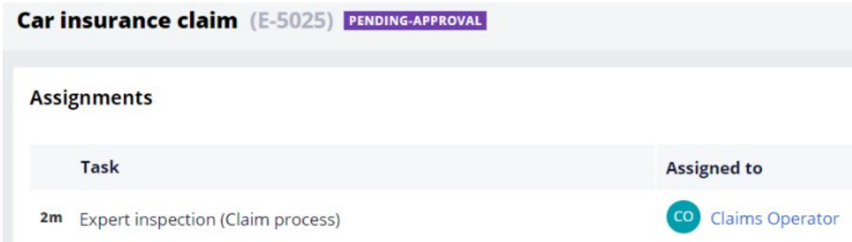
Next, in the **Decision** step in the **Detect fraud** stage of the life cycle, implement the prediction. Add the condition that only claims with a very low predicted fraud risk qualify for straight-through processing.



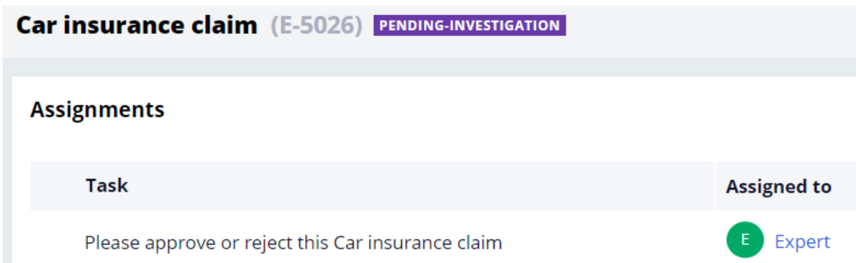
Replace the condition that routes a claim to a fraud expert based on the claimed amount with a condition that is based on the outcome of the fraud model and change the logical operator to generate the control group.



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.

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.

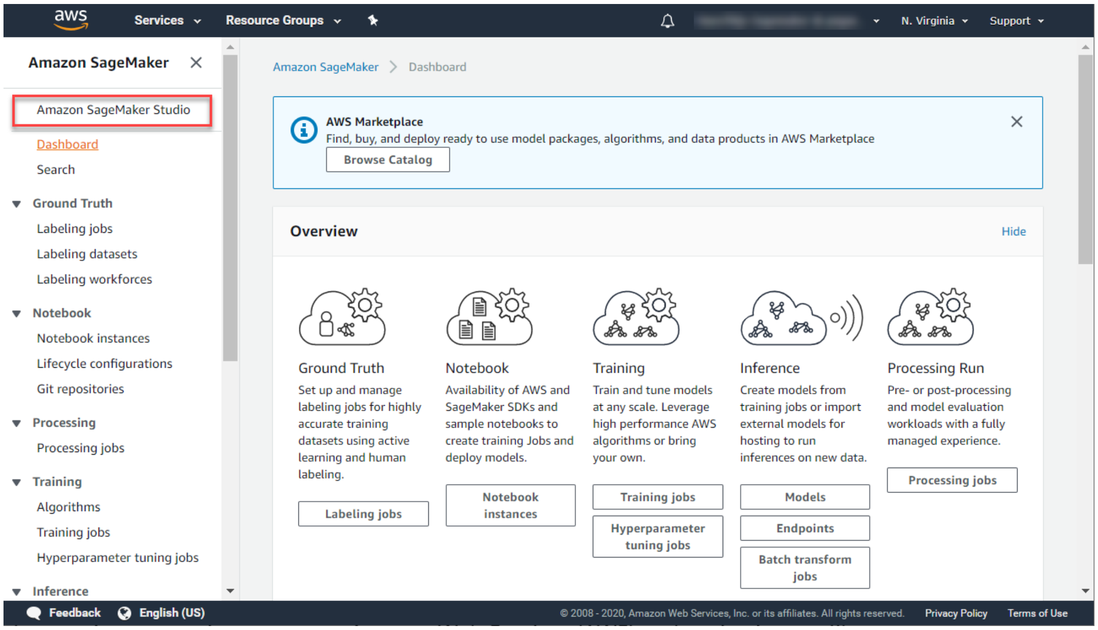
Transcript

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.

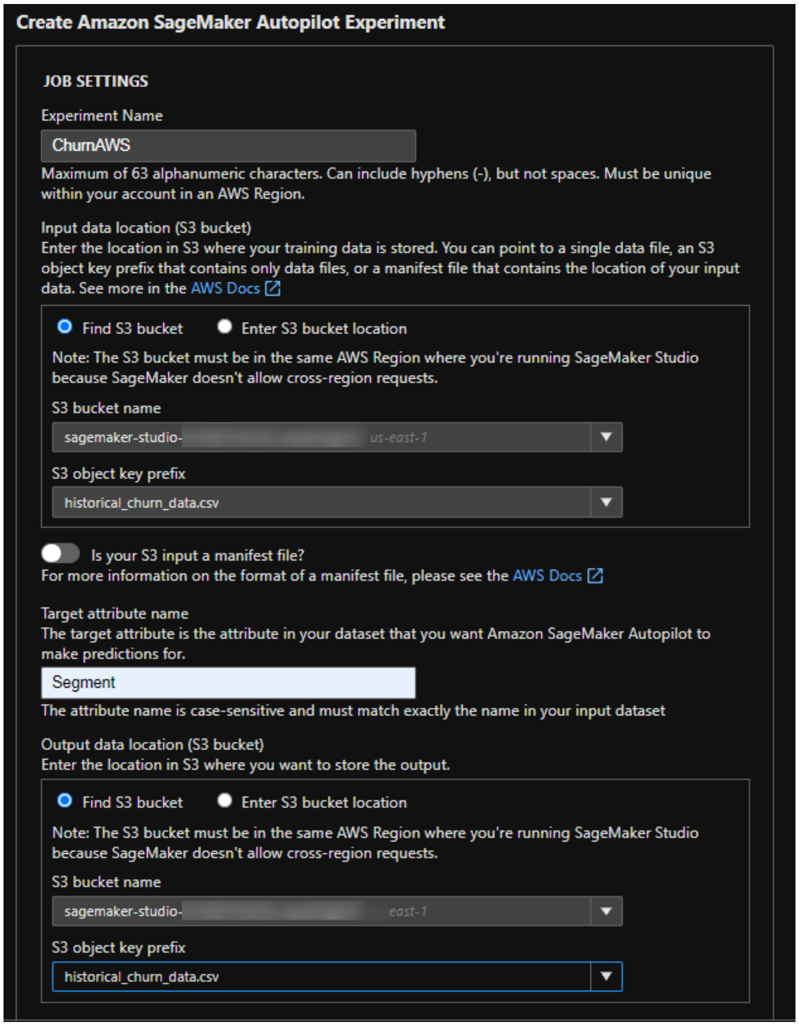
We will showcase this using Amazon SageMaker. The steps are similar to using other machine learning services such as Google AI Platform. Using a machine learning service instead of a model that runs locally may involve costs and possible down time of the service.

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.

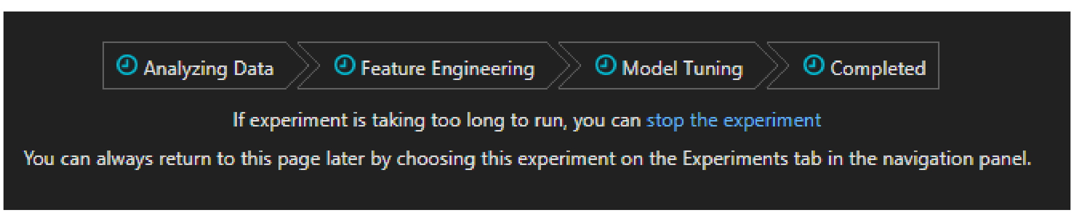
Amazon SageMaker allows you to build, train and deploy machine learning models in a fully managed service. The Autopilot feature automates this process and trains and tunes the best machine learning models for classification or regression, based on your data. After setting up your AWS environment, you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



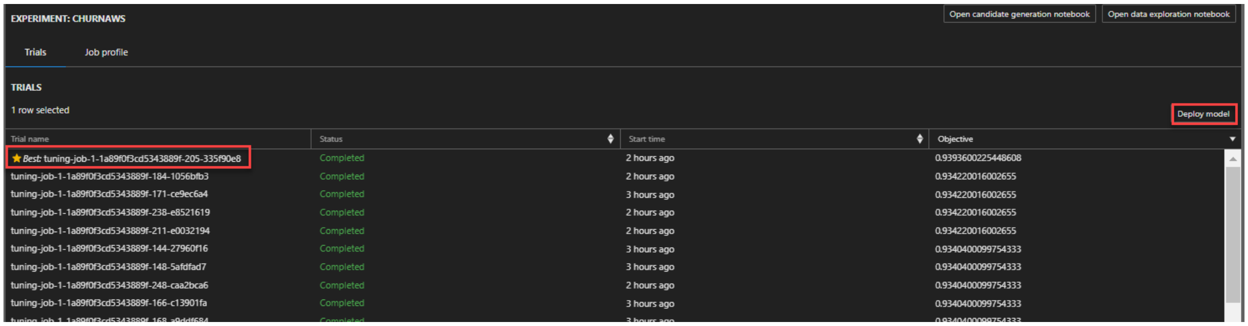
In the Job settings, select the data file you want to build the model on, specify the outcome field, choose the location where the output should be stored and create the experiment.



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

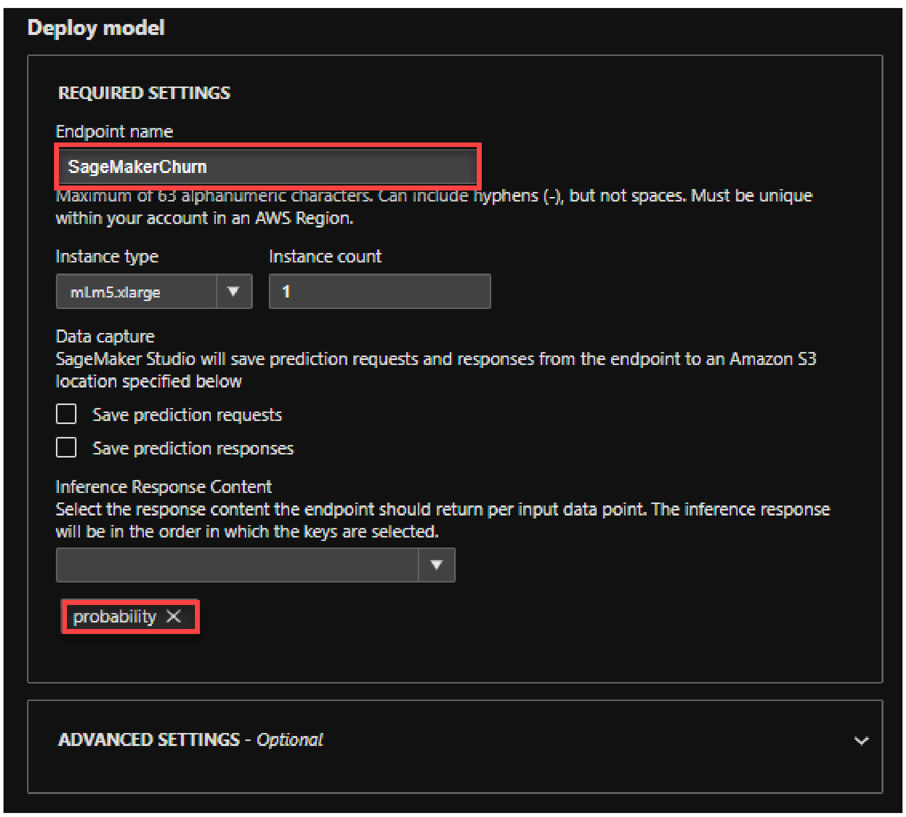


To deploy the best candidate model, select the tuning job with the highest Objective value. This value indicates the predictive power of the model.

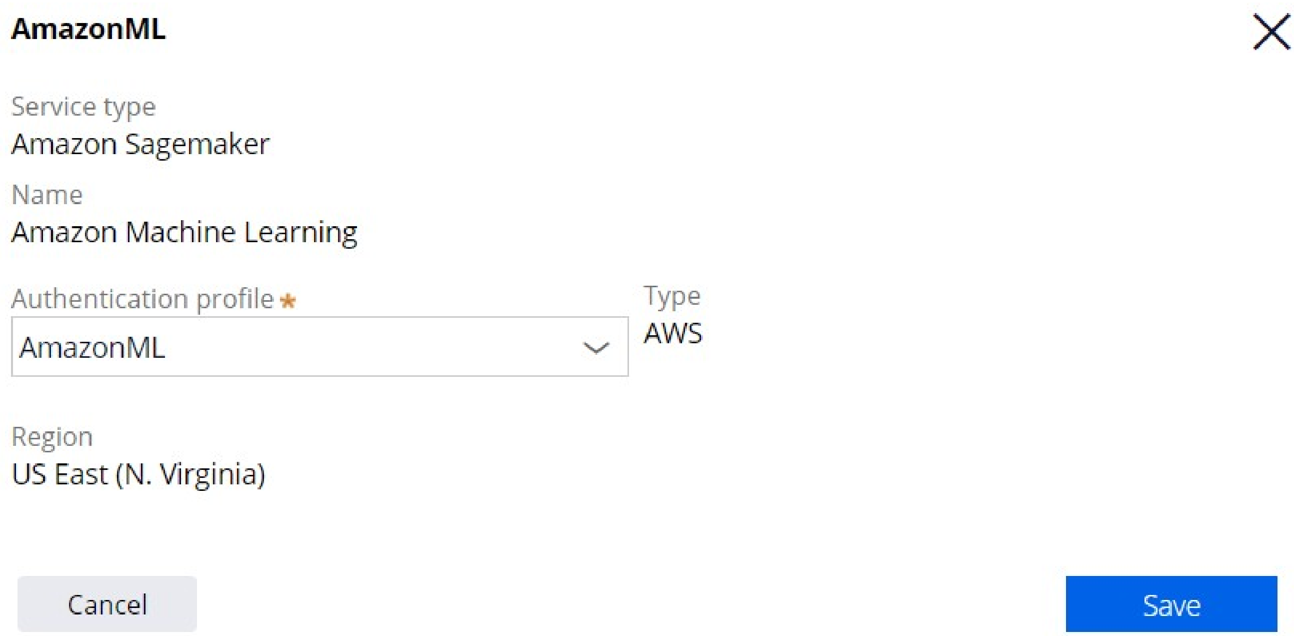


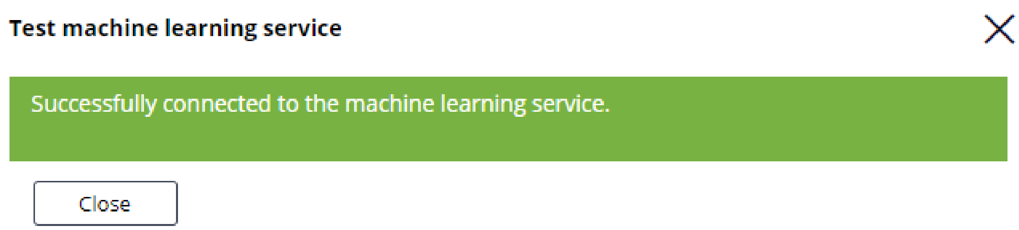
An endpoint that can be reached from Pega is automatically created. A binary classification, as in this example, predicts if an event will happen or not, based on a cut-off value. By default, the response content for a binary model is set to this ‘predicted_label’.

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.

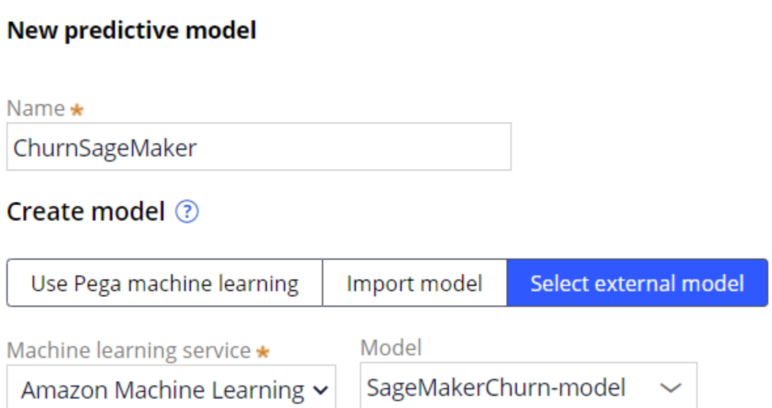


In Prediction Studio, you can define a machine learning service to connect to your cloud service instance. To move messages securely to and from Pega, the system architect has set up an authentication profile.





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.

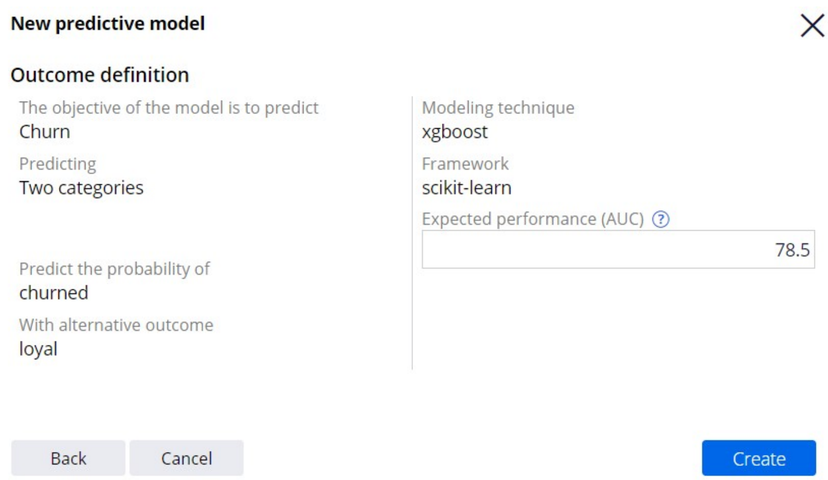
```json
{
	"predictMethodUsesNameValuePair": false,
	"predictorList": [{
			"name": "GENDER",
			"type": "CATEGORICAL"
		},{
			"name": "AGE",
			"type": "NUMERIC"
		}
	],
	"model": {
		"objective": "Churn",
		"outcomeType": "BINARY",
		"expectedPerformance": 78.5,
		"framework": "SCIKIT_LEARN",
	"modelingTechnique":"Naive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

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

For SageMaker, the file must include the framework property. This property determines the input format and output format of the model. In Google AI Platform, this property is automatically fetched. Finally, the metadata file must include the modeling technique and the outcome values.

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

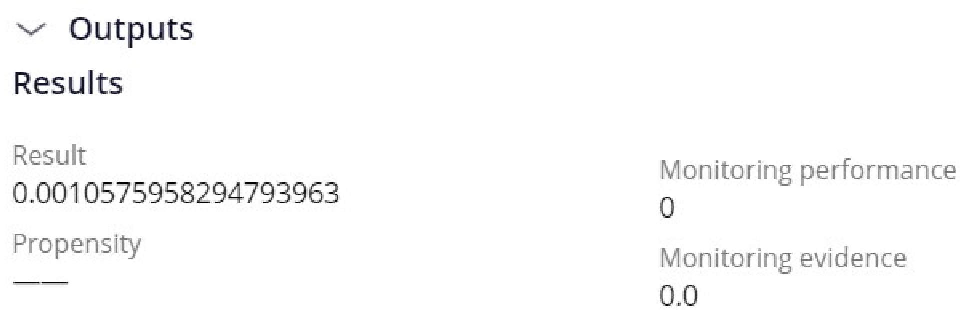
Next, set the correct context of the model if required. The default context is the customer class. You can review the model metadata, such as the objective of the model and the type of problem to solve, before proceeding.



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.

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

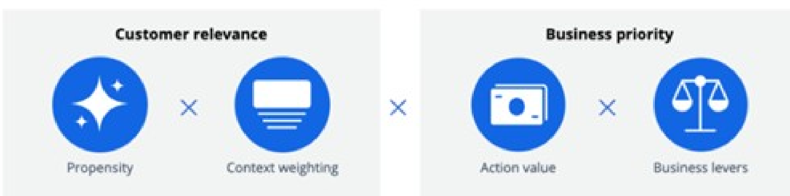
Enhance decision strategies with predictive models built on customer interaction data and let Pega Customer Decision Hub™ bring even more relevance to every customer engagement. Build models using Pega's machine learning capabilities, import models built with third-party tools and incorporate the latest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

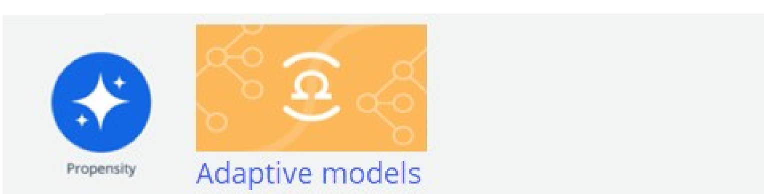
Next best actions balance customer relevance and business priorities by selecting the actions with the highest priority.

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.

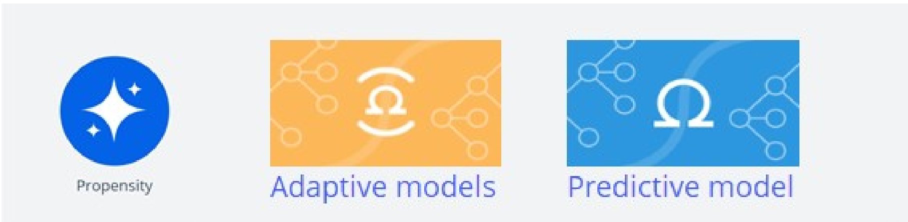
This is calculated by predictive models. In Pega, self-learning Naive Bayes models, which are generated for each action, are a key feature.



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.



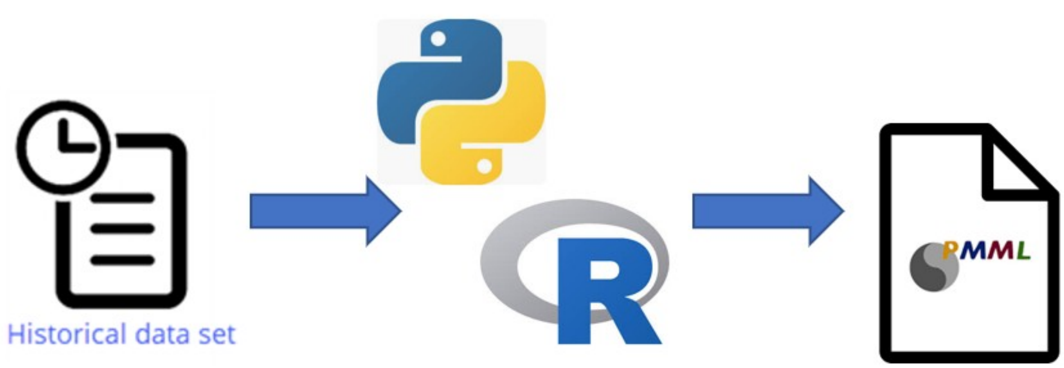
The first method is to use Pega machine learning. You can import a file containing the historical customer interaction data set and build a model in Prediction Studio.

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.

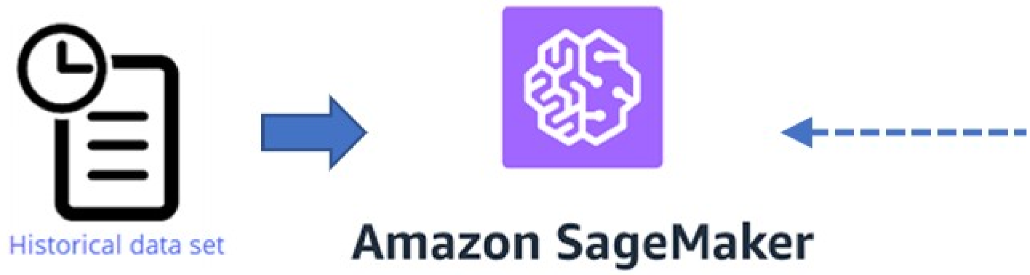
When decision strategies using the imported models execute, the models are executed inside the Pega platform.

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

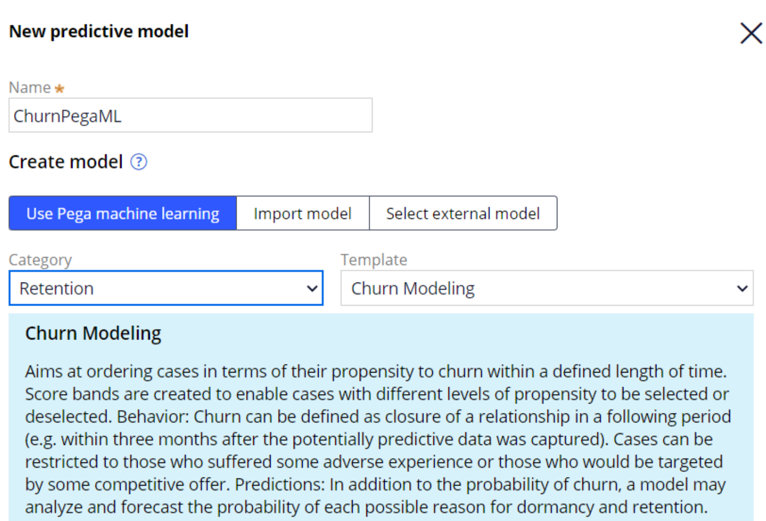
Prediction Studio offers several options for leveraging customer data to create predictive models. Learn how to develop powerful and reliable models that can predict customer behavior, such as offer acceptance, churn rate, credit risk, or other types of behavior by using Pega machine learning.

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.

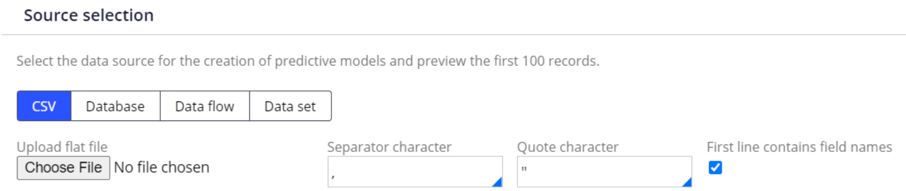
When starting to build a new model, you will be presented with the option to create a model on a template that is used for streamlining model development. One of these is churn modeling.



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.

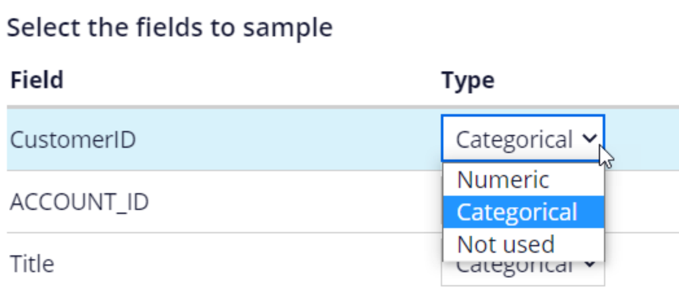
Preview for first 10 records of historical_data.csv

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.

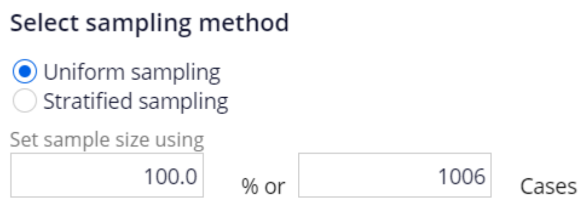


By default, all fields are considered potential predictors. When setting predictors, it's important to use some common sense.

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'.

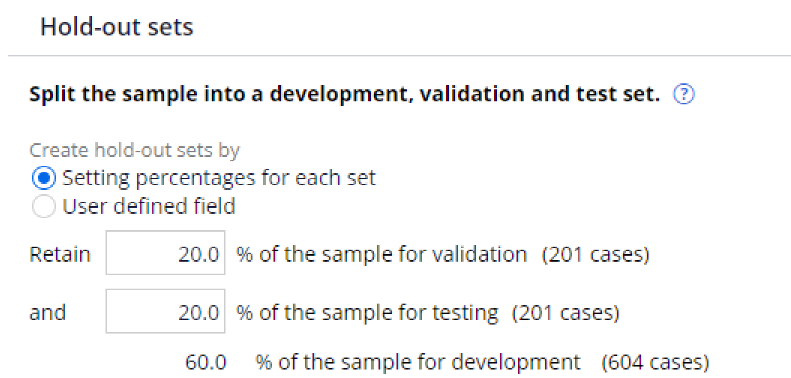
If the data contains a relatively small number of cases, you will want to use 100% of the records. If the data source is large, a sample will be sufficient.



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.

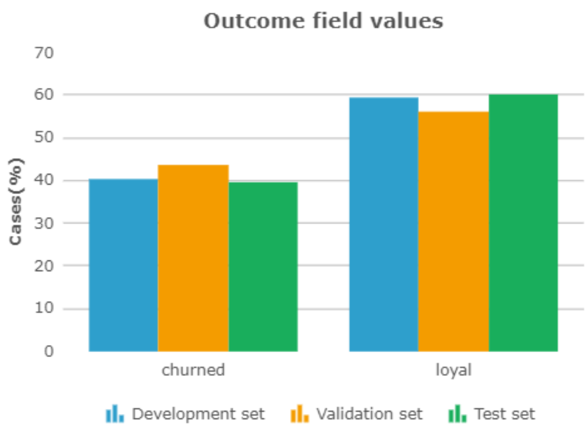
Outcome type: Binary

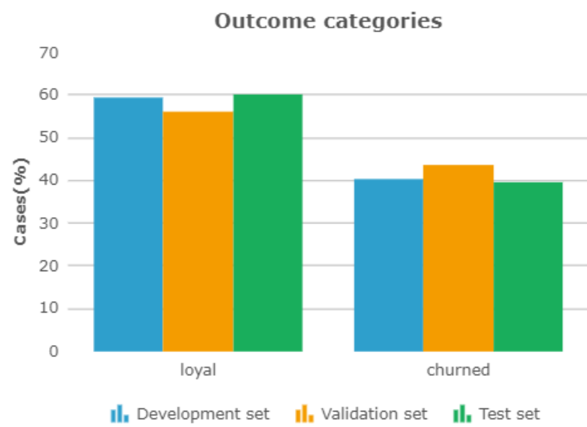
Outcome field to predict: Segment

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

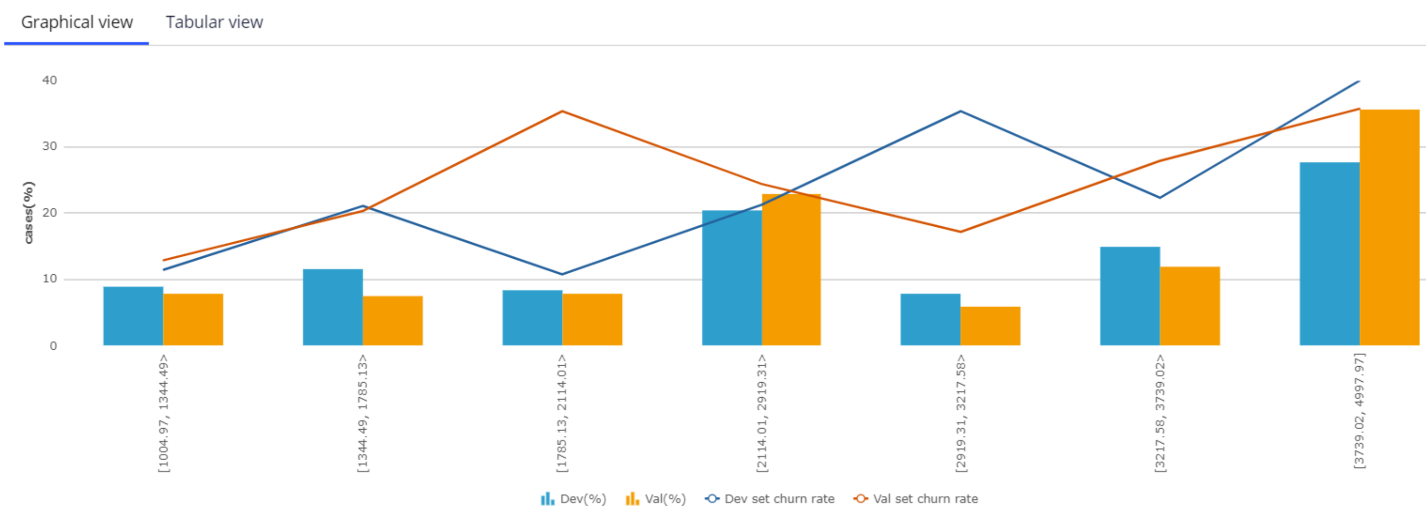
Predictor	Type	Role	Binned intervals	Grouped intervals	Grouped performance
RiskScore	Numeric	VALUE	200	11	90.27
AverageSpent	Numeric	PREDICTOR	200	7	**68.28**
MonthlyPremium	Numeric	PREDICTOR	200	10	**64.84**
Age	Numeric	PREDICTOR	70	8	**63.87**

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**

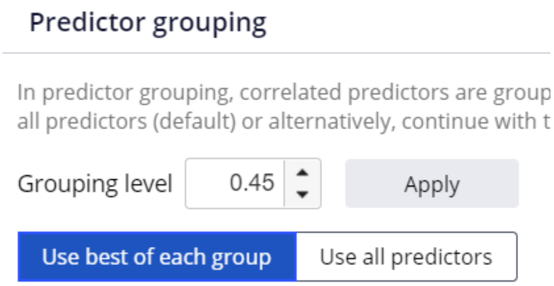
Data analysis creates a binned, ordinal view of individual predictors. Both Binning and Granularity are automatically set but can be manually adjusted.



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.

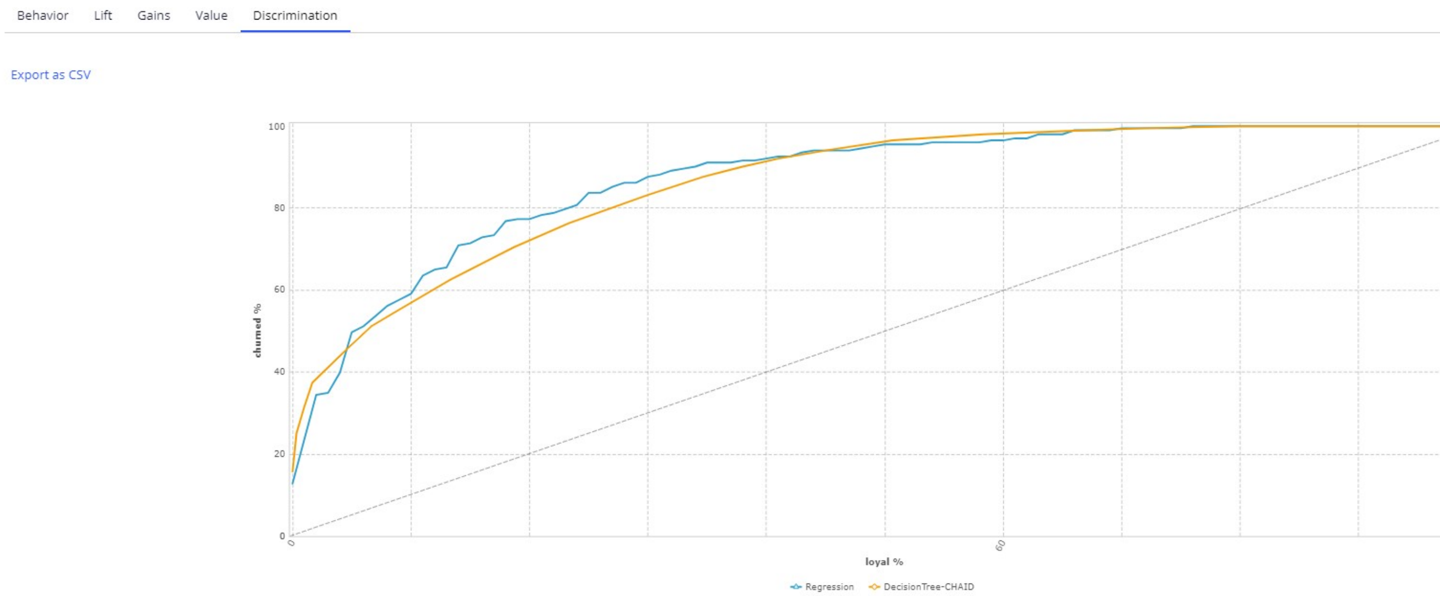
The purpose of the next step, Model Analysis, is to select the best model for your use case.

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

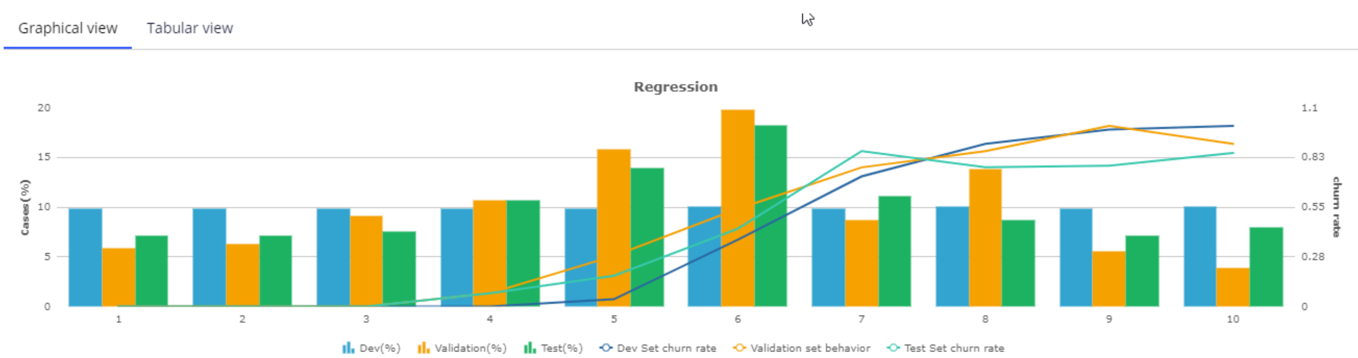
Prediction Studio uses Area Under the Curve (or AUC) to measure the performance of predictors and models.

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.

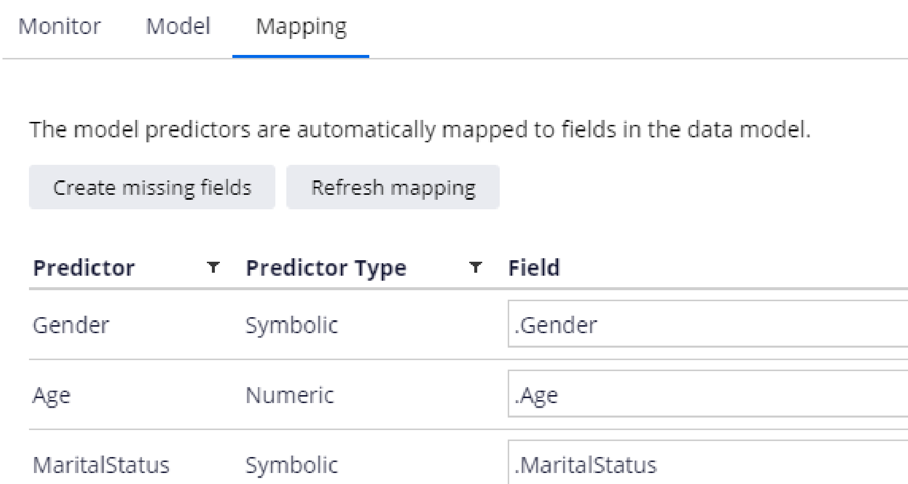
In the ‘Class comparison’ step, you can analyze and compare models after the score distribution has been adjusted.

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.

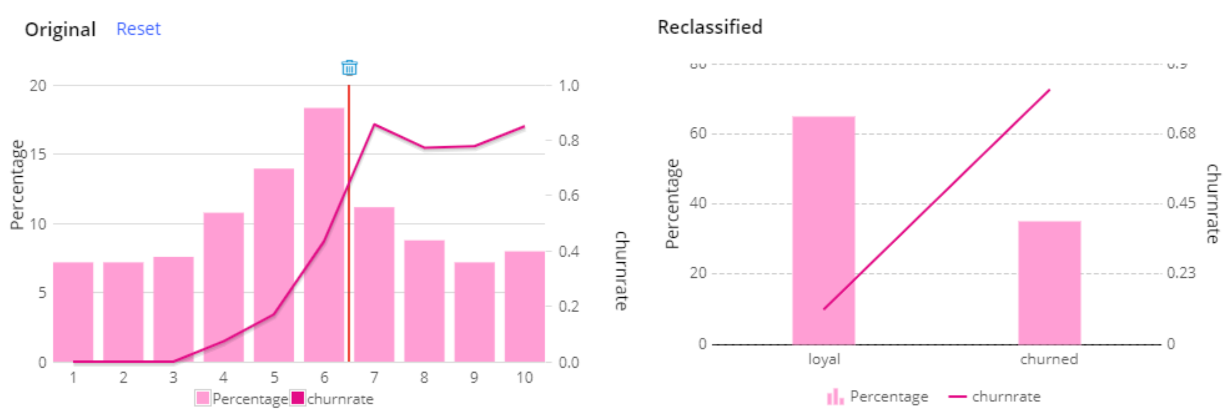
Before you can save the model, check the mapping of the predictors to the properties in the customer class.

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

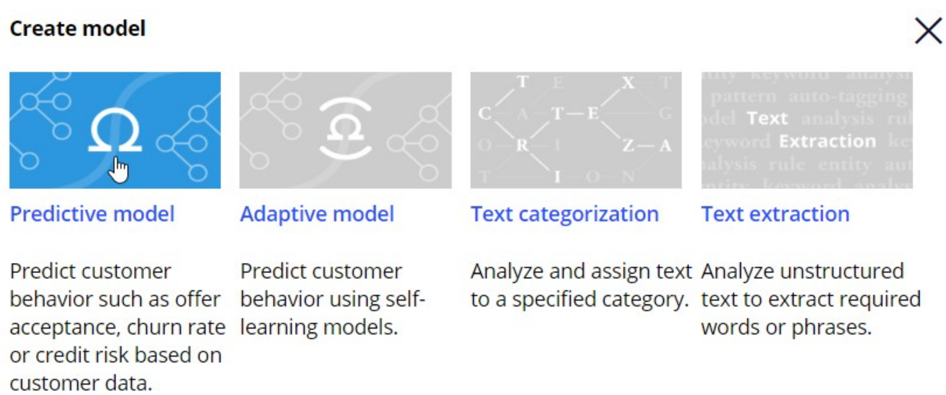
During a Pega Decision Management implementation project, you may discover that the company already uses predictive models. These assets can be reused in Pega Decision Management to help make customer predictions.

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.

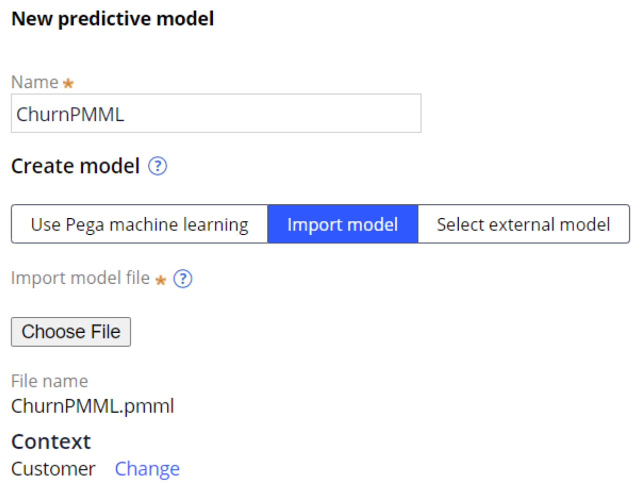
Prediction Studio supports two external model formats. First, you can import models in the Predictive Model Markup Language (PMML) format. PMML is an XML-based language aimed at easily sharing predictive models between applications. It is the de facto standard for representing not only predictive models,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

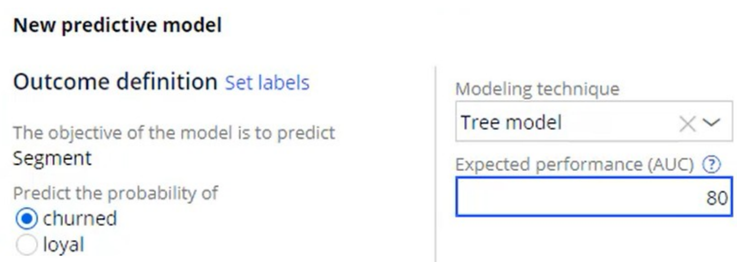


Prediction Studio offers three options for creating a predictive model: using Pega machine learning, importing a previously built model, or using an external model.

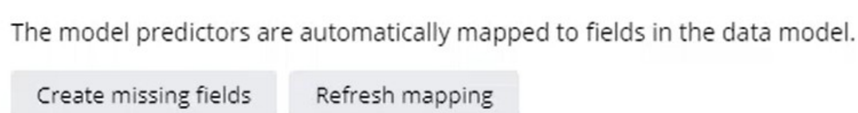
To leverage an existing model file, select the **Import model** option. Upload the PMML or H2O model file. The default context of the model is the **Customer class,** where the customer data model properties are stored. You can change this class if required.



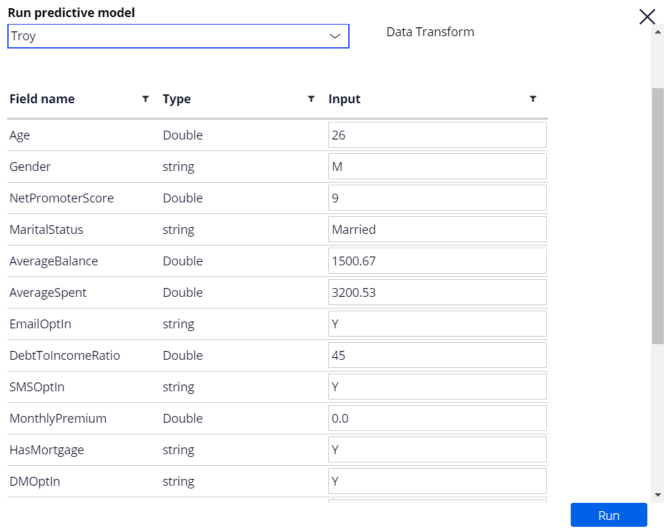
In the **Outcome definition** dialog box, you define which probability you would like to predict and the expected performance of the model, which is used as a benchmark when monitoring the model.



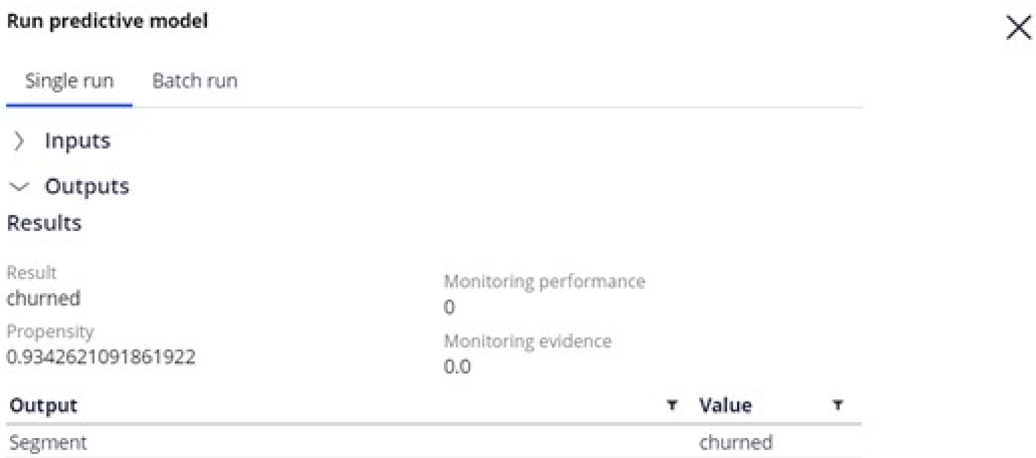
Import the model and, on the **Mapping** tab, make sure that all predictors are mapped to fields in the data model. Missing fields can be created, but this should be discussed with the system architect beforehand.



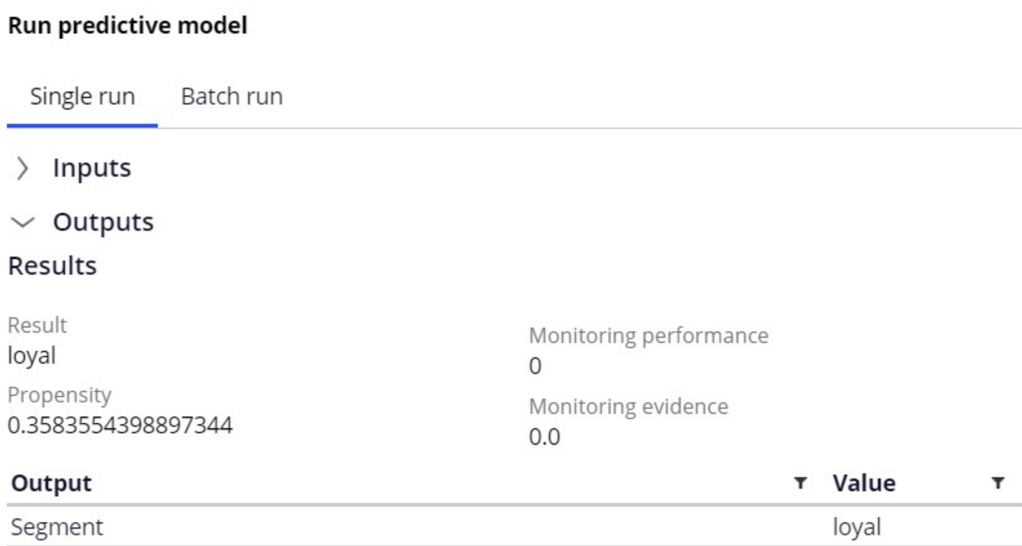
After the model is saved, you can test it for a single customer or run it for a batch of customers.



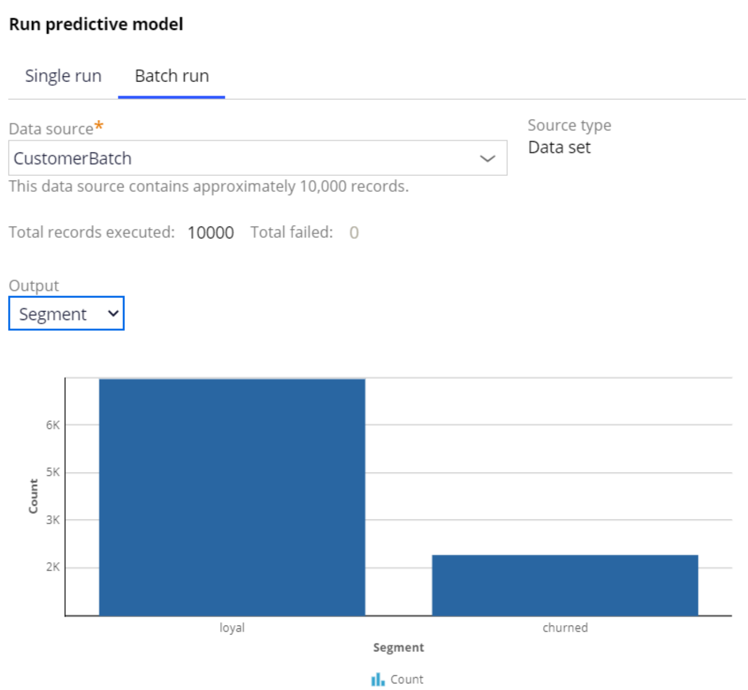
When you test the model for a single customer, you can use a data transform as input data. When customer Troy is used as the data source, the model predicts that he is likely to churn. The model also outputs his propensity to churn, which is, in this case, 93.42%.



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.

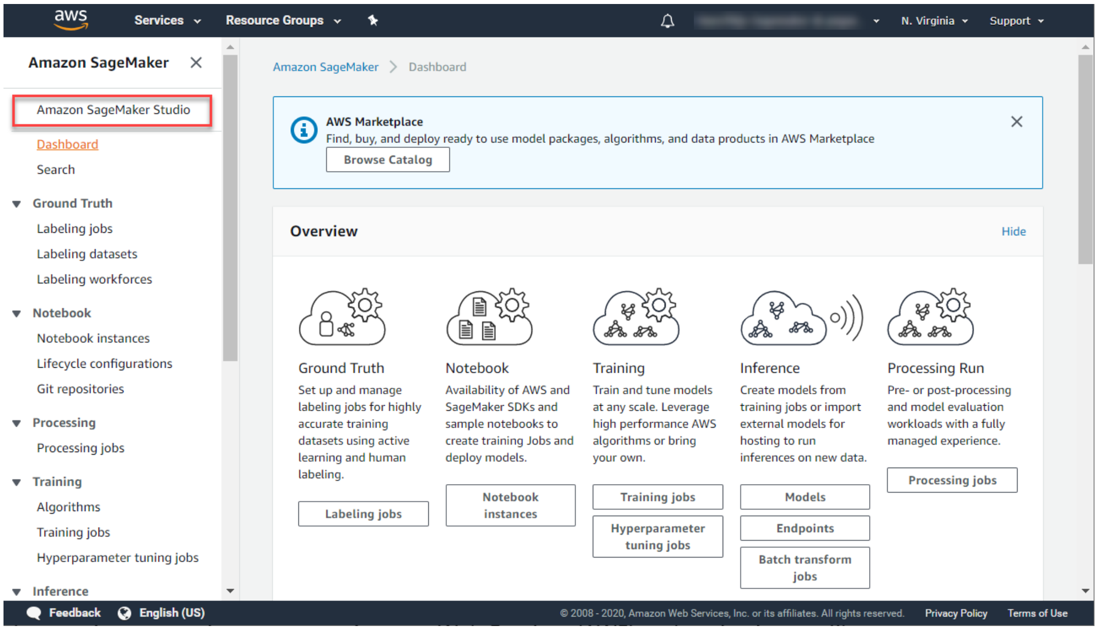
Transcript

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.

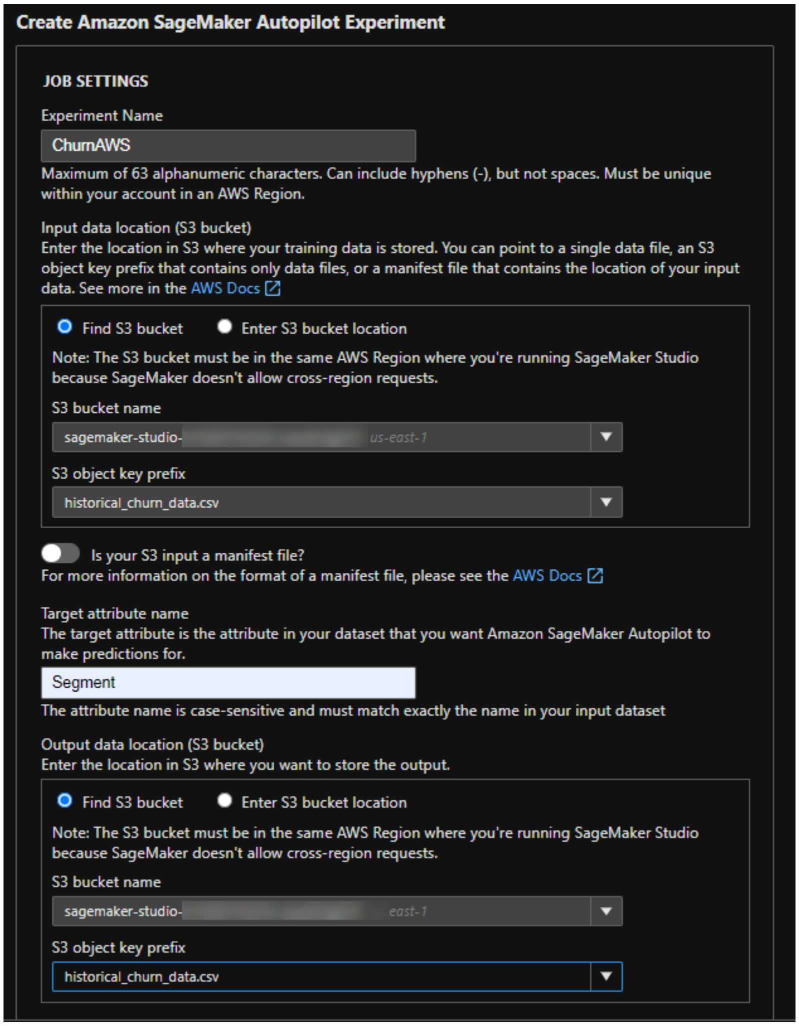
We will showcase this using Amazon SageMaker. The steps are similar to using other machine learning services such as Google AI Platform. Using a machine learning service instead of a model that runs locally may involve costs and possible down time of the service.

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.

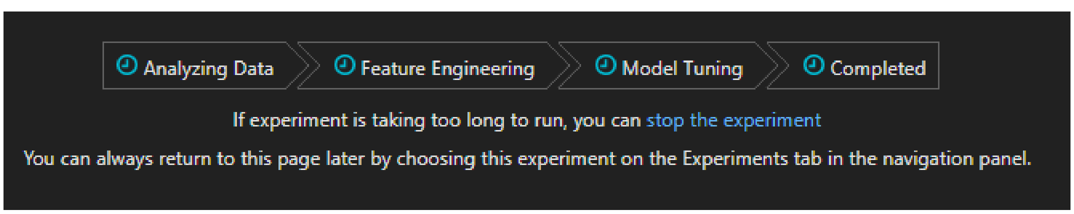
Amazon SageMaker allows you to build, train and deploy machine learning models in a fully managed service. The Autopilot feature automates this process and trains and tunes the best machine learning models for classification or regression, based on your data. After setting up your AWS environment, you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



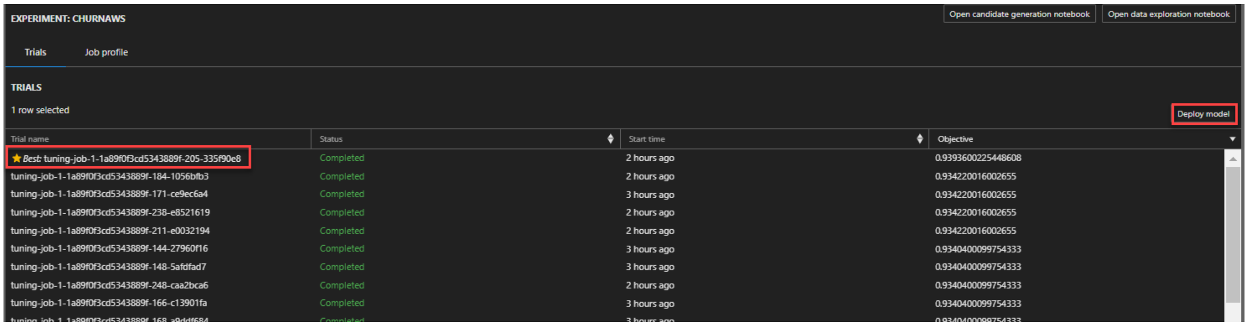
In the Job settings, select the data file you want to build the model on, specify the outcome field, choose the location where the output should be stored and create the experiment.



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

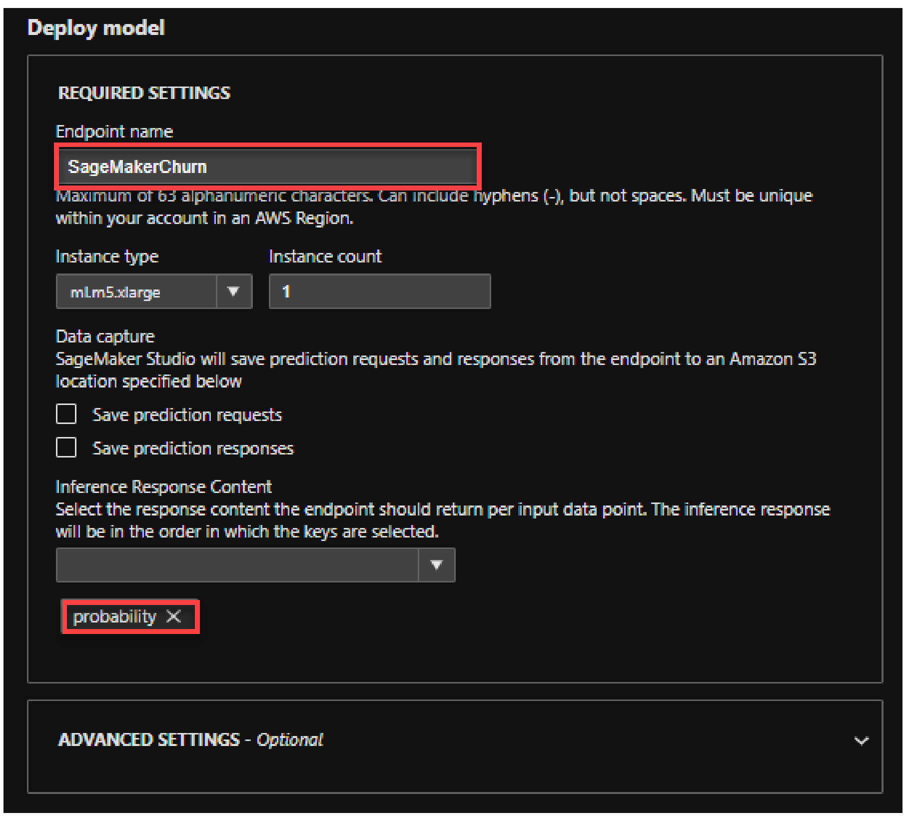


To deploy the best candidate model, select the tuning job with the highest Objective value. This value indicates the predictive power of the model.

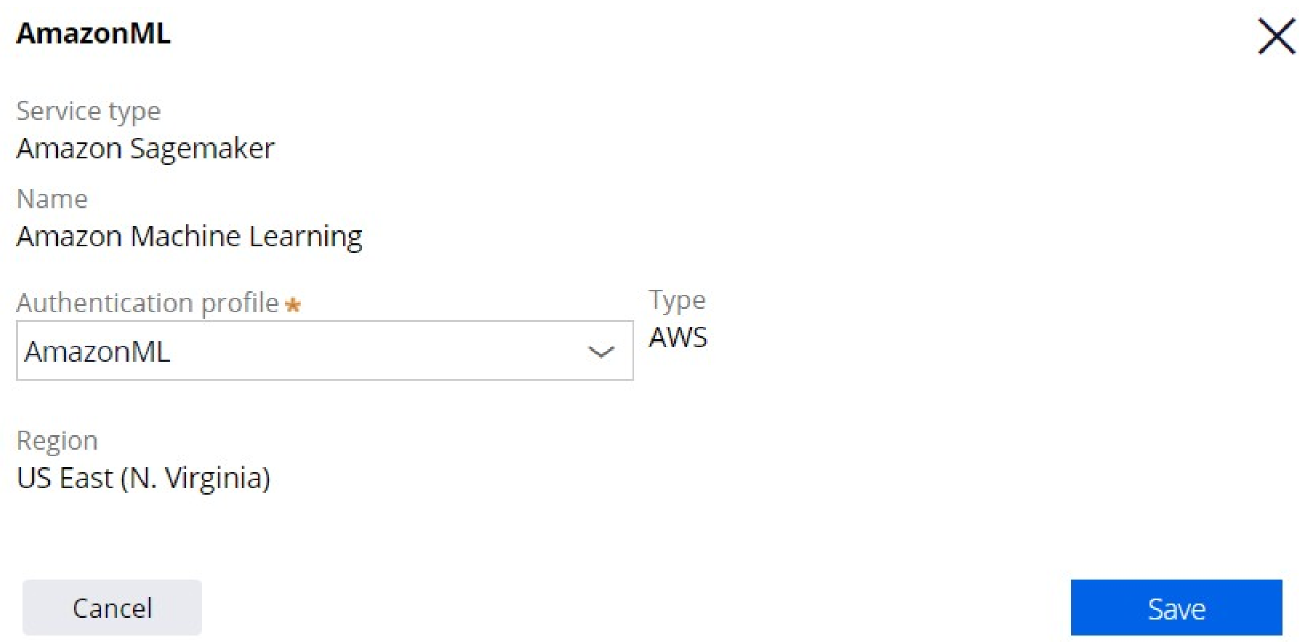


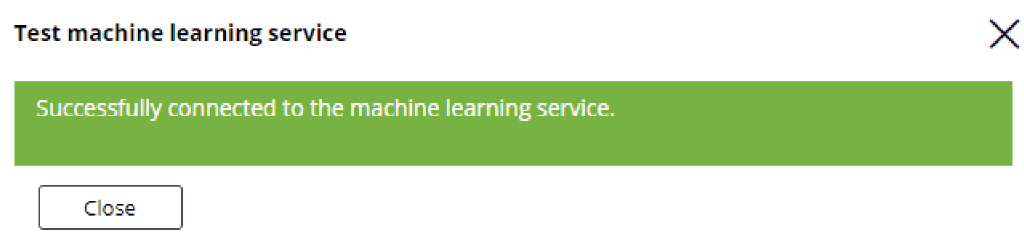
An endpoint that can be reached from Pega is automatically created. A binary classification, as in this example, predicts if an event will happen or not, based on a cut-off value. By default, the response content for a binary model is set to this 'predicted_label'.

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.

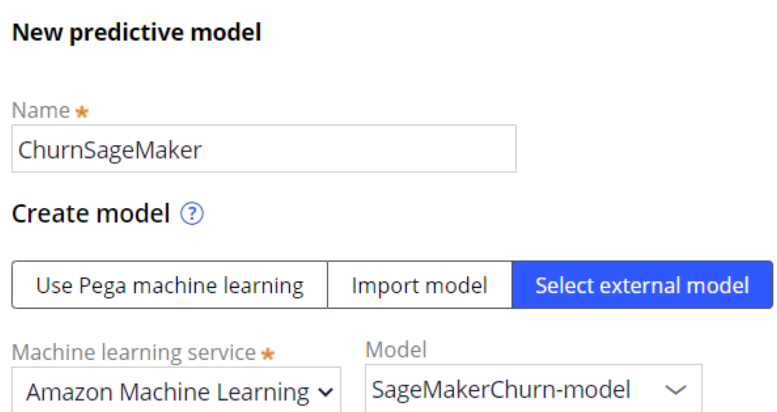


In Prediction Studio, you can define a machine learning service to connect to your cloud service instance. To move messages securely to and from Pega, the system architect has set up an authentication profile.

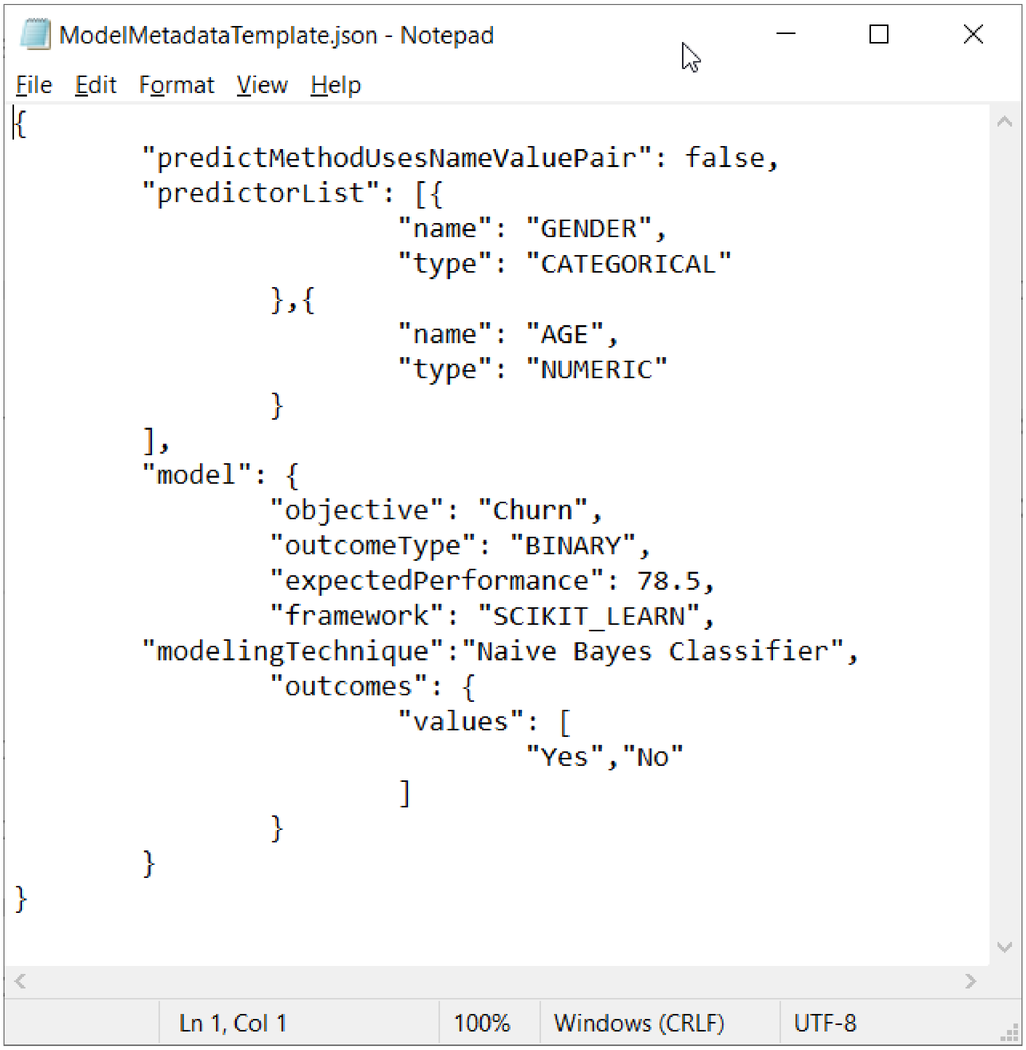




Once the connection to the machine learning is established, start by creating a new predictive model to leverage the service. Select the machine learning service and the model that you want to reference.



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



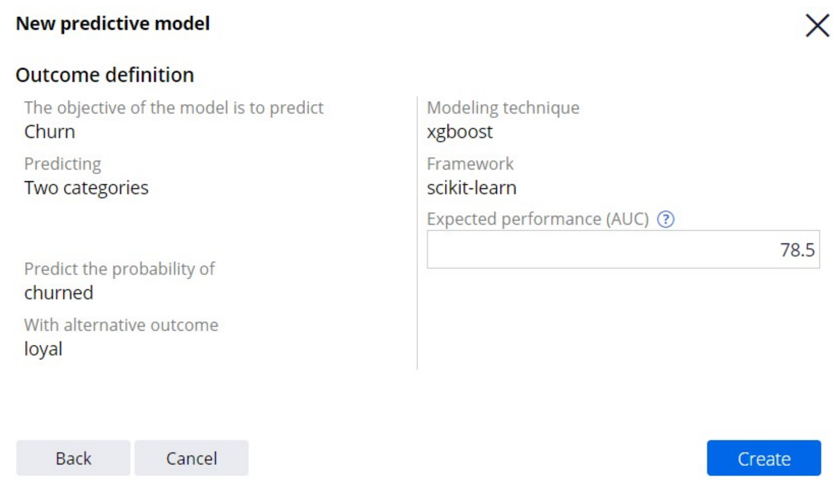

```
{
	"predictMethodUsesNameValuePair": false,
	"predictorList": [{
			"name": "GENDER",
			"type": "CATEGORICAL"
		},{
			"name": "AGE",
			"type": "NUMERIC"
		}
	],
	"model": {
		"objective": "Churn",
		"outcomeType": "BINARY",
		"expectedPerformance": 78.5,
		"framework": "SCIKIT_LEARN",
	"modelingTechnique":"Naive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

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

For SageMaker, the file must include the framework property. This property determines the input format and output format of the model. In Google AI Platform, this property is automatically fetched. Finally, the metadata file must include the modeling technique and the outcome values.

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

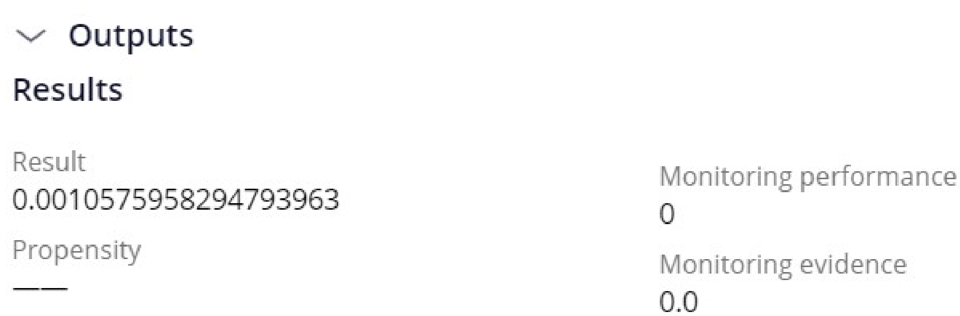
Next, set the correct context of the model if required. The default context is the customer class. You can review the model metadata, such as the objective of the model and the type of problem to solve, before proceeding.



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?

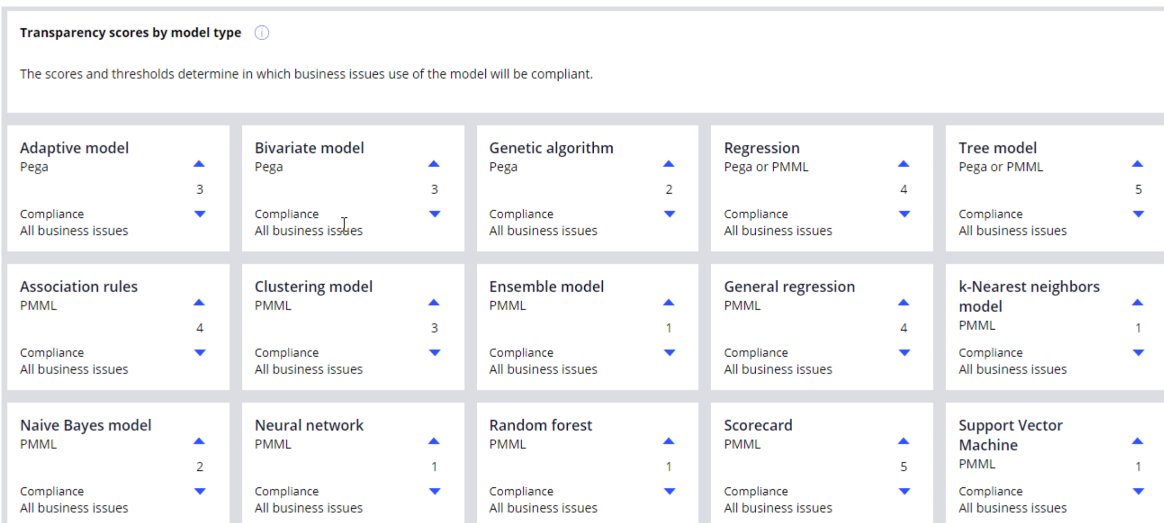
- The high level steps involved in creating a model using Amazon SageMaker Autopilot.
- How to connect to external machine learning services and run a model externally.

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.



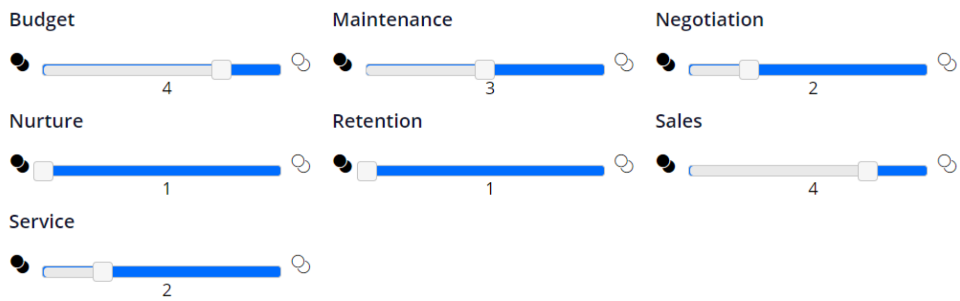
Configuring the model transparency policy

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.

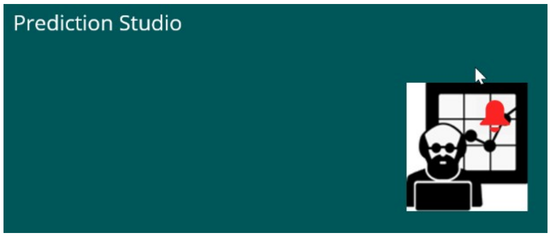
If a candidate predictive model is deployed in shadow mode, it can be monitored with real production data without impacting the business outcomes. If the model proves effective, it is deployed as the active model.

Transcript

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

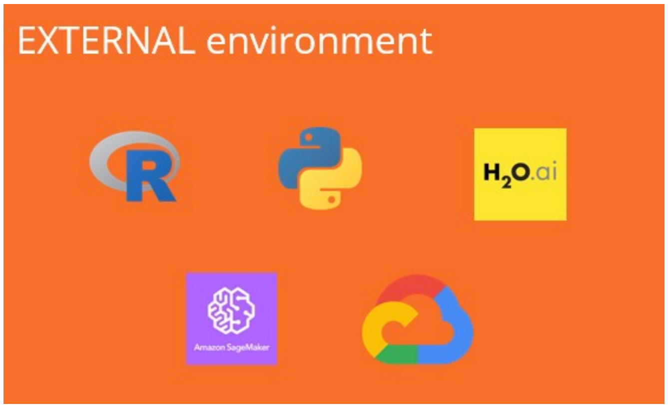
In the standardized Machine Learning Operations (MLOps) process, the active model is replaced with a better one in a production environment by using the shadow mode option. A prediction is driven by an adaptive model, a predictive model, a scorecard model, a field model, or a combination of these models.

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.

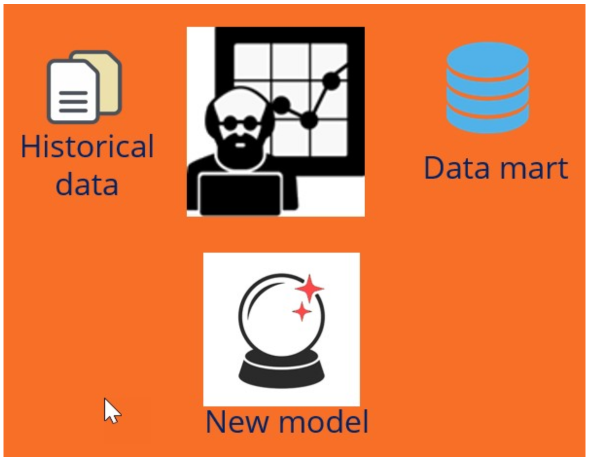


To build a new model, you can use Pega machine learning or an external environment. You can use data science tools that can export models in the PMML format, such as R or Python.

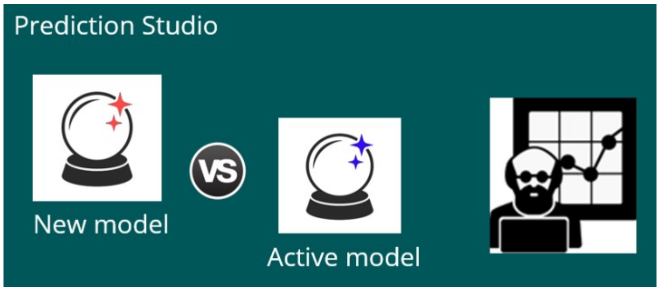
The H20 format is another option. You can also connect to the Amazon SageMaker or Google Cloud machine learning services.



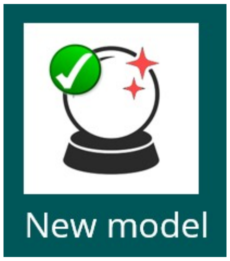
You can utilize the historical data of models captured by the system by importing these records into your external environment of choice. The historical data can be combined with data from other sources to build a new model.



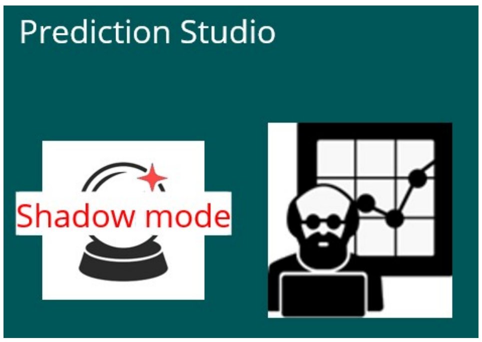
Once the new predictive model is developed, you validate the active model and the candidate model against the same data set to compare their metrics in Prediction Studio.



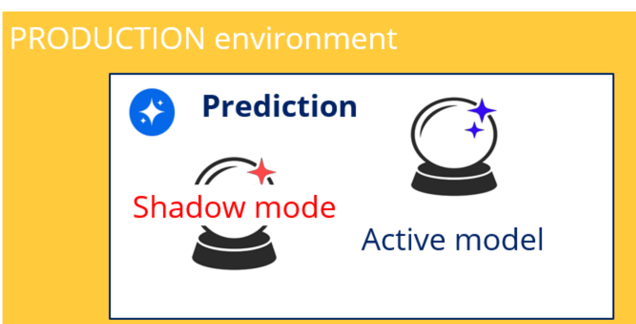
If the candidate model outperforms the active model, approve the model.



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

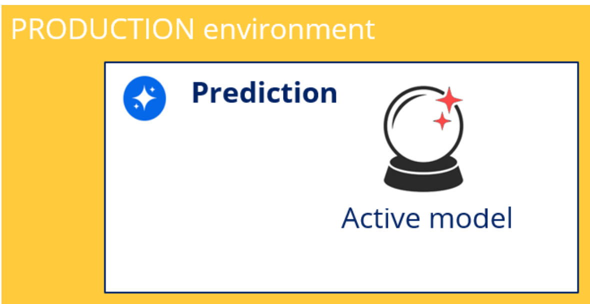


The new model is then promoted to the production environment in a revision.



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

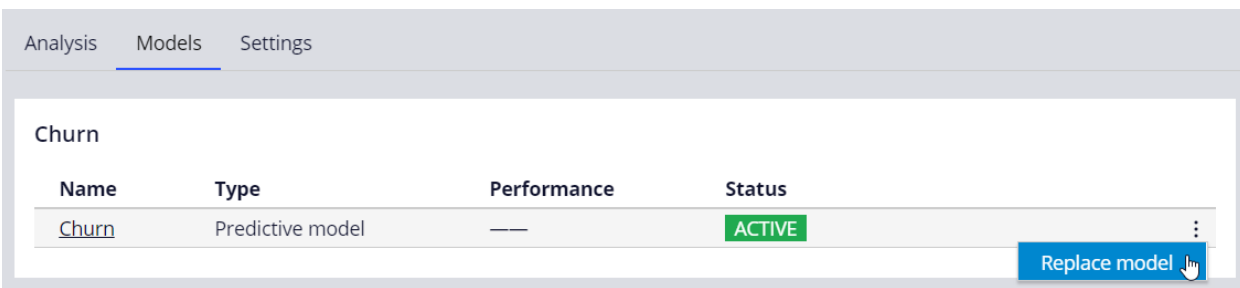
As a data scientist, you can approve changes to the models that drive predictions for deployment to the production environment. You can change models independently or respond to a Prediction Studio notification that a prediction does not generate enough lift.

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.

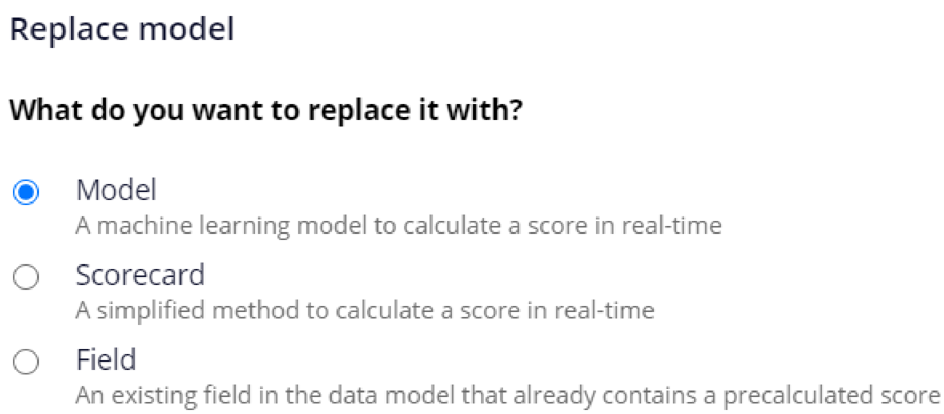
Model deployment

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.

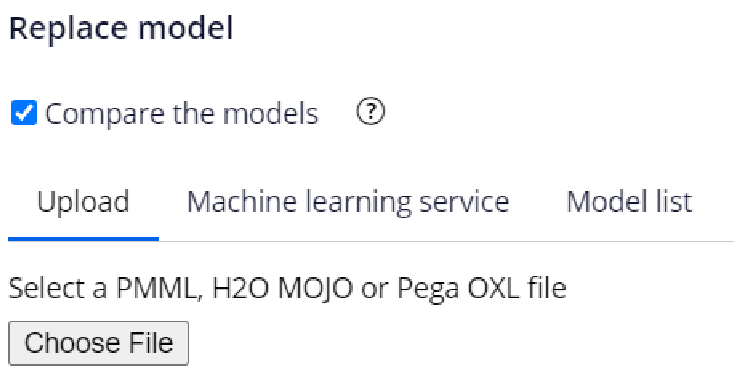
The process begins when you update a prediction in your non-production environment.



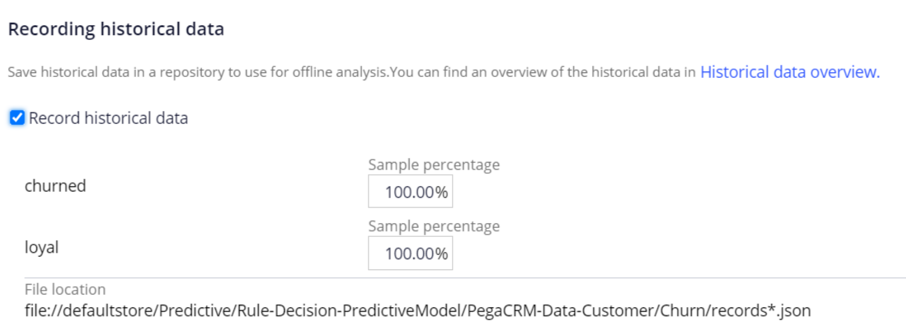
You can replace the active model with a predictive model, scorecard, or field in the data model that contains a score.



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.

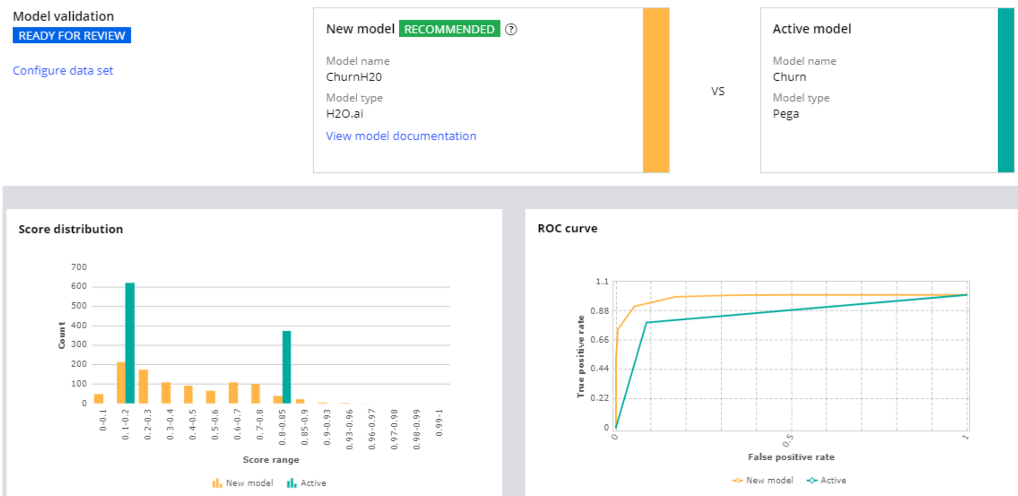


When creating a machine learning model in a third-party environment, you can use historical data from Pega Platform™ and other sources to train the model.



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.

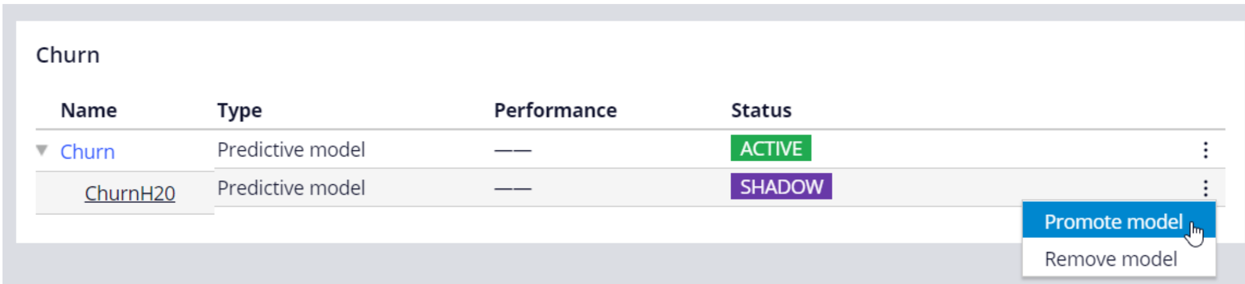
If you deploy the candidate model to production in shadow mode, it runs alongside the original model, receives production data, and generates outcomes, but the outcomes do not impact business decisions.

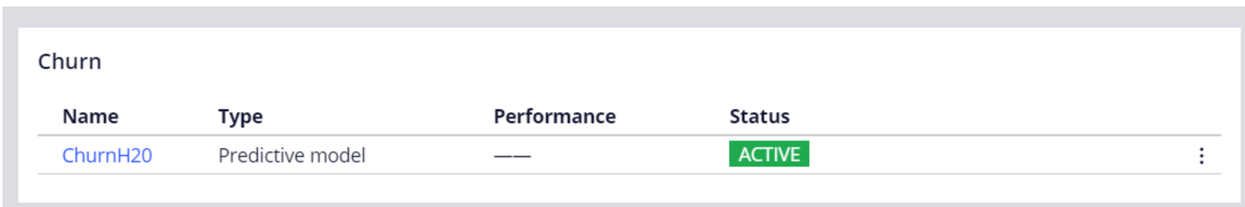
Churn

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

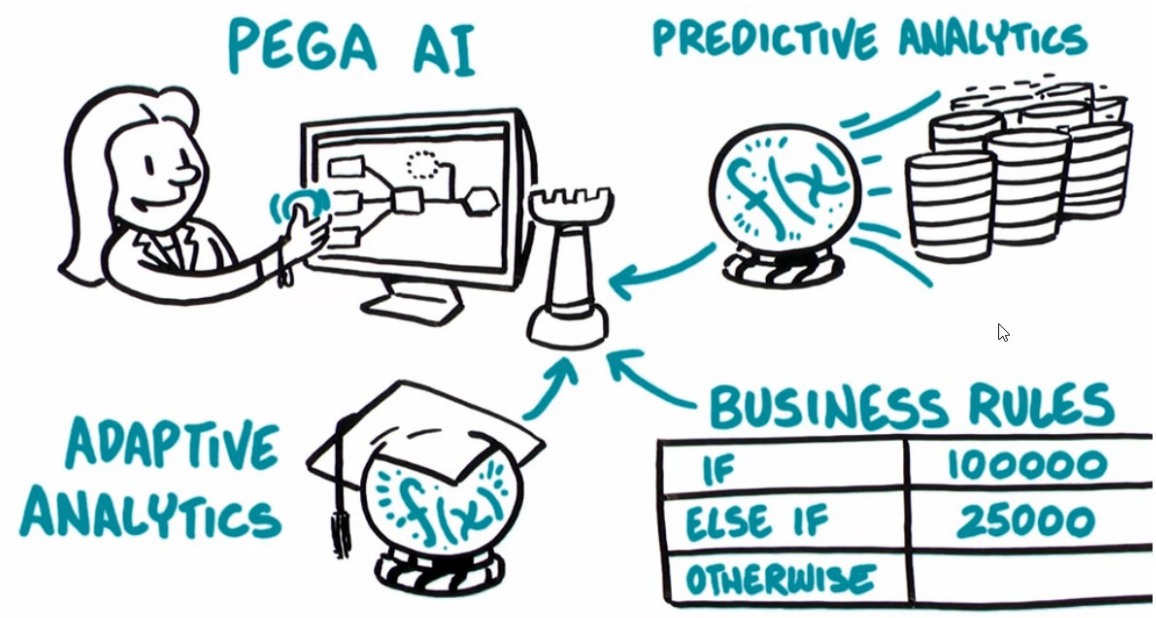
Adaptive analytics

The effectiveness of adaptive models

Applying simple business rules to a sales strategy enables you to identify eligible actions for a customer. However, business rules alone will not enable you to select the best action for a customer, or the action the customer is most likely to accept. As a result, action acceptance rates can be low when only business rules are used to make sales decisions.

> Please be aware that actions can be renamed and are sometimes referred to as propositions or offers.

To improve acceptance rates, augment the business rules in a decision strategy with analytics.



Applying adaptive analytics to your decision strategies enables the strategies to detect changes in customer behavior in real time so that you can act on the changes immediately.

Pega Adaptive Decision Manager

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

Adaptive decisioning continuously increases the accuracy of its decisions by learning from each response to an action. For example, if a customer is offered and then accepts a product, the likelihood that customers with a similar profile also accept that offer increases slightly. The mathematical expressions of these probabilities in the model are regularly updated.

ADM is a closed-loop system that automates the model creation, deployment, and monitoring process. The component can manage a large number of models without human intervention.

In contrast to predictive analytics, which requires historical data and human resources to develop a reliable predictive model, adaptive decisioning can start to calculate who is likely to accept or reject an offer without using any historical information, learning on the fly. Adaptive decisioning captures and analyzes response data in real time, which is useful in situations where the behavior itself is volatile. A typical use case is predicting customer behavior following the introduction of a new offering.

You can use predictive models as an alternative to, or in conjunction with, cases where data is available for offline modeling.

Adaptive decisioning creates binary models and uses these models for predictions. The full adaptive modeling cycle consists of the following steps:

1. Capture response data in real time from every customer interaction.
2. Regularly:
    a. Use sophisticated auto-grouping to create coarse-grained, statistically reliable numeric intervals or sets of symbols.
    b. Use predictor grouping to assess inter-correlations in the data.
    c. Use predictor selection to establish an uncorrelated view that contains all relevant aspects of the action.
    d. Use the resulting, statistically robust adaptive binary model for scoring customers.
3. Whenever new data is available, update the data model.

Adaptive decisioning can also build channel-specific models that account for differences in customer responses to outbound versus real-time inbound offers.

Predictors and outcomes of an adaptive model

Predictors

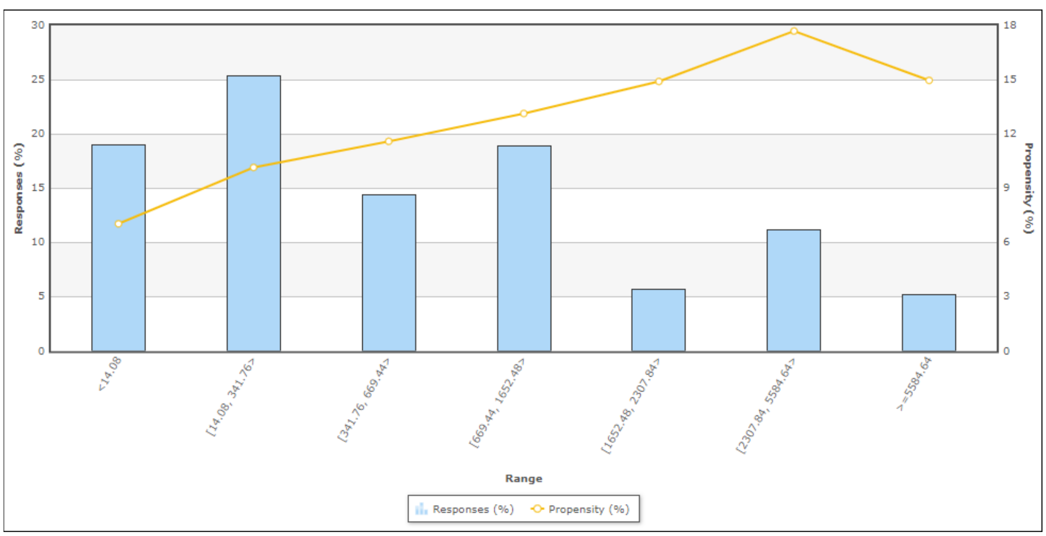
Configuring adaptive models involves selecting potential predictors and setting outcomes that identify positive and negative customer behavior. Unless you are a highly experienced data scientist, it is strongly recommended to leave the advanced settings at their default

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.

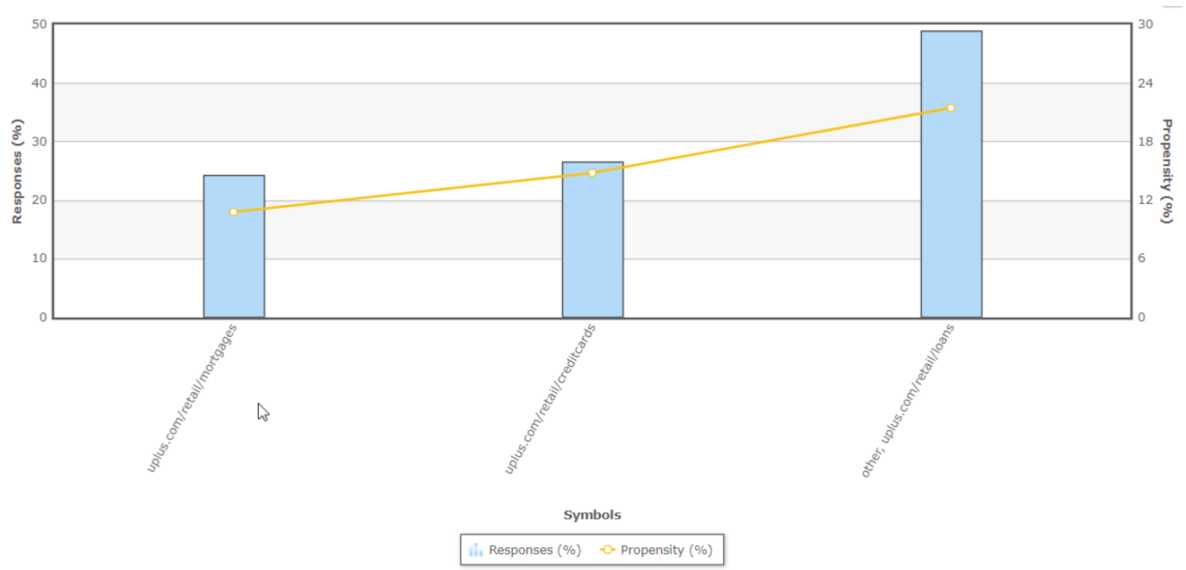
Adaptive Decision Manager (ADM) automatically selects the best subset of predictors. ADM groups predictors into sets of correlated predictors and then selects the best predictor from each group, that is, the predictor that has the strongest relationship to the outcome. In adaptive decisioning, this predictor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



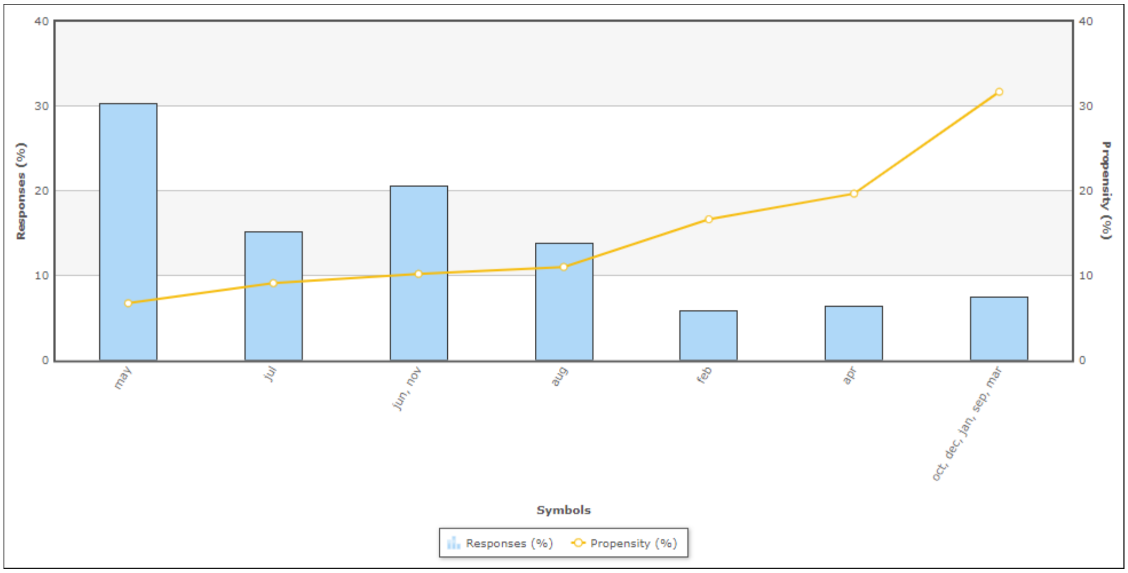
Symbolic data - You can feed predictors with up to 200 distinct string values without any preprocessing. Such data is automatically categorized into relevant value groups, such as the **PreviousWebpage** predictor in the following example. For predictors with more than 200 distinct values, group the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

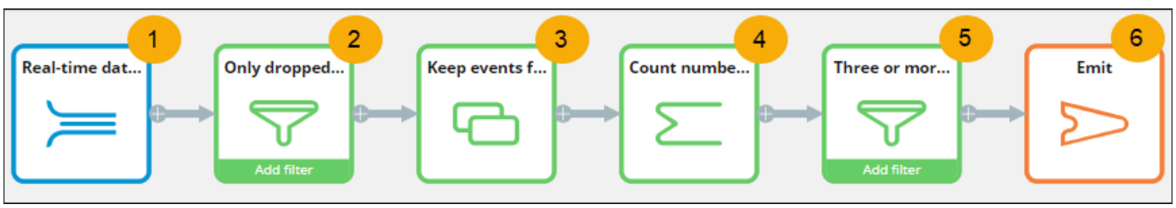
Codes - For meaningful numeric fields, feed code fragments to the model as separate predictors. Simple values require only basic transformation. For example, you can shorten postal codes to the first 2 or 3 characters which, in most countries, denote geographical location.

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:

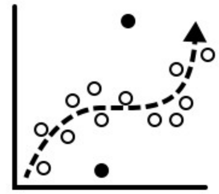
Customer profile data such as age, income, gender, and current product subscriptions. This information is usually part of the Customer Analytic Record (CAR) and is refreshed regularly.

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.

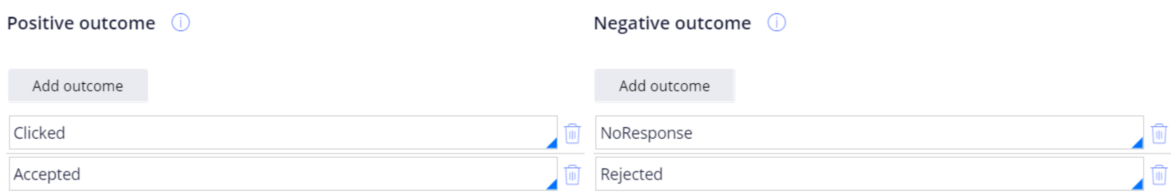


Outcomes

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.



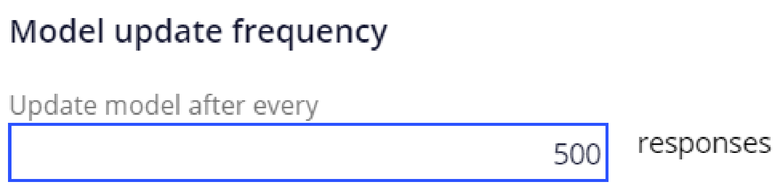
Advanced settings of an adaptive model

Default values

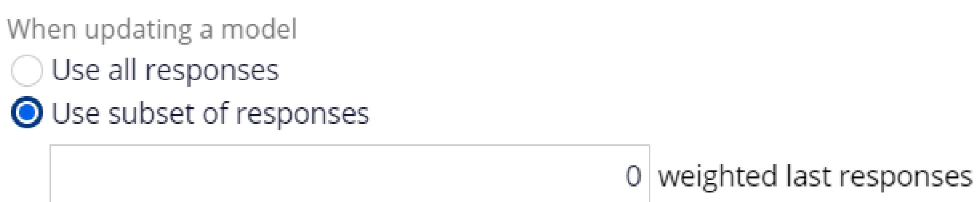
The default values for the adaptive model advanced settings are based on best practices and should only be changed by a highly experienced data scientist.

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.

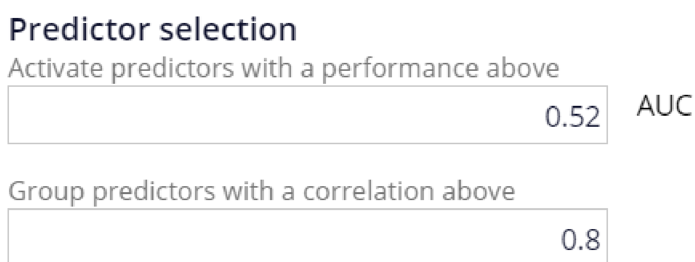
Grouping

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.

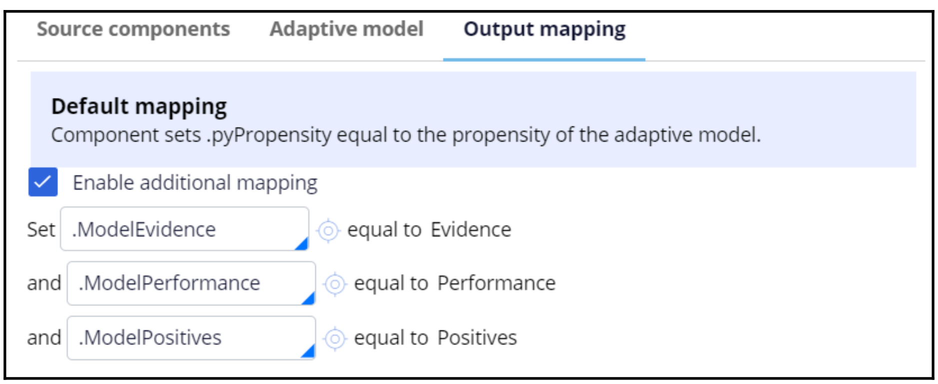
Evidence is the number of responses used in the calculation of the Propensity.

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.



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

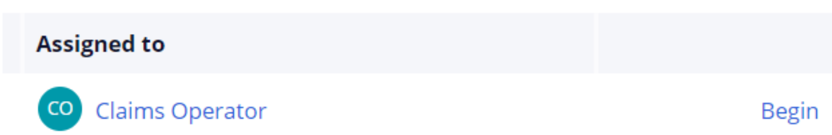
Pega Process AI™ can help to distinguish regular from complex claims. Complex claims often escalate into a lengthy process, which is costly and leads to a bad customer experience. The distinction lets you detect these claims early and address them at once.

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.

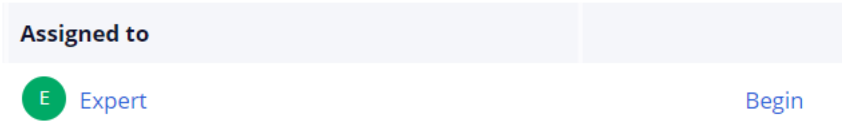
Transcript

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.

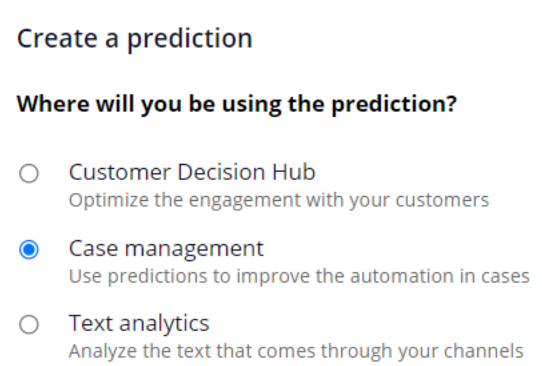


Claims that involve casualties can be very complex and are always routed to an expert instead of a regular claims operator as a precaution.

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.

Create a prediction

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

Prediction name
Predict regular case completion

Outcome
Case completion

Case completion

Predict how likely a case will reach successful completion.

Subject
Car insurance claim

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.

Do you have historical data?

- (•) I do not have historical data
- ( ) I have historical data

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.

Response labels for Case completion

Target label	Alternative label
Resolve	Fail

Prediction output

Predicting Case completion for each Stage Name

Data mapping

Outcomes mapped to 'Resolve'
Resolved-Completed
Resolved-Rejected

Outcomes mapped to 'Fail'
Resolved-CX-Completed
Resolved-CX-Rejected

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.

Case completion

Name	Type	Performance	Status
Case completion	Adaptive model	——	ACTIVE

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.

- Current page (EscalateApp)
  - Page Claim
  - Page Customer
  - Page Policy

☐	Name	Data type
☑	ClaimFrequency	Integer
☑	ClaimRatio	Decimal
☐	EndDate	DateTime
☑	EntitledAmount	Decimal
☐	PolicyID	Text
☐	StartDate	DateTime
☑	SuspiciousClaim	Decimal

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.

When

Custom condition

(Casualities is false)

Go to

Regular process

Add path

Otherwise go to

To 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.

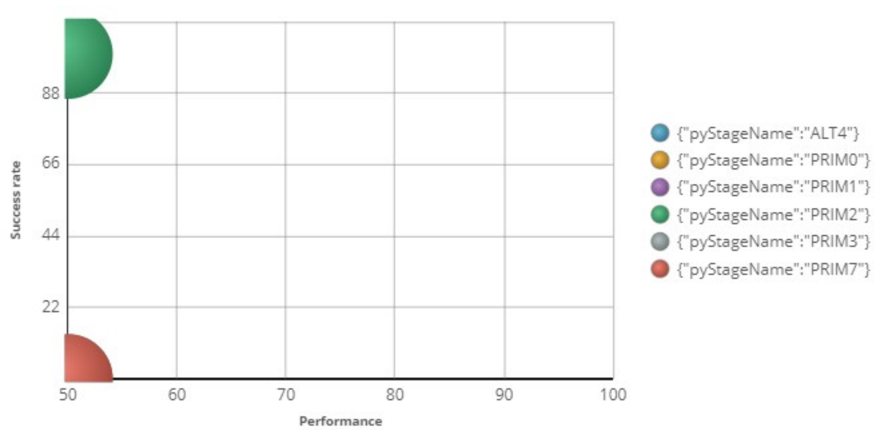
Probability | is greater than | 0.2

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.

CREATE > DETECT FRAUD > CLAIM PROCESS > COMPLEX CLAIM > RESOLVED-CX-COMPETED

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.

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

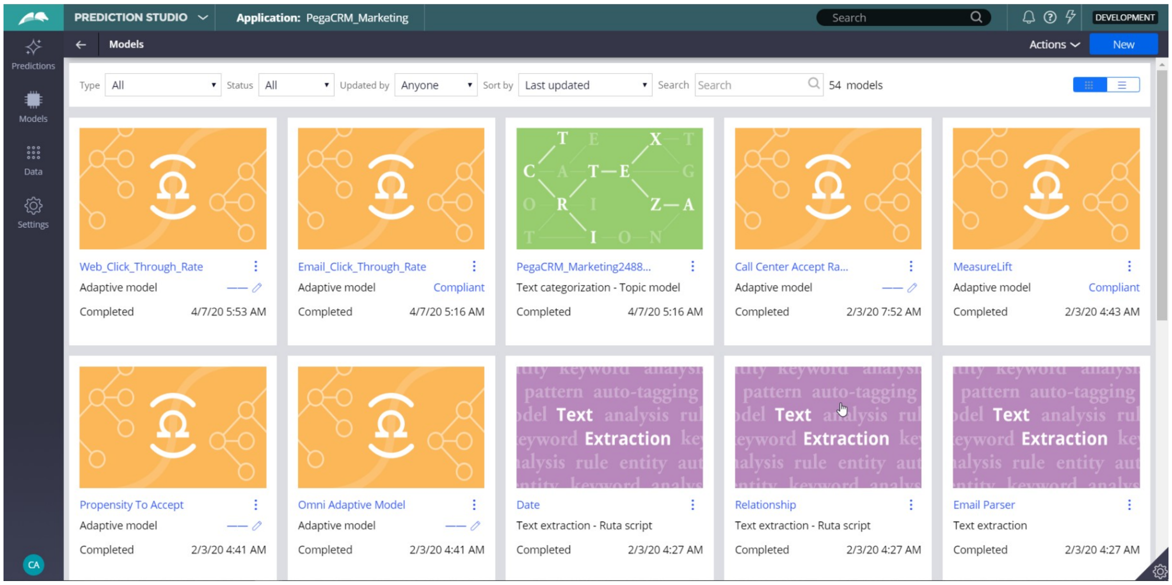
Introduction

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

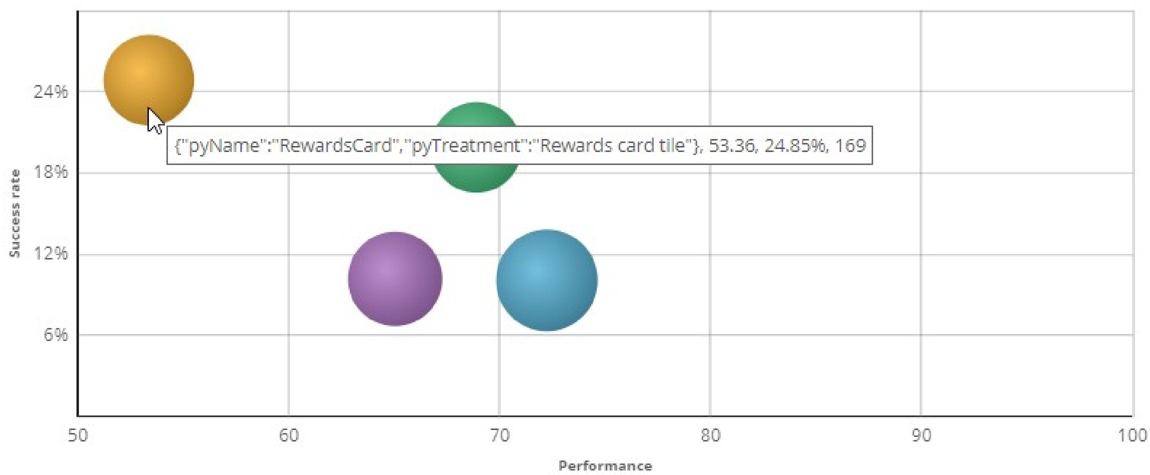
This demo will show how to inspect the health of your adaptive models and their predictors. This is a regular data scientist task.

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.



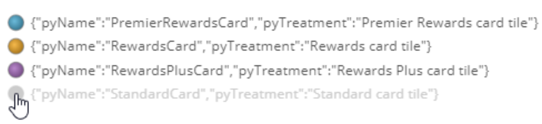
Each bubble represents the model for a specific action.

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.

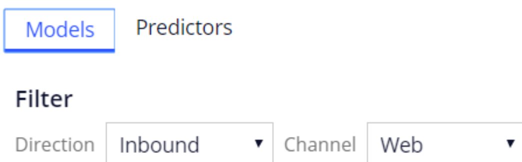
The information you see here is extracted from the Adaptive Data Mart, which is a reporting view of the Adaptive Decision Manager (ADM) server.

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

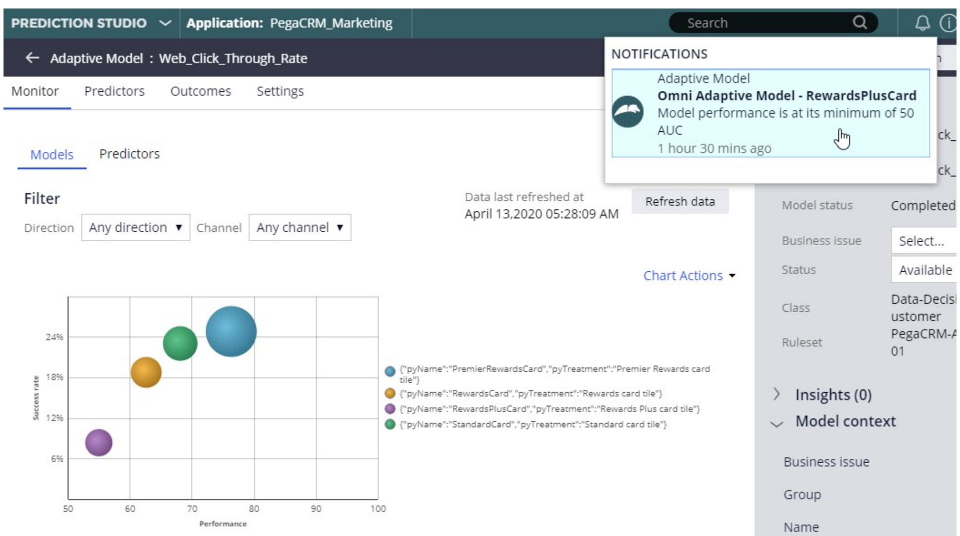
This means that the data as reported may not be the same as the data you see when you view it in real-time. You can refresh the view to synchronize the data.

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.

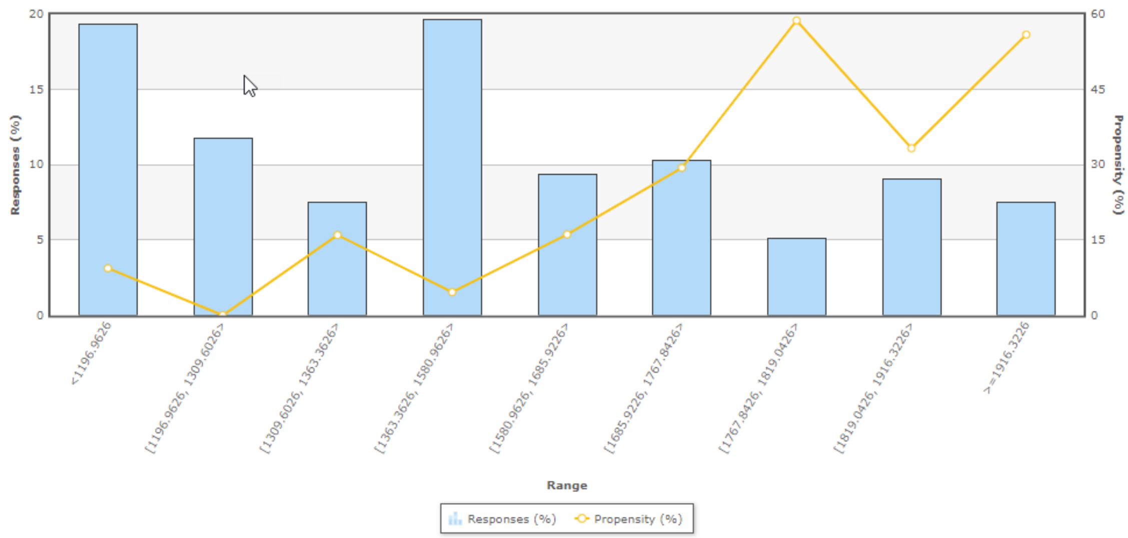
Predictors	Status	Type	Performance (AUC)	Range/Symbols(#)	Bins(#)
Customer.AverageSpent	Active	Numeric	80.86	[1001.33; 1997.33]	9
Customer.Age	Active	Numeric	75.54	[19.0; 80.0]	9
Customer.InteractionContext.PreviousWebpage	Active	Symbolic	74.22	4.00	4
Customer.AverageBalance	Active	Numeric	73.51	[506.21;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

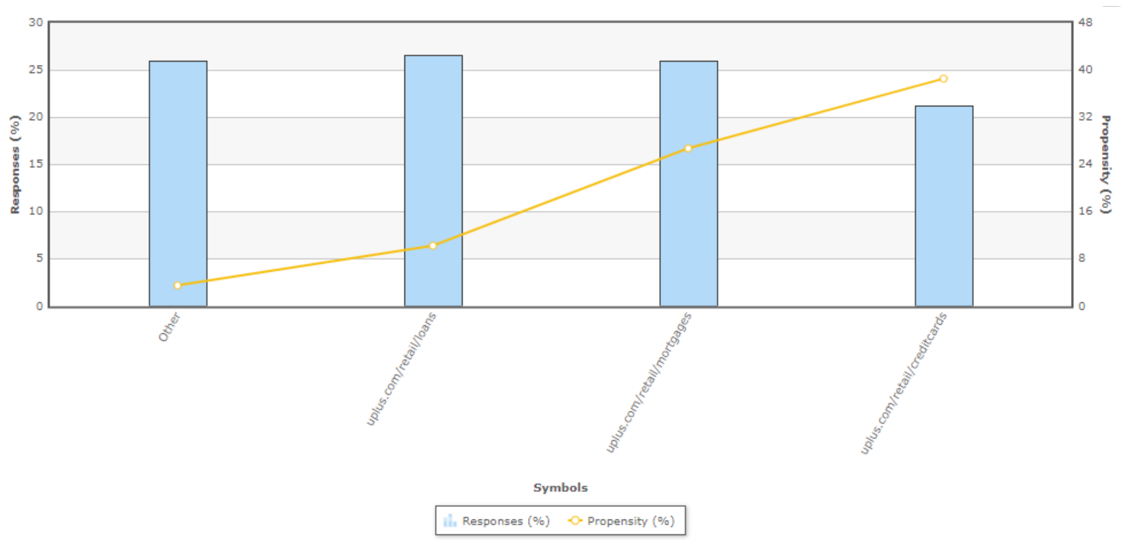
The system split the AverageSpent predictor into 9 bins. Each bin has its own offer propensity.



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

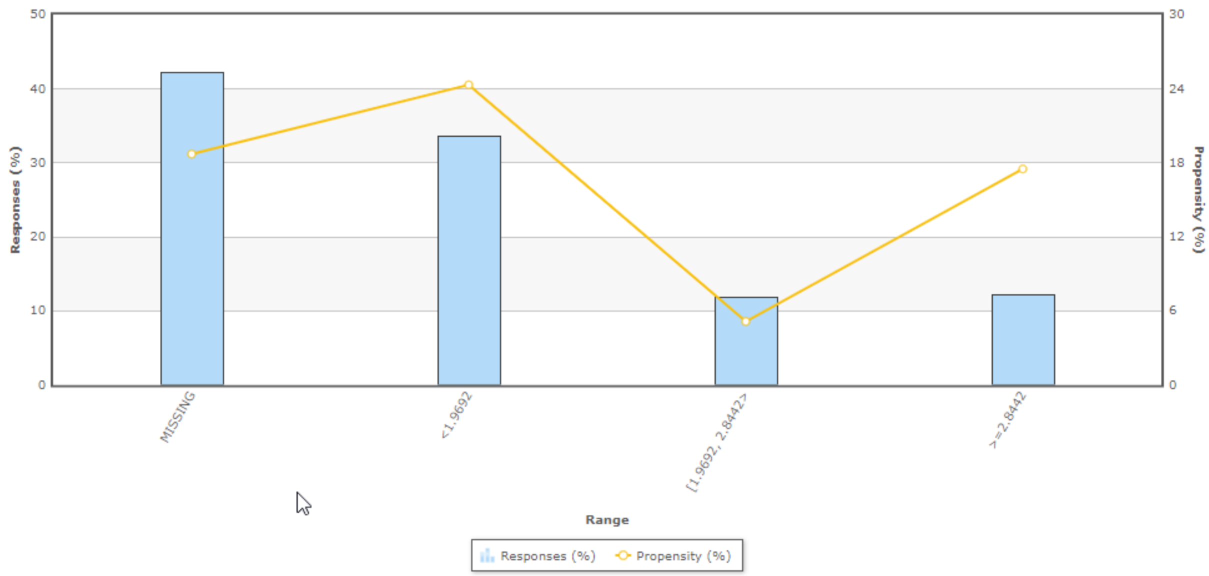
Now, let's examine the symbolic PreviousWebpage predictor.

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.

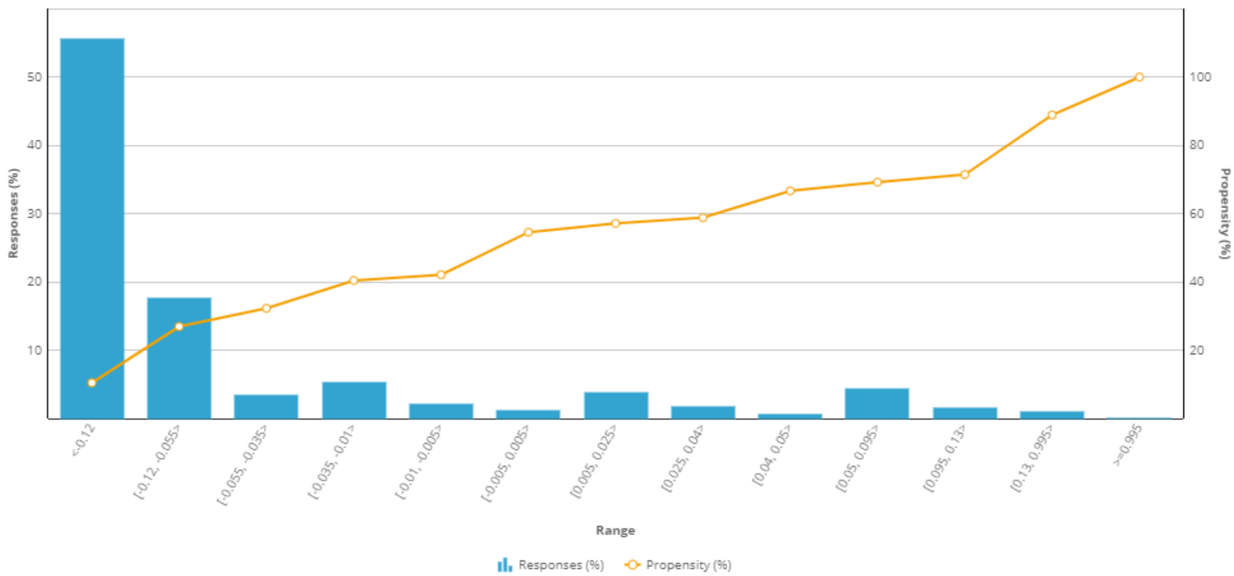
In this example, the adaptive system established that the number of days since the offer was accepted is a well-performing predictor.



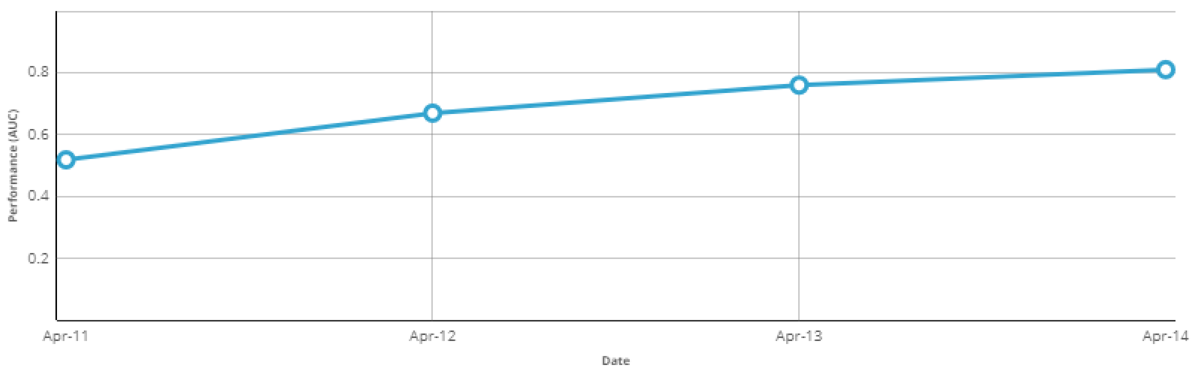
The system grouped three predictors that are correlated. It then marked two of them as inactive. Inactive predictors are not used in the propensity calculation.

▼ 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

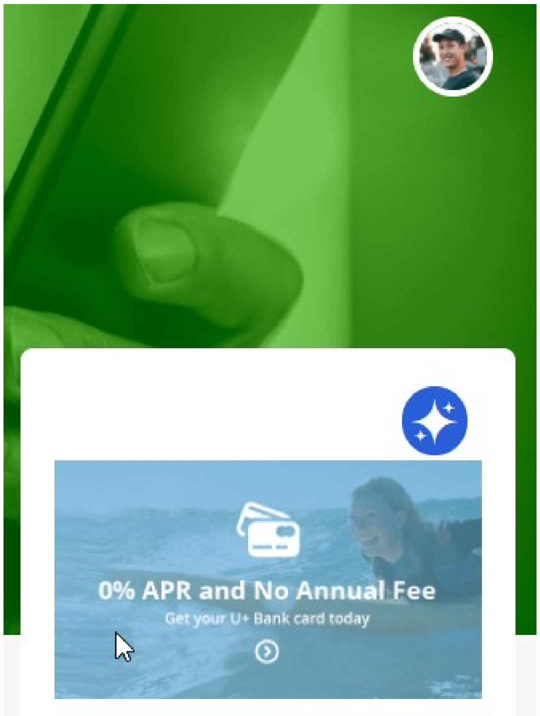
Introduction

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 customers are eligible for multiple credit cards, adaptive models decide which card to show them.

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

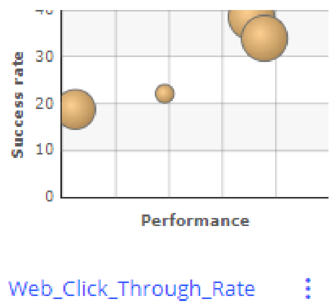
When the customer clicks on the banner, the model regards this as positive behavior.

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.

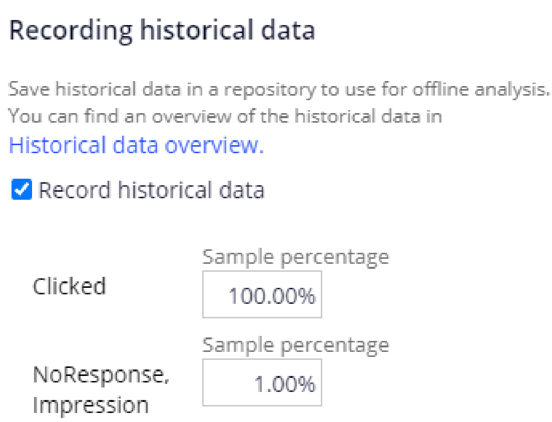
The adaptive model that drives the decision about which banner to display is the **Web Click Through Rate** model.



To extract the data, 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.



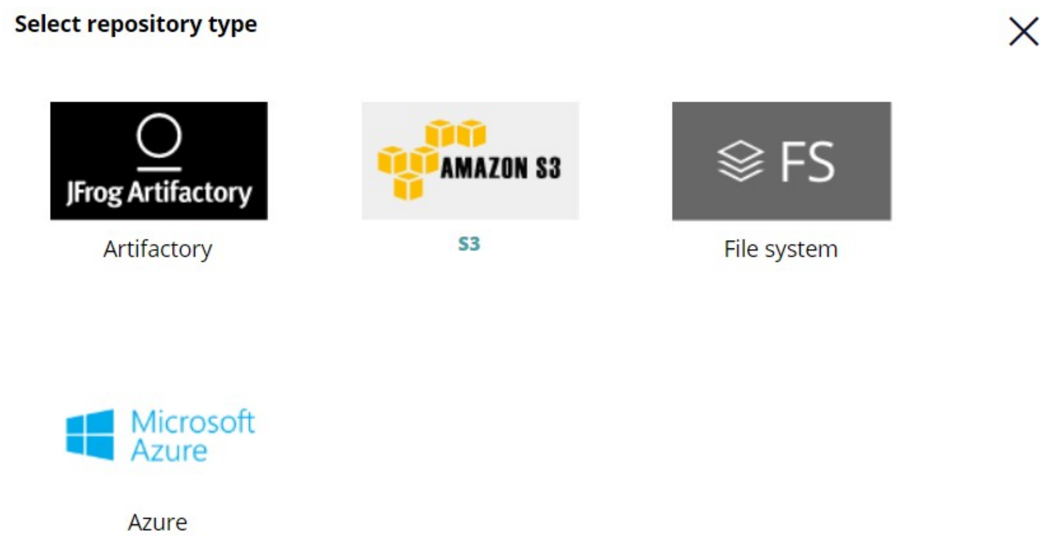
The sample percentages determine the likelihood that a customer response will be recorded.

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

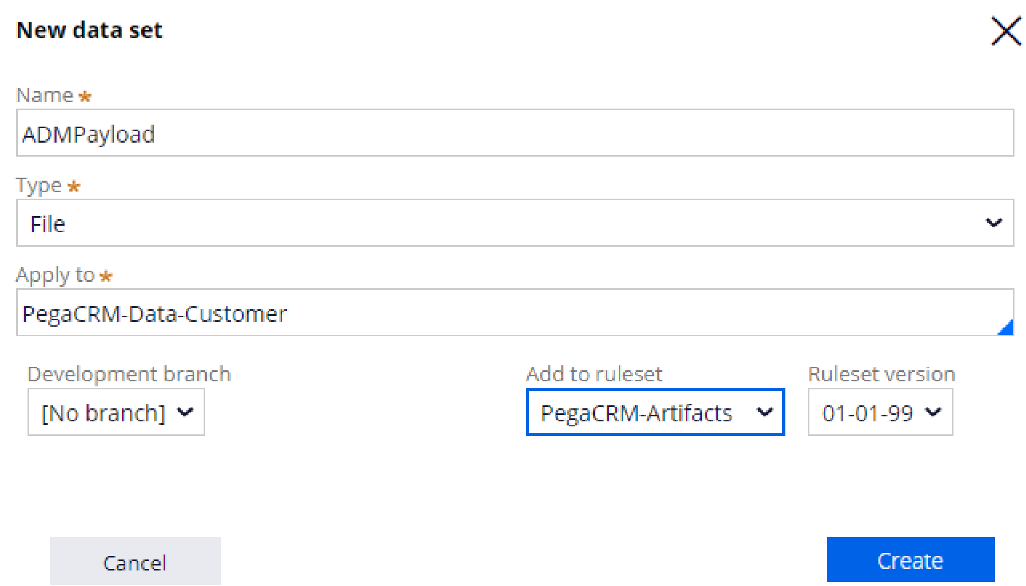
By default, the data is stored for 30 days in the **defaultstore** repository.

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

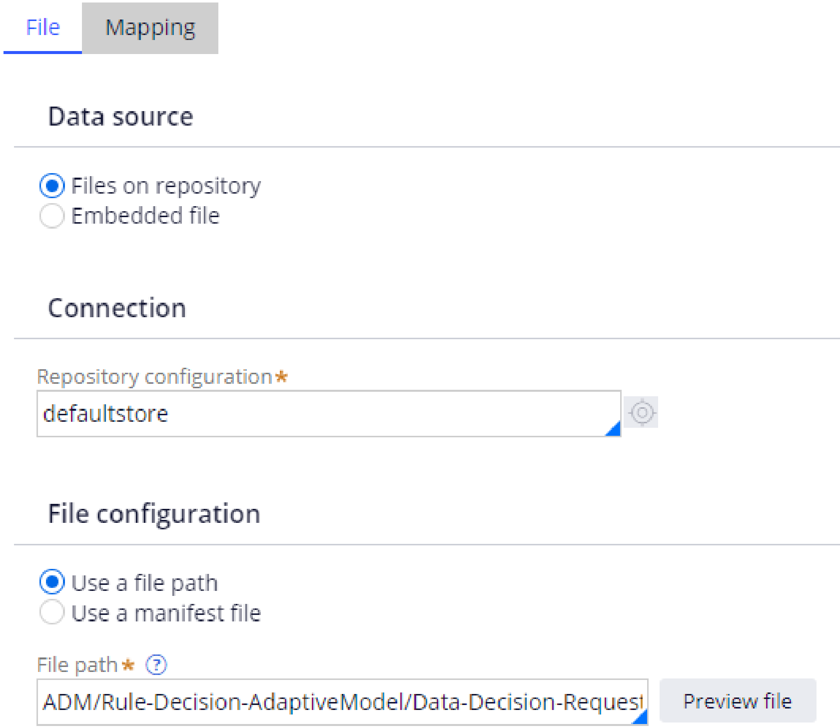
Supported repository types include Microsoft Azure and Amazon S3.



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

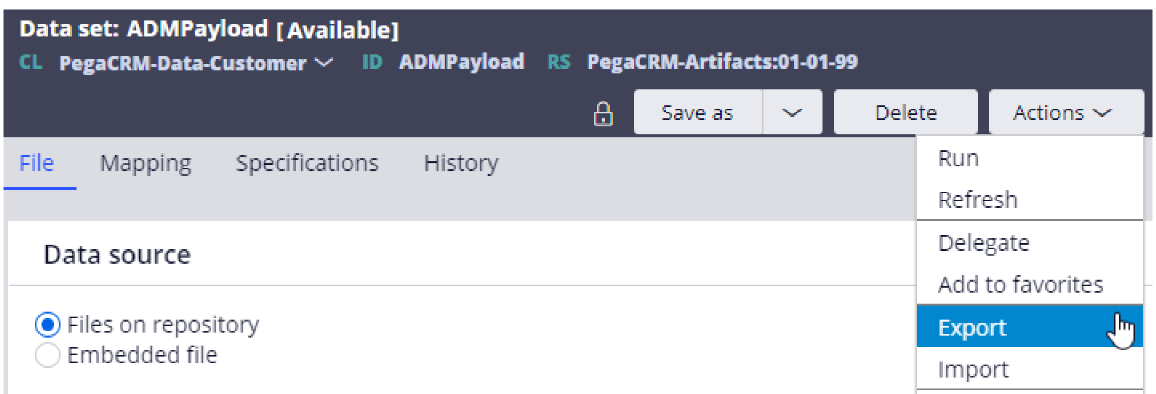


The data set is mapped to the file that contains the recorded historical data.



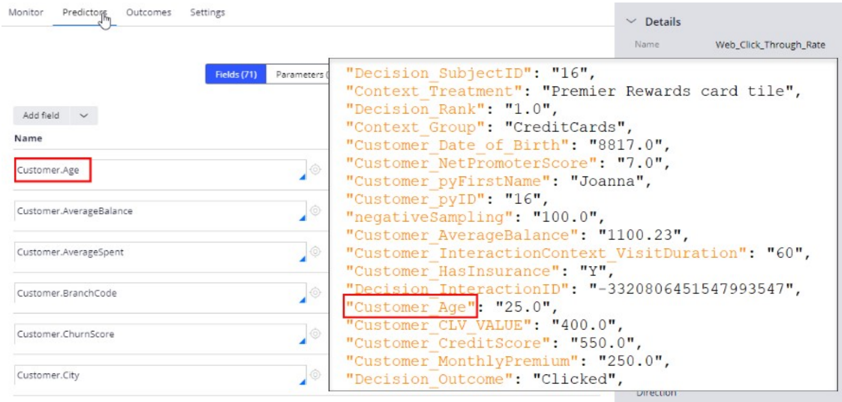
With these settings in place, the input data used for the prediction and associated outcome are stored in the configured data set when customers see an offer and click on an offer.

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 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.