



# Pega Process AI essentials

## STUDENT GUIDE

**© Copyright 2021**  
**Pegasystems Inc., Cambridge, MA**  
All rights reserved.

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

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

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

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

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

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

**Mission:** Intelligent case management

**Product:** Pega Customer Decision Hub™ 8.6

**URL:** <https://academy.pega.com/mission/intelligent-case-management/v1>

**Date:** 18 October 2021

# Contents

Pega Process AI overview.....	4
Pega Process AI overview.....	5
Prediction Studio.....	9
Predictive models drive predictions.....	12
Predicting fraud.....	17
Importing predictive models.....	18
Using machine learning services.....	23
Predicting fraud.....	30
Upgrading predictions.....	36
Model operations process.....	37
Placing a predictive model in shadow mode.....	43
Predicting case completion.....	46
Adaptive analytics.....	47
Predictors and outcomes of an adaptive model.....	49
Predicting case completion.....	53
Monitoring adaptive models.....	58
Regular monitoring of adaptive models.....	59
Inspecting adaptive models.....	61
Exporting historical data.....	68

# 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

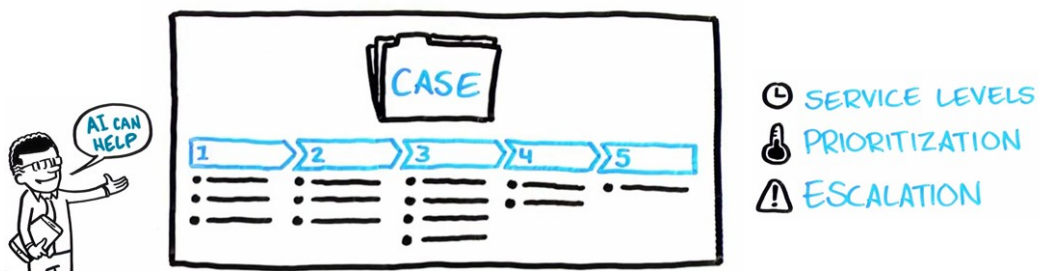
## Introduction

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.

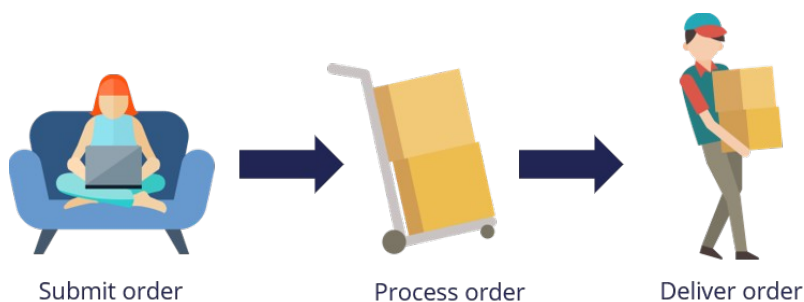
## Transcript

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.

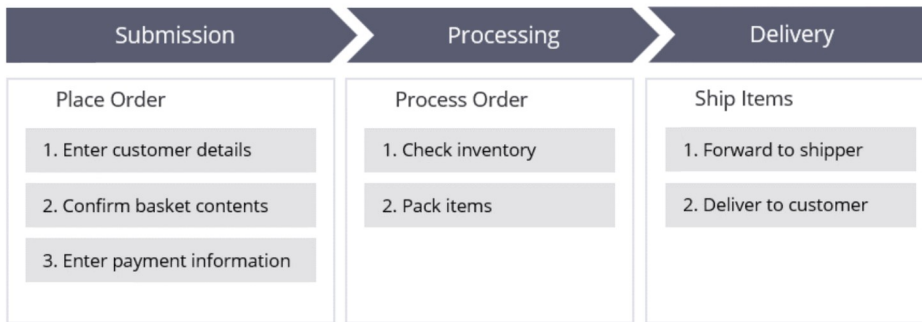


An Pega Platform application that models the online order process follows the same sequence as a series of stages. A **case type** is the abstract model of that process.

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

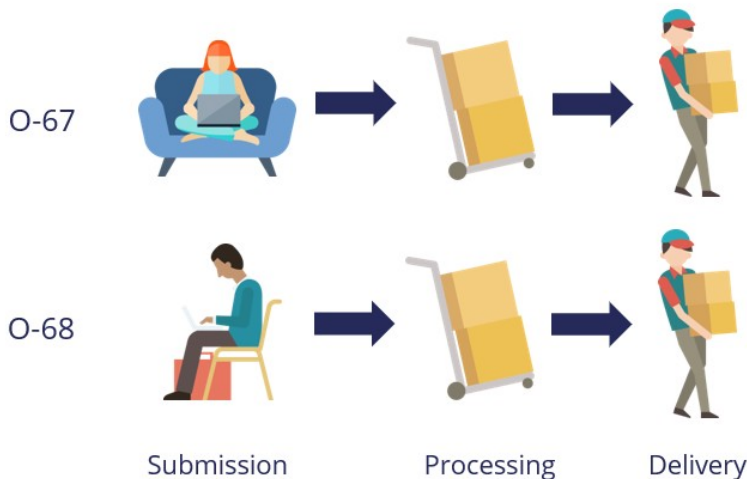
The **case life cycle** for a case type helps to visualize the work to complete as part of a business transaction.

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.



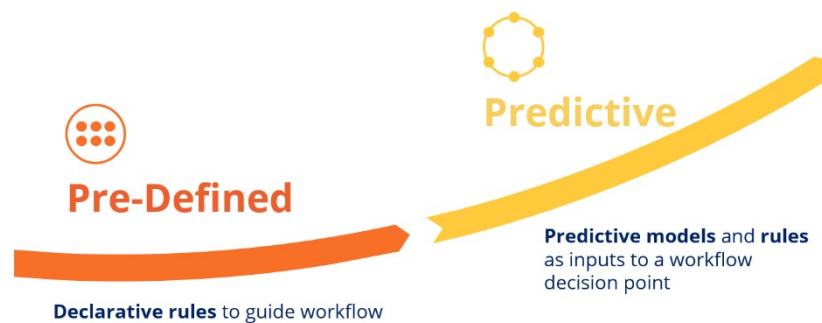
## Pre-Defined

Declarative rules to guide workflow

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

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

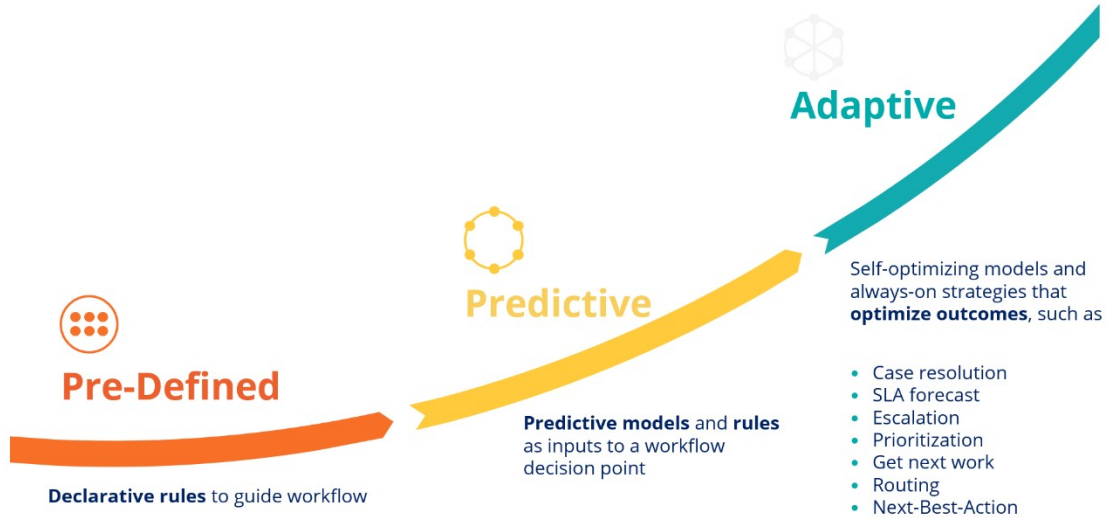
For example, the decision not to process an order can be based on a high credit risk score, and then the application can route the dubious claim for closer inspection.



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.

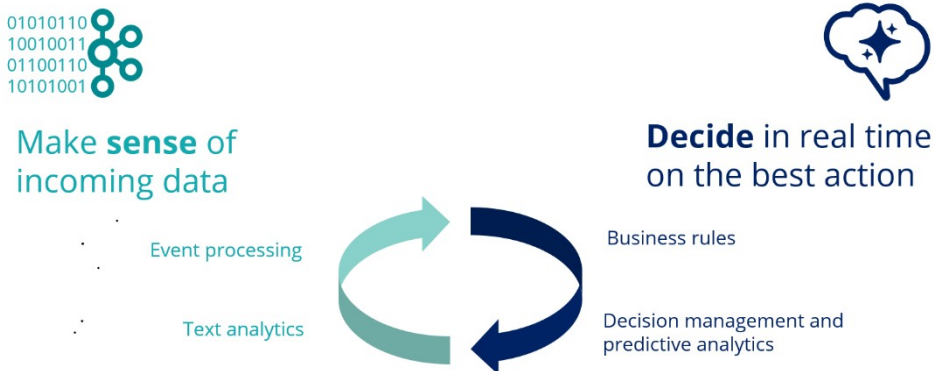
To optimize case outcomes, use adaptive models that can predict outcomes, such as case resolution, or intelligently prioritize and route cases to optimize business value and customer experience.



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.



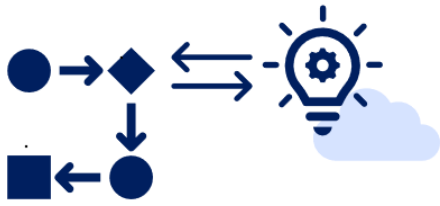
The decision is based on the business rules and supported by predictive analytics. This process is repeated every time that the case requests a decision.

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

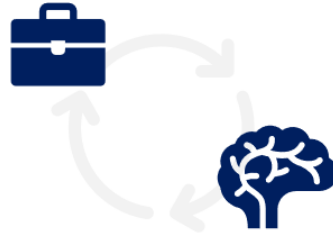
To summarize, Pega Process AI uses artificial intelligence in case management to produce better business outcomes.

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**

# Prediction Studio

## Introduction

The AI and automated decisioning in Pega Platform™ is a fundamental element of Center-out™ business architecture. A lead system architect (LSA) must be aware of this feature and understand what constitutes the centrally managed brain of Pega. Ideally, an LSA is familiar with decisioning functionality and terminology.

Prediction Studio is a Pega Platform workspace in which users create prediction models. For example, data scientists or decision analysts design and create analytical models. The analytical models created in Prediction Studio can also be used with Dev Studio and App Studio as part of the case type life cycle for decision-making, such as when rules.

Adding the pxPredictionStudio portal to your access group gives you access to Prediction Studio; however, the appropriate role should be added to perform the desired operations.

Prediction Studio has four work areas: Predictions, Models, Data, and Settings. The workspace also provides data scientists everything they need to author, deploy, govern, monitor, and change AI models.

## Work areas in Prediction Studio

### Predictions

This landing page is used to manage or create predictions. The new prediction can be one of the three types, as shown in the image below. For example, you can use the Case management prediction in the case life cycle for decision making. Models drive predictions.

### Create a prediction





**Where will you be using the prediction?**

- Customer Decision Hub  
Optimize the engagement with your customers
- Case management  
Use predictions to improve the automation in cases
- Text analytics  
Analyze the text that comes through your channels

## Model

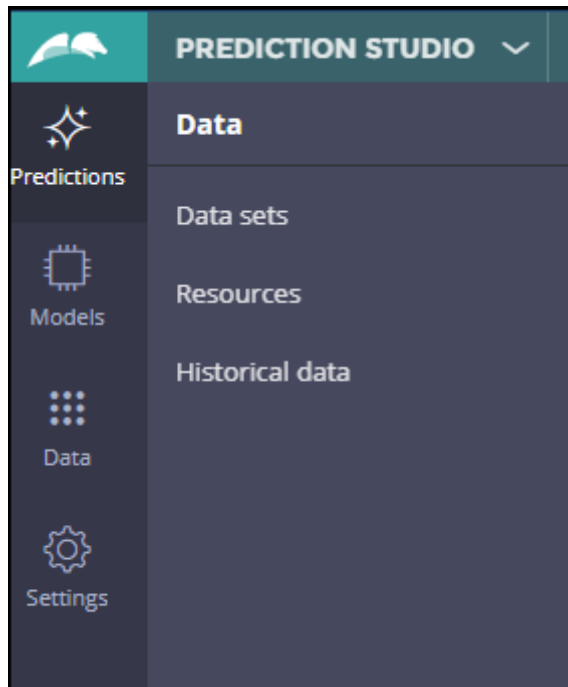
This landing page defines the models: predictive, adaptive, text categorization, and text extraction. You use models to make predictions. Once created, models can be self-learning; these models are included in the decision strategies to fulfill the customer needs and achieve the business goals.

### Create model

			
<b>Predictive model</b>	<b>Adaptive model</b>	<b>Text categorization</b>	<b>Text extraction</b>
Predict customer behavior such as offer acceptance, churn rate or credit risk based on customer data.	Predict customer behavior using self-learning models.	Analyze and assign text to a specified category.	Analyze unstructured text to extract required words or phrases.

## Data

This work area is used to define data sets, resources, and historical data.



The screenshot shows the Prediction Studio interface. At the top, there is a teal header with a white icon of two hands and the text "PREDICTION STUDIO" with a dropdown arrow. Below the header is a dark blue sidebar with four main sections: "Predictions" (with a star icon), "Models" (with a chip icon), "Data" (with a grid icon), and "Settings" (with a gear icon). The "Data" section is expanded, showing a list of options: "Data", "Data sets", "Resources", and "Historical data".

In the **Data sets** work area, you define a collection of records. The collection can be from different sources and formats. For example, a data set instance can be as simple as sourced from a database table, or it can be sources from stream services or even Social media sources like Twitter, YouTube. Once a data set is configured, you can read, insert, and remove the records through the data management operations.

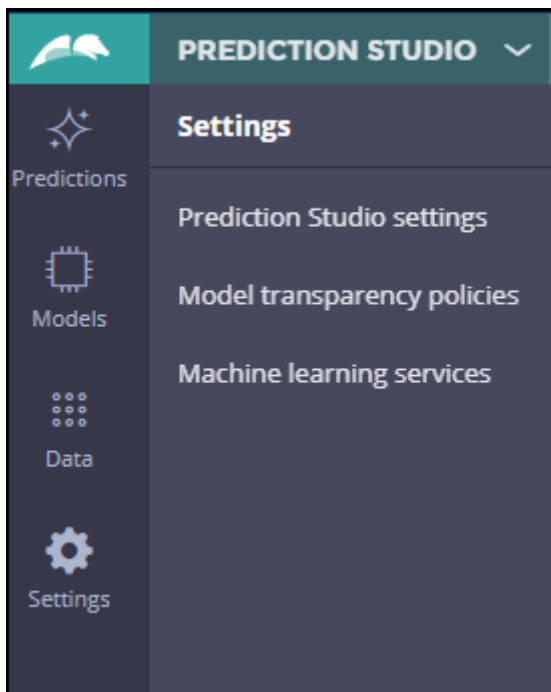
Data sets are referenced in data flows and are mostly used in Pega Customer Decision Hub™ applications. Data sets can be executed programmatically using the DataSet-Execute method in an activity.

In the **Resource** work area, you can access resources such as taxonomies or sentiment lexicons that provide features for building machine learning models.

The **Historical data** work area is where you upload the historical data from local file systems to train the models.

## Settings

You can manage the Prediction Studio settings in the Settings work area.



For example, you can set the work queue for the data scientist, language of operations, the internal database, and set the license key of the H2O Driverless AI models.

Model transparency policies define the transparency threshold for each business issue and provide scores for each predictive model type in Customer Decision Hub applications.

This work area can connect to third-party machine learning platforms. Pega Platform currently supports Google AI Platform and Amazon SageMaker models.

# Predictive models drive predictions

## Introduction

Learn how to better address your customers' needs by predicting customer behavior and business events with predictions. For example, you can determine the likelihood of customer churn or chances of successful case completion.

Predictions combine predictive analytics and best practices in data science.

## Transcript

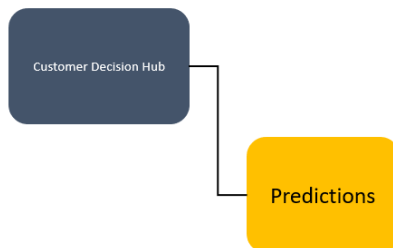
This video shows you how predictions can predict events in a business activity.

You can create and use predictions for several use cases.

Pega Customer Decision Hub™ predictions are used in decision strategies to optimize engagement with customers.

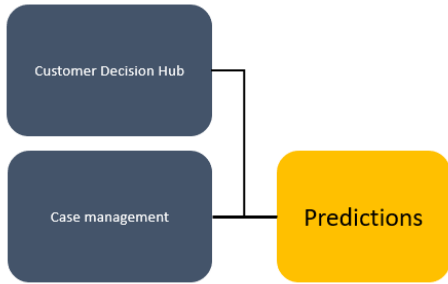
The most common predictions are provided out of the box. You can use the default predictions or create your own.

For example, you can use predictions to predict whether customers are likely to accept your offer or cancel a subscription.

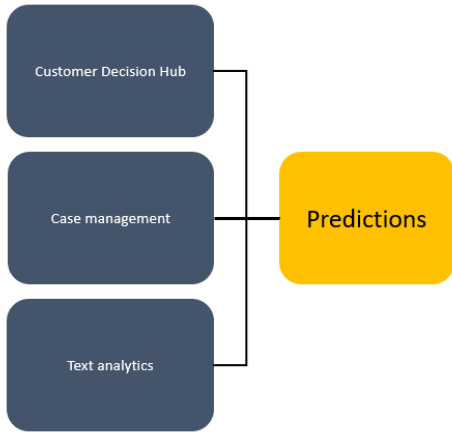


Case management predictions are used in case automation to route cases or prioritize work.

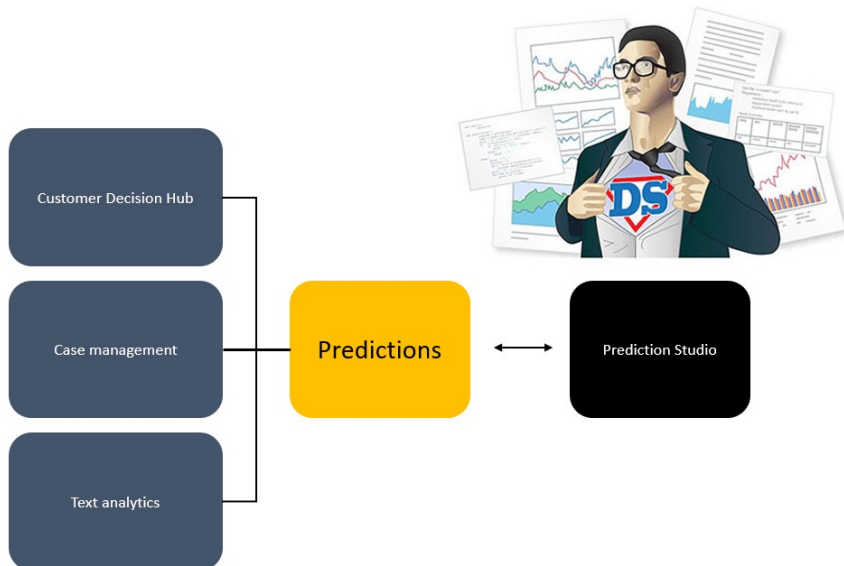
For example, you can create a prediction that predicts whether a case is likely to be completed successfully, or the probability that a claim is fraudulent, which might be useful in straight-through processing of claims with a low score.



Text analytics predictions analyze the text that comes through your channels, such as email or chat, to predict the topic or sentiment of the text.



Predictions are created and managed in Prediction Studio, the dedicated workspace for data scientists.



A next-best-action designer can include the predictions in decision strategies in Customer Decision Hub, to better adjust to customer needs and achieve business goals at the same time.

This handover of predictions from the data scientist to the next-best-action designer contributes to a separation of concerns.

Predictions combine predictive analytics and best practices in data science.

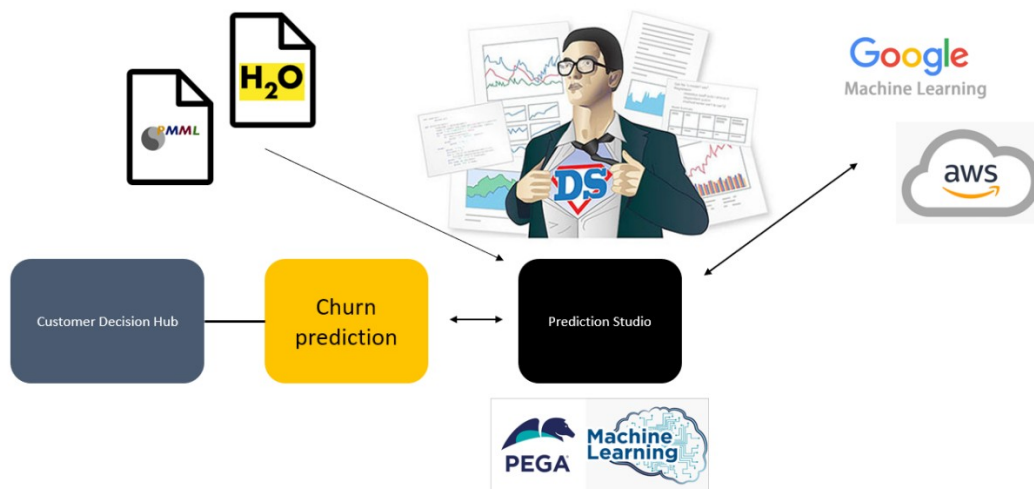
A predictive model drives a prediction. A data scientist can replace the predictive model at any time. However, the prediction always predicts the same outcome.

For example, you can create a new Customer Decision Hub prediction that calculates the likelihood that a customer might end a subscription soon.

To drive the prediction, a data scientist creates a predictive churn model on a platform of choice.

Prediction Studio supports the PMML and H2O model file formats. It can connect to machine learning services like Google ML and Amazon SageMaker.

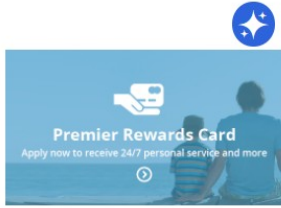
You also have the option to use Pega machine learning.



One of the predictions that is shipped with Customer Decision Hub, **Predict Web Propensity**, calculates the likelihood that a customer clicks on a web banner.

Customer Decision Hub decides which banner to show based on the calculated propensity and weighting in business requirements and context.





**Premier Rewards card**

Receive 24/7 personal service and more

[Learn more](#)

In this case, the Premier Rewards card has a propensity of 0.8, and it is ranked as the next best action.

The **Predict Web Propensity** prediction is driven by an adaptive model that learns from each customer interaction.

Best practices in data science include the use of a control group. Customers in the control group are shown a random banner instead of one based on the calculated propensity.

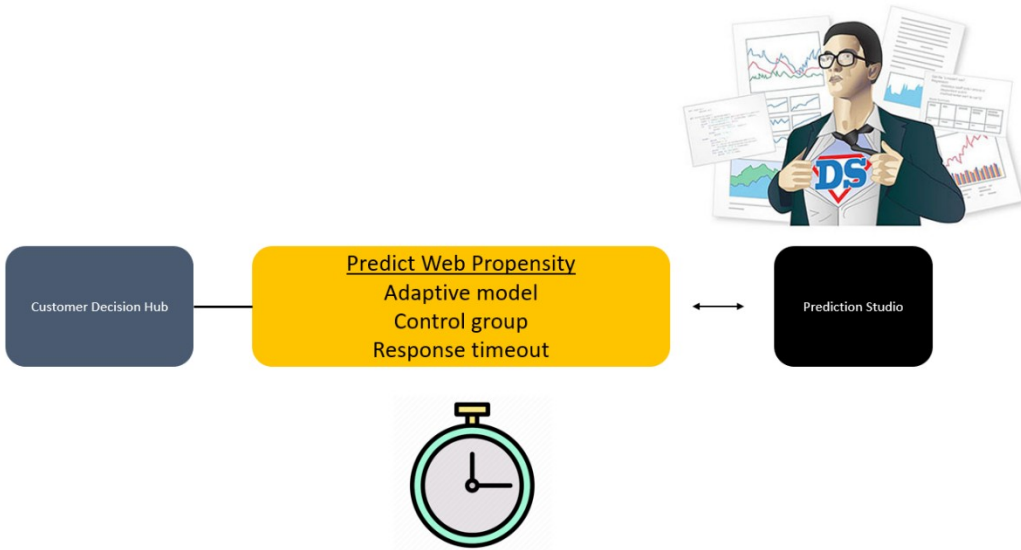
Comparing the success rate in the control group and the test group measures lift, which is the boost in success rate that the prediction generates. Lift is an important business metric.

Also, the use of a control groups allows continuous exploration by the predictive models.

When the lift drops significantly over time, Prediction Studio notifies the data scientist.



The response timeout is built into predictions. When the timeout expires, NoResponse is automatically recorded as the outcome of the interaction.



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

- How predictions can predict events in the business activity
- How a prediction uses a control group to measure the boost in success rate that is generated by the prediction
- How NoResponse is automatically recorded as the outcome of the interaction when the response timeout expires

# 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

# Importing predictive models

## Introduction

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

## Transcript

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

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

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

### Create model



Predictive model

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



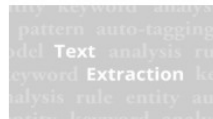
Adaptive model

Predict customer behavior using self-learning models.



Text categorization

Analyze and assign text to a specified category.



Text extraction

Analyze unstructured text to extract required words or phrases.

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

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

## New predictive model

Name \*

ChurnPMML

Create model ?

Use Pega machine learning

Import model

Select external model

Import model file \* ?

Choose File

File name

ChurnPMML.pmml

Context

Customer [Change](#)

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

## New predictive model

Outcome definition [Set labels](#)

The objective of the model is to predict  
Segment

Predict the probability of

churned

loyal

Modeling technique

Tree model

Expected performance (AUC) ?

80

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

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

Create missing fields

Refresh mapping

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

Run predictive model

Troy Data Transform

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

Run

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

Run predictive model

Single run Batch run

> Inputs

< Outputs

Results

Result	Monitoring performance
churned	0
Propensity	Monitoring evidence
0.9342621091861922	0.0

Output	Value
Segment	churned

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

## Run predictive model

Single run **Batch run**

> Inputs

∨ Outputs

### Results

Result	Monitoring performance
loyal	0
Propensity	Monitoring evidence
0.3583554398897344	0.0

Output	Value
Segment	loyal

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

## Run predictive model

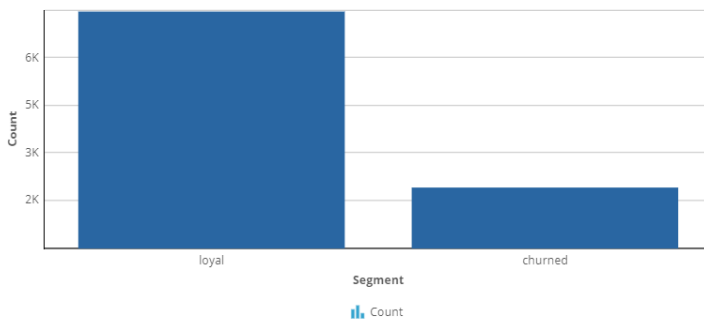
Single run **Batch run**

Data source\*  
CustomerBatch  
This data source contains approximately 10,000 records.

Source type  
Data set

Total records executed: 10000 Total failed: 0

Output  
Segment



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

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

# Using machine learning services

## Introduction

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

## Transcript

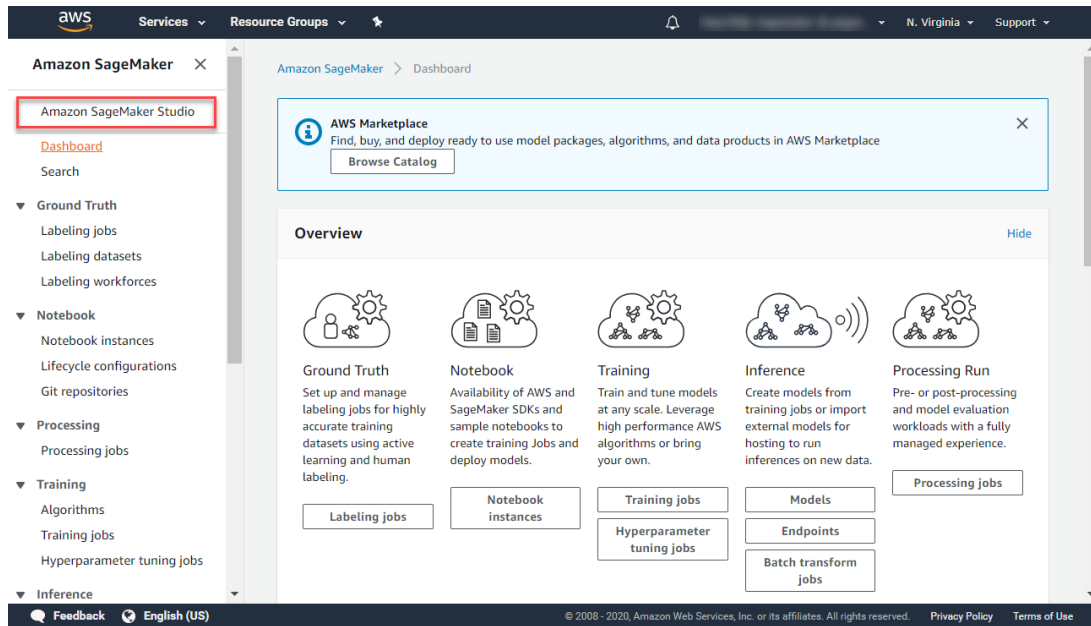
This demo will show you how to leverage a machine learning service by running a churn model created externally and using its outputs in Pega Prediction Studio.

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

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

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





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

### Create Amazon SageMaker Autopilot Experiment

**JOB SETTINGS**

Experiment Name

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

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

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

S3 bucket name

S3 object key prefix

Is your S3 input a manifest file?  
 For more information on the format of a manifest file, please see the [AWS Docs](#)

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

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

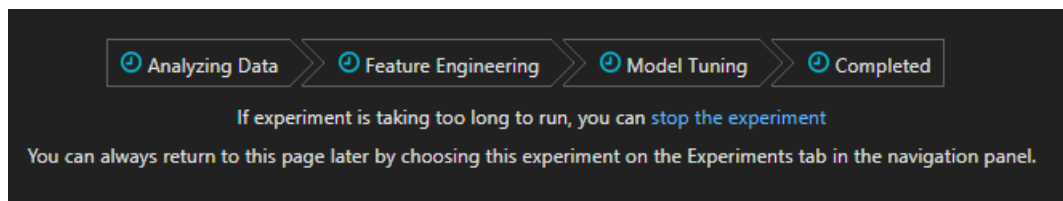
Output data location (S3 bucket)  
 Enter the location in S3 where you want to store the output.

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

S3 bucket name

S3 object key prefix

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



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

EXPERIMENT: CHURNAWS Open candidate generation notebook Open data exploration notebook

Trials Job profile

TRIALS

1 row selected Deploy model

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

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

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

**Deploy model**

**REQUIRED SETTINGS**

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

Instance type  Instance count

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

Save prediction requests  
 Save prediction responses

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

**ADVANCED SETTINGS - Optional**

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

## AmazonML



Service type  
Amazon SageMaker

Name  
Amazon Machine Learning

Authentication profile \*  
AmazonML ▼ Type  
AWS

Region  
US East (N. Virginia)

Cancel

Save

## Test machine learning service



Successfully connected to the machine learning service.

Close

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

### New predictive model

Name \*  
ChurnSageMaker

Create model ?

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

Machine learning service \*   Model  
Amazon Machine Learning ▼   SageMakerChurn-model ▼

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

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

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

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

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

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

### New predictive model ✕

**Outcome definition**

The objective of the model is to predict  
Churn

Predicting  
Two categories

Predict the probability of  
churned

With alternative outcome  
loyal

Modeling technique  
xgboost

Framework  
scikit-learn

Expected performance (AUC) [?](#)

Back Cancel Create

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

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

#### ∨ Outputs

##### Results

Result	Monitoring performance
0.9071381688117981	0
Propensity	Monitoring evidence
---	0.0

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

#### ∨ Outputs

##### Results

Result	Monitoring performance
0.0010575958294793963	0
Propensity	Monitoring evidence
---	0.0

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

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

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

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

# Predicting fraud

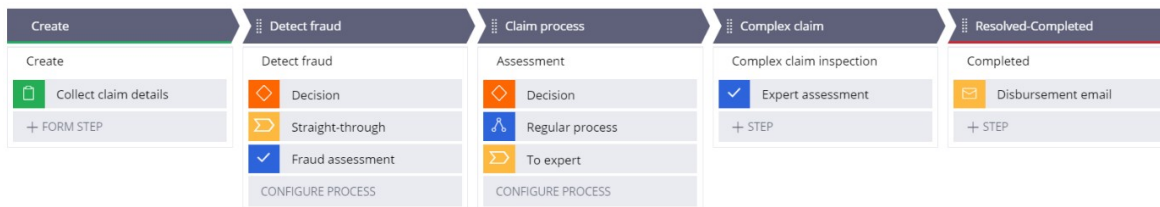
## Introduction

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

## Transcript

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

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



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

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

When  
Custom condition

(ClaimedAmount is less than 100)

Go to  
Straight-through

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

ClaimedAmount <input type="text"/>	is greater than <input type="text"/>	1000 <input type="text"/>
and <input type="text"/>		
Is Random Check <input type="text"/>	is true <input type="text"/>	



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

ClaimedAmount

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

**Car insurance claim (E-5004)** RESOLVED-COMPLETED

*Subject: Disbursement of claim*


Dear customer,

This is to inform you that claim number E-5004 has been resolved. The claimed amount will be disbursed.

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

**Car insurance claim (E-5013)** PENDING-INVESTIGATION

**Assignments**


Task	Assigned to
Please approve or reject this Car insurance claim	 Expert

CREATE DETECT FRAUD CLAIM PROCESS

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

**Car insurance claim (E-5013)** PENDING-APPROVAL

**Assignments**

Task	Assigned to
2m Expert inspection (Claim process)	 Claims Operator

CREATE DETECT FRAUD 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.

What is the outcome type? [?](#)

- Two categories
- Continuous value

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

Prediction name  
Predict Claim Fraud

Outcome  
Custom  
Case completion  
Claims fraud  
Custom  
Continuous value

Subject  
Claim

A placeholder scorecard initially drives the prediction.

#### Claims fraud

Name	Type	Performance	Status
Predict Fraud	Scorecard	---	ACTIVE

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

## Replace model

Compare the models [?](#)

[Upload](#) Machine learning service Model list

Select a PMML, H2O MOJO or Pega OXL file

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

### Evaluate FraudH2O ×

Evaluate the model and provide your feedback. [?](#)

Evaluation

- Approve candidate model and replace current active model  
 Reject candidate model

Reason ★

H2O model replaces placeholder scorecard

The fraud model now drives the prediction.

### Claims fraud

Name	Type	Performance	Status
FraudH2O	Predictive model	---	ACTIVE

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

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

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

## ▼ Outputs

### Results

Result

abnormal

Propensity


0.8375238099694252

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

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

### Predictions

Manage predictions and associated objectives


Prediction	Objective	Data object
<a href="#">Predict Fraud Risk</a> 	Claims fraud	Claim

[+ Add prediction](#)

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

ClaimedAmount	▼	is less than	▼	100
and ▼				
Probability	▼	is less than	▼	10


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

Probability	▼	is greater than	▼	50
or ▼				
Is Random Check	▼	is true	▼	

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

**Car insurance claim (E-5025)** PENDING-APPROVAL


**Assignments**

Task	Assigned to
2m Expert inspection (Claim process)	 Claims Operator

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

**Car insurance claim (E-5026)** PENDING-INVESTIGATION

**Assignments**

Task	Assigned to
Please approve or reject this Car insurance claim	 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.

# Upgrading predictions

## Description

Learn how 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

- Import a model built on an external platform
- Connect to a machine learning service
- Deploy a new model in shadow mode
- Promote the new model to the active model status

# Model operations process

## Introduction

Learn how you can improve the performance of your predictions by using a standardized model 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 model operations process, the active model is replaced with a better one in a production environment by using the shadow mode option.

Pega Customer Decision Hub™ uses predictions to determine the next best action for each customer.

A prediction is driven by an adaptive model, a predictive model, a scorecard model, a field model, or a combination of these models.

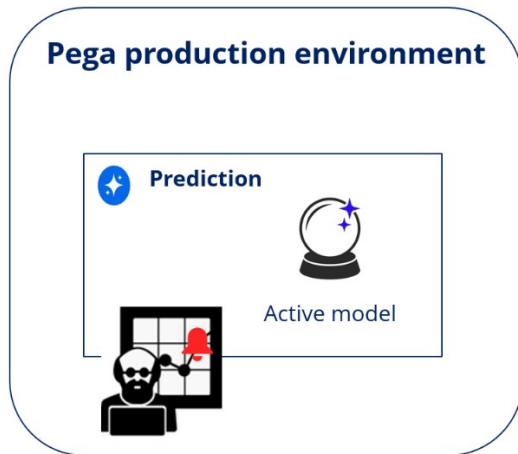


You can replace a model in the production environment at any time through the model operations process.

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.

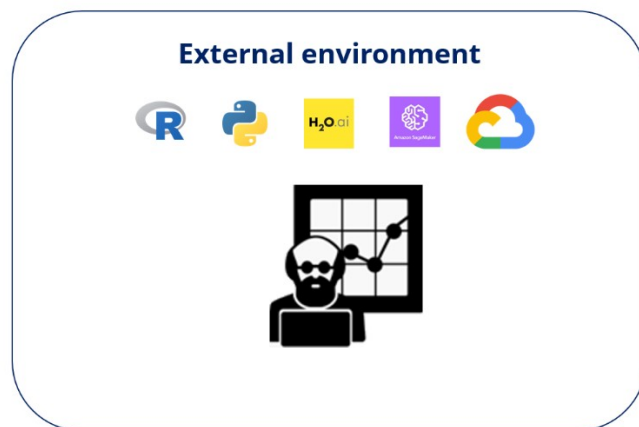
Or you can update a prediction regularly, for example, whenever you develop a new model.



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 H2O format is another option.

You can also connect to the Amazon SageMaker or Google Cloud machine learning services.

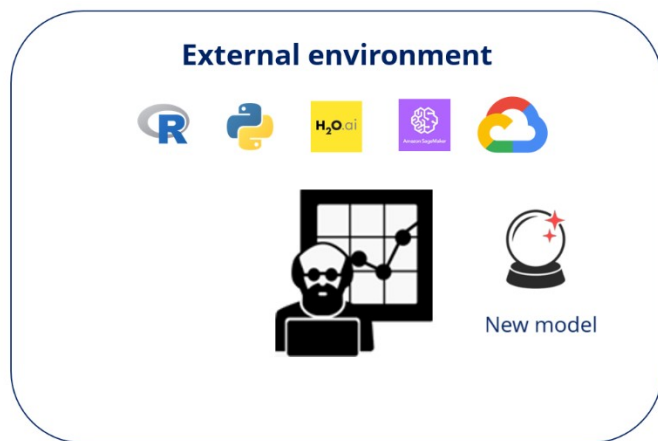
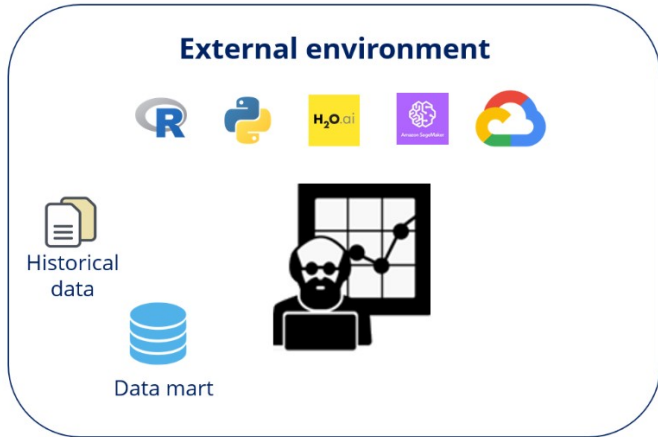


When an application runs in the production environment, you can capture predictor and outcome data for every customer interaction.

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.

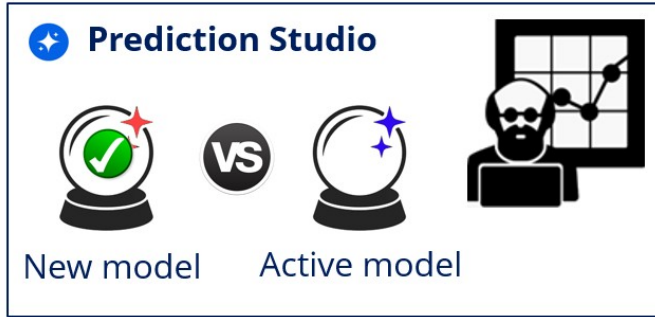




Once the new predictive model is developed, you validate the active model and the candidate model against the same data set to compare their metrics in Prediction Studio.



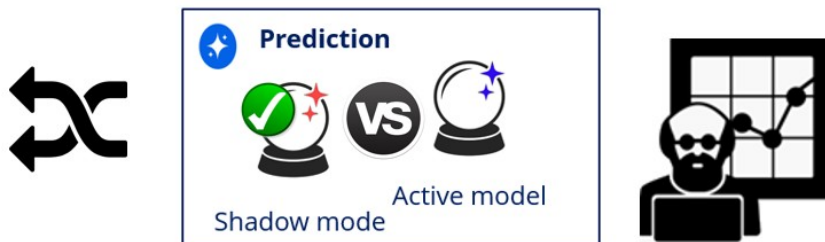
If the candidate model outperforms the active model, approve the model.



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



In the business operations environment, a change request is automatically generated.

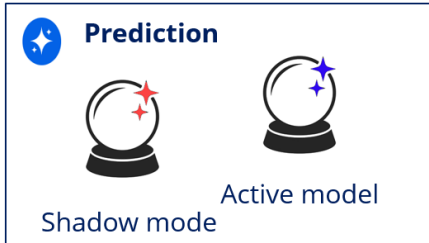


The revision manager approves the change request and packages it in a revision.



The revision is then promoted to the production environment.

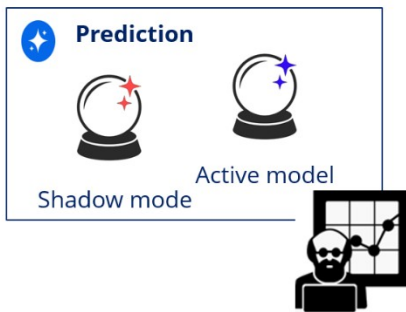
## Pega production environment



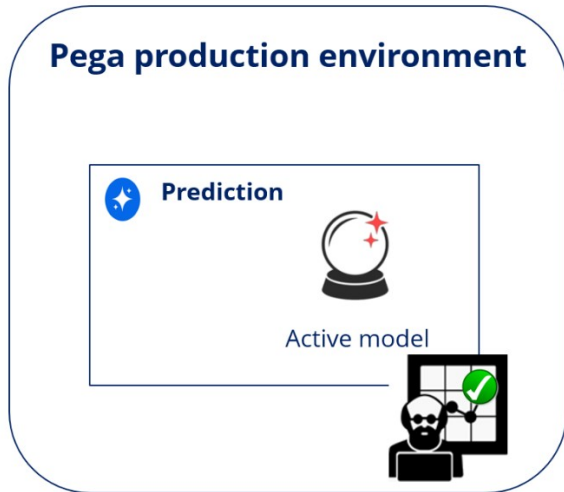
If you deploy the new model in shadow mode, the new model is exposed to the production data but does not drive the prediction yet.

Shadow mode allows you to monitor the model performance in a production environment before deploying it as an active model.

## Pega production environment



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

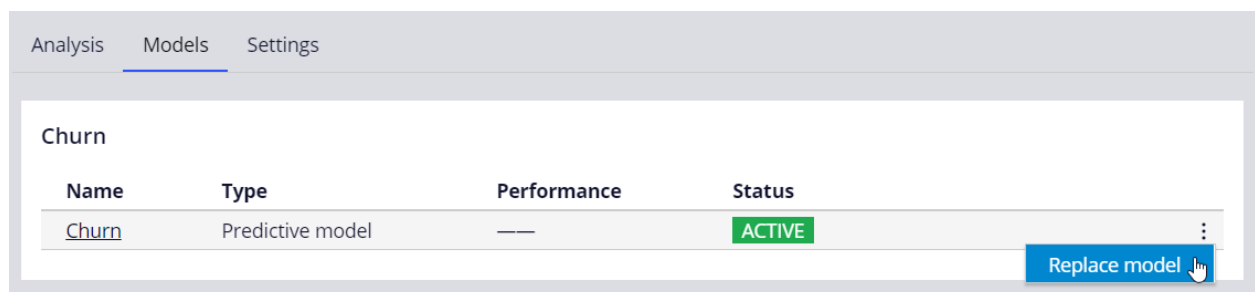
As a data scientist, you can approve changes to the models that drive predictions for deployment to the production environment. You can change models independently or respond to a Prediction Studio notification that a prediction does not generate enough lift.

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.

## Replace model

### What do you want to replace it with?

- Model  
A machine learning model to calculate a score in real-time
- Scorecard  
A simplified method to calculate a score in real-time
- Field  
An existing field in the data model that already contains a precalculated 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.

## Replace model

Compare the models ?

Upload   Machine learning service   Model list

Select a PMML, H2O MOJO or Pega OXL file

Choose File

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.

### Recording historical data

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

Record historical data

churned	Sample percentage 100.00%
loyal	Sample percentage 100.00%

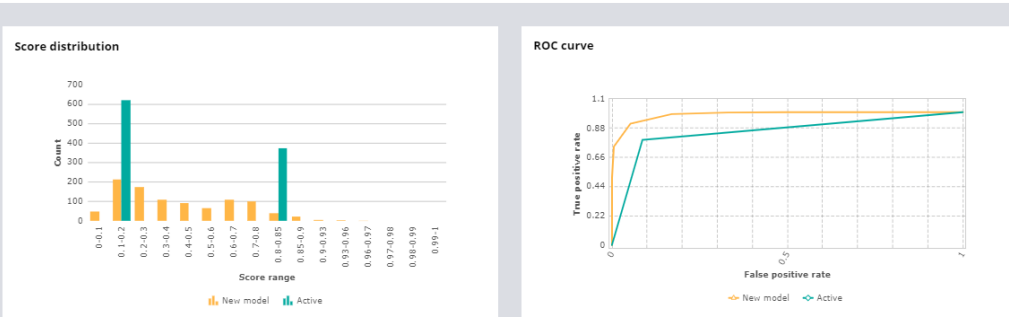
File location  
file:///defaultstore/Predictive/Rule-Decision-PredictiveModel/PegaCRM-Data-Customer/Churn/records\*.json

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.

Model validation  
**READY FOR REVIEW**  
 Configure data set

<b>New model</b> <span style="background-color: green; color: white; padding: 2px;">RECOMMENDED</span> <span>?</span> Model name ChurnH20 Model type H2O.ai <a href="#">View model documentation</a>	VS	<b>Active model</b> Model name Churn Model type Pega
---	----	--



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.

Churn

Name	Type	Performance	Status	
▼ Churn	Predictive model	---	ACTIVE	⋮
<u>ChurnH20</u>	Predictive model	---	SHADOW	⋮

Promote model  
Remove model

Churn

Name	Type	Performance	Status	
<u>ChurnH20</u>	Predictive model	---	ACTIVE	⋮



# 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

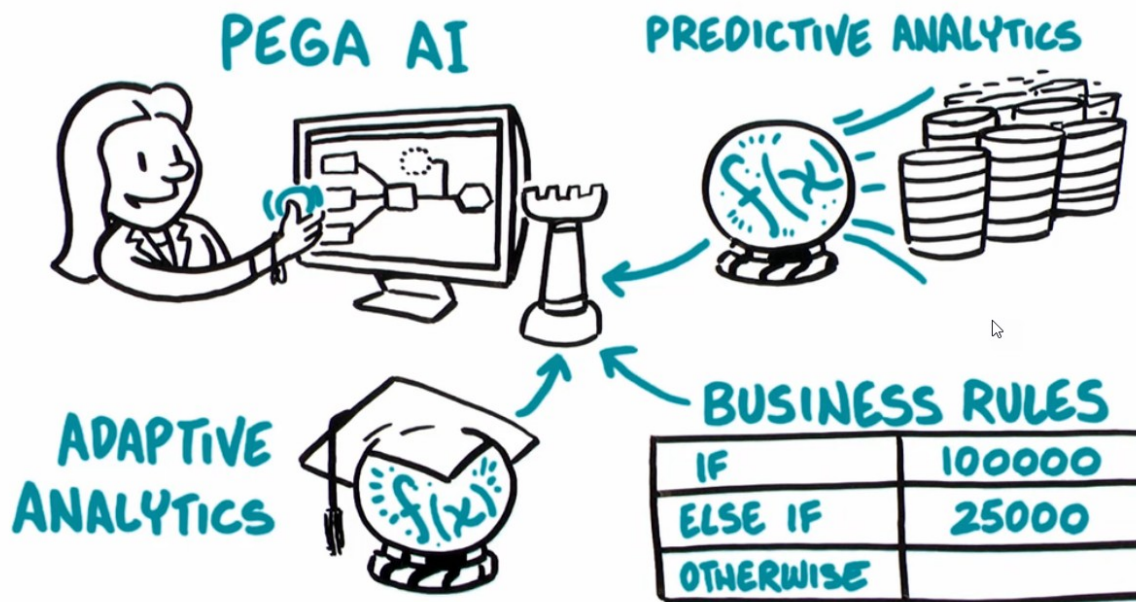
# Adaptive analytics

## 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 the real-time detection of complex fraud patterns or 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

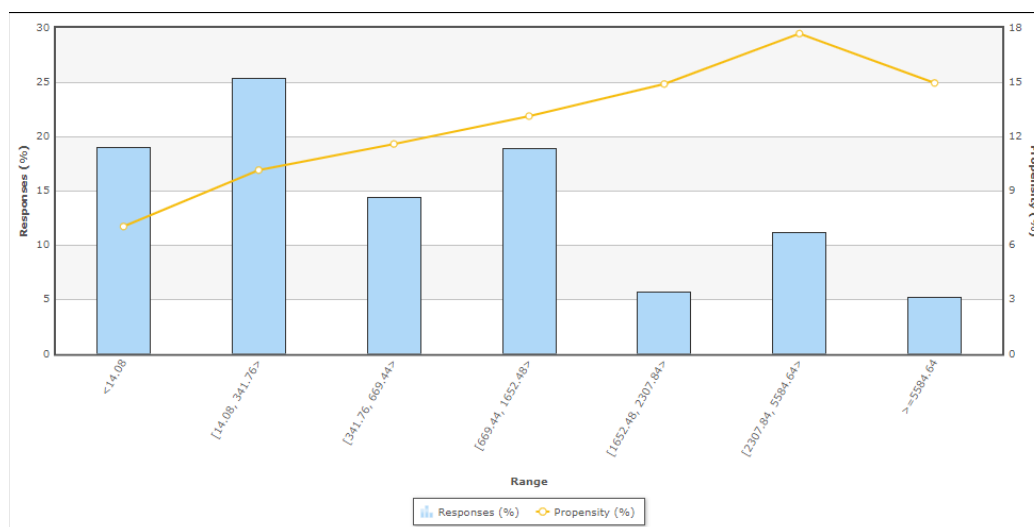
Configuring adaptive models involves selecting potential predictors and setting outcomes that identify positive and negative customer behavior. Unless you are a highly experienced data scientist, it is strongly recommended to leave the advanced settings at their default

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.

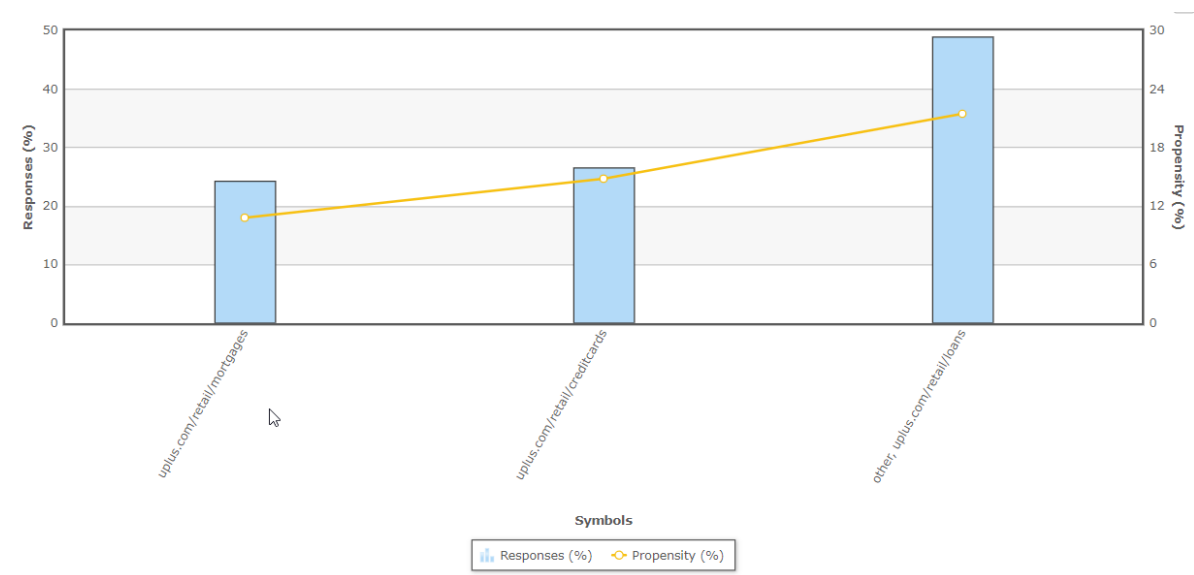
You can use several data types in adaptive analytics, including:

**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

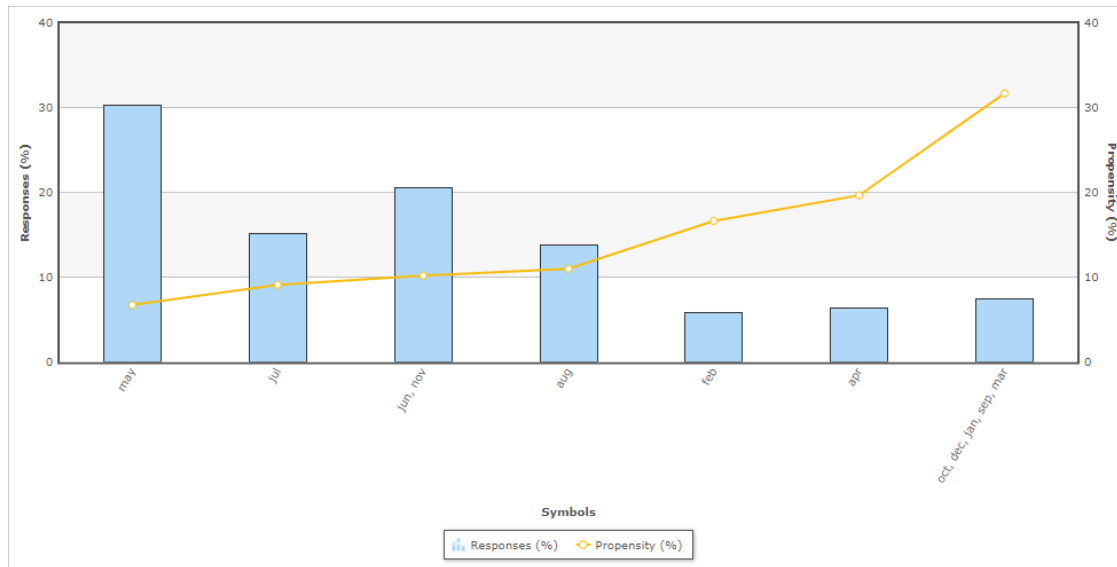
the **PreviousWebpage** predictor in the following example. For predictors with more than 200 distinct values, group the data into fewer categories for better model performance.



**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.

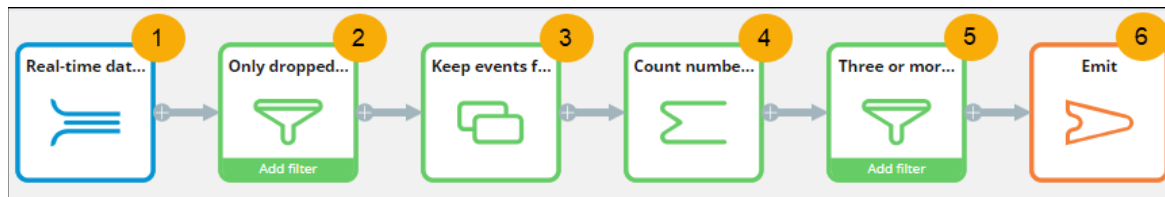
**Codes** - For meaningful numeric fields, feed code fragments to the model as separate predictors. Simple values require only basic transformation. For example, you can shorten postal codes to the first 2 or 3 characters which, in most countries, denote geographical location.

**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.

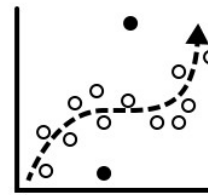
In summary, to achieve the best results, use predictors that provide data from many different sources, including:

**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.

Positive outcome ⓘ

Add outcome

Clicked

Accepted

Negative outcome ⓘ

Add outcome

NoResponse

Rejected



# 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.

Claim Customer CustomerID
<input type="text" value="C-1"/>
Claim AccidentDate
<input type="text" value="08/08/2021"/>
Claim AccidentLocation
<input type="text" value="High Street, Boston"/>
<input type="checkbox"/> Claim Casualties
<input type="checkbox"/> Claim Fatalities

<b>Assigned to</b>	
 Claims Operator	<a href="#">Begin</a>

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

<b>Task</b>
<b>2m</b> Expert inspection (Claim process)

**Assigned to**

E Expert Begin

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

Claim Customer CustomerID  
C-1

Claim AccidentDate  
08/08/2021

Claim AccidentLocation  
High Street, Boston

Claim Casualties

Claim Fatalities



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.

### Create a prediction

#### Where will you be using the prediction?

- Customer Decision Hub  
Optimize the engagement with your customers
- Case management  
Use predictions to improve the automation in cases
- Text analytics  
Analyze the text that comes through your channels

Process AI offers a wizard to create 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
<a href="#">Case completion</a>	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.

	<input type="checkbox"/> Name	Data type
▼ Current page (EscalateApp)	<input checked="" type="checkbox"/> ClaimFrequency	Integer
▶ Page Claim	<input checked="" type="checkbox"/> ClaimRatio	Decimal
▶ Page Customer	<input type="checkbox"/> EndDate	DateTime
▶ <b>Page Policy.</b>	<input checked="" type="checkbox"/> EntitledAmount	Decimal
	<input type="checkbox"/> PolicyID	Text
	<input type="checkbox"/> StartDate	DateTime
	<input checked="" type="checkbox"/> 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 

(Casualties 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
<a href="#">Predict regular case completion</a> 	Case completion
<a href="#">+ Add prediction</a>	

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

▼

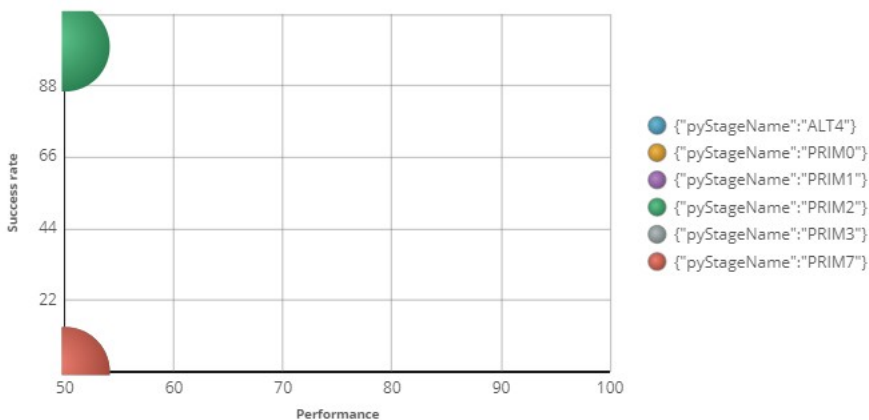
20

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



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

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



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

This demo has concluded. What did it show you?

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

# 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 module showcases 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

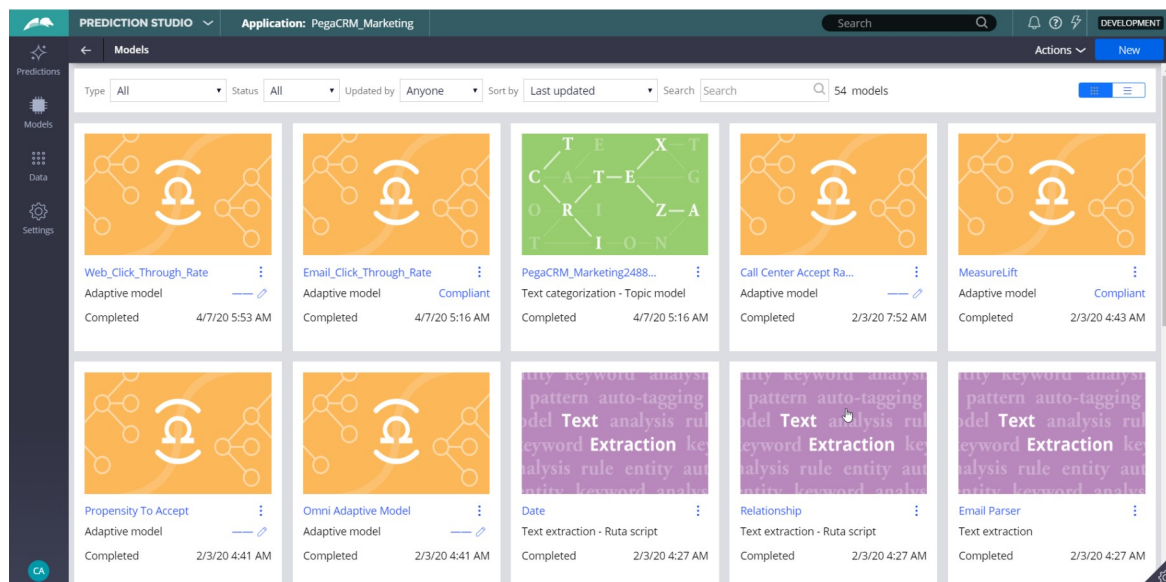
## Introduction

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

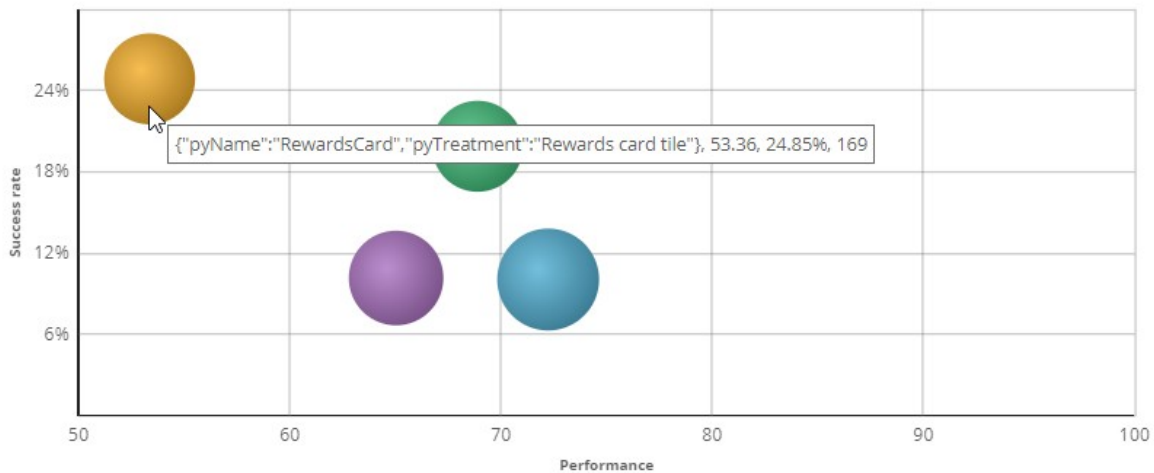
This demo will show how to inspect the health of your adaptive models and their predictors. This is a regular data scientist task.

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.



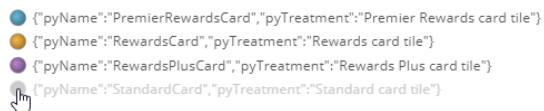
Each bubble represents the model for a specific action.

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.

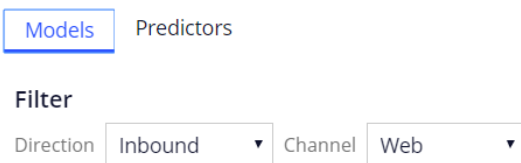
The information you see here is extracted from the Adaptive Data Mart, which is a reporting view of the Adaptive Decision Manager (ADM) server.

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

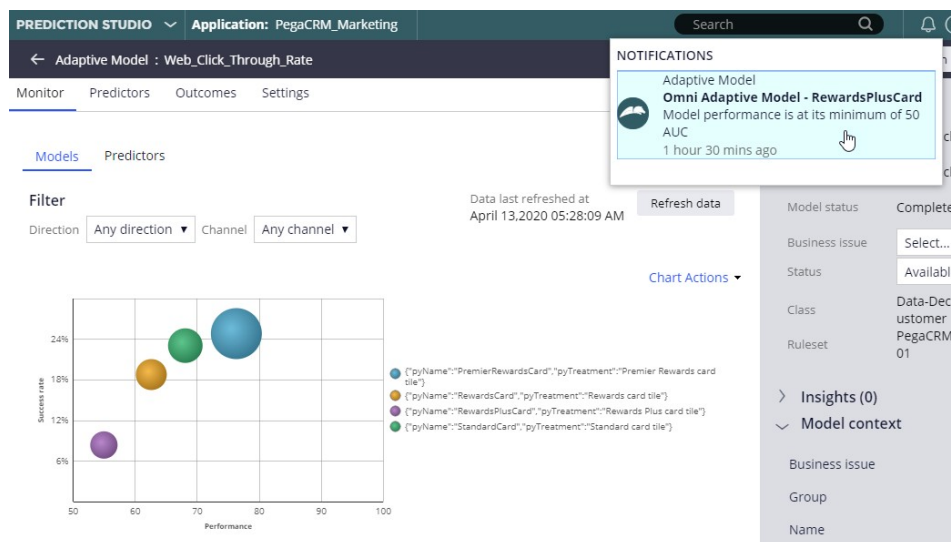
This means that the data as reported may not be the same as the data you see when you view it in real-time. You can refresh the view to synchronize the data.

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.px.LastOutcomeTime.DaysSince	4	0
Customer.HasMortgage	3	1

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

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

The default value for this threshold is 52 percent.

### Predictor selection

Activate predictors with a performance above

 AUC

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

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

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

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

Predictors   Score distribution   Trend

---

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

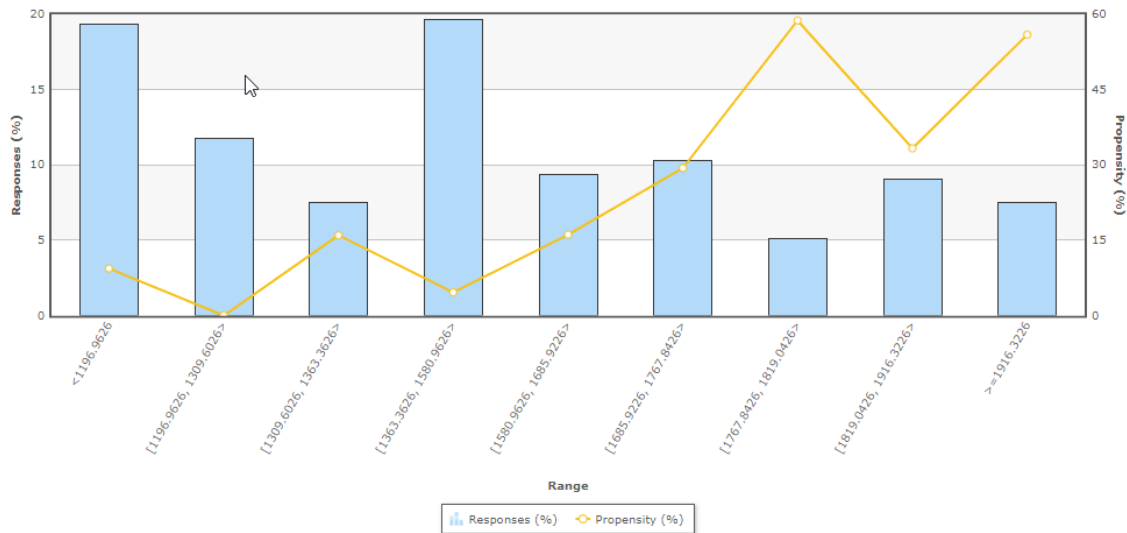
Predictors	Status	Type	Performance (AUC)	Range/Symbols(#)	Bins(#)
<a href="#">Customer.AverageSpent</a>	Active	Numeric	80.86	[1001.33; 1997.33]	9
<a href="#">Customer.Age</a>	Active	Numeric	75.54	[19.0; 80.0]	9
<a href="#">Customer.InteractionContext.PreviousWebpage</a>	Active	Symbolic	74.22	4.00	4
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

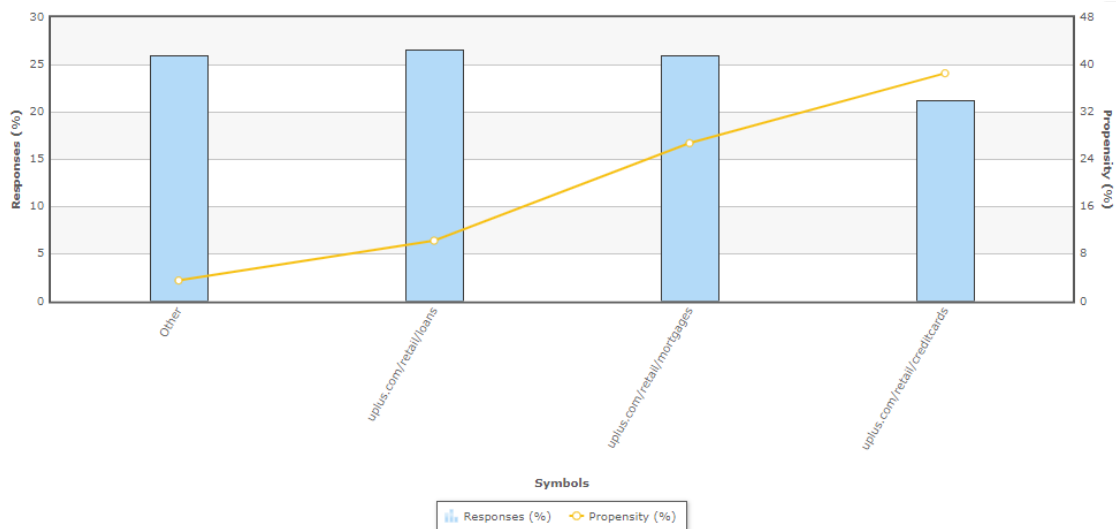
The system split the AverageSpent predictor into 9 bins. Each bin has its own offer propensity.



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

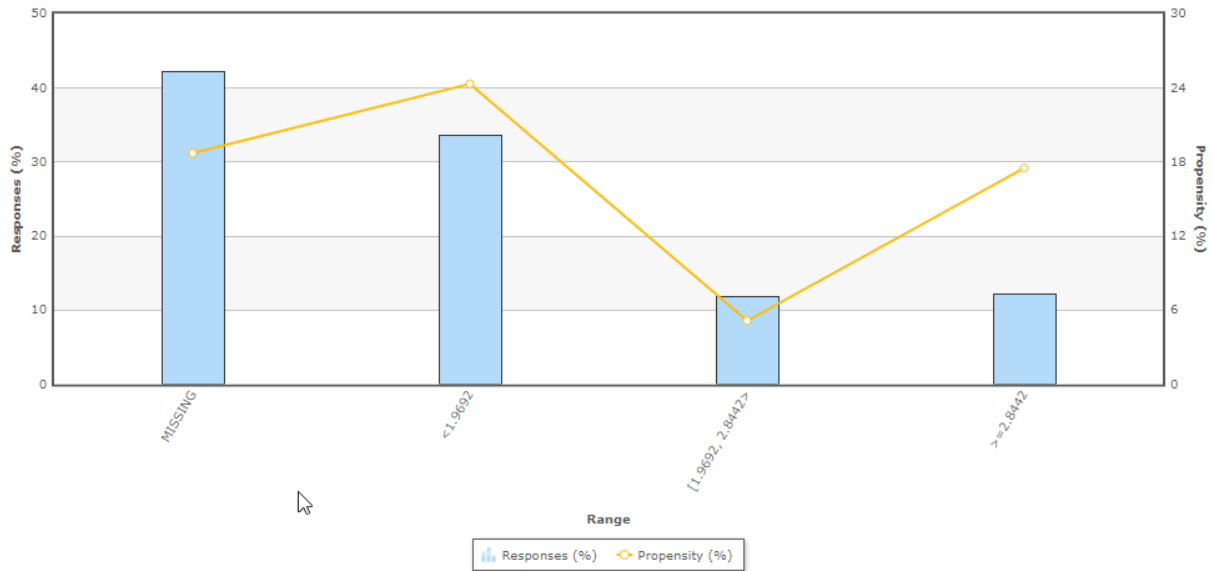
Now, let's examine the symbolic PreviousWebpage predictor.

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.

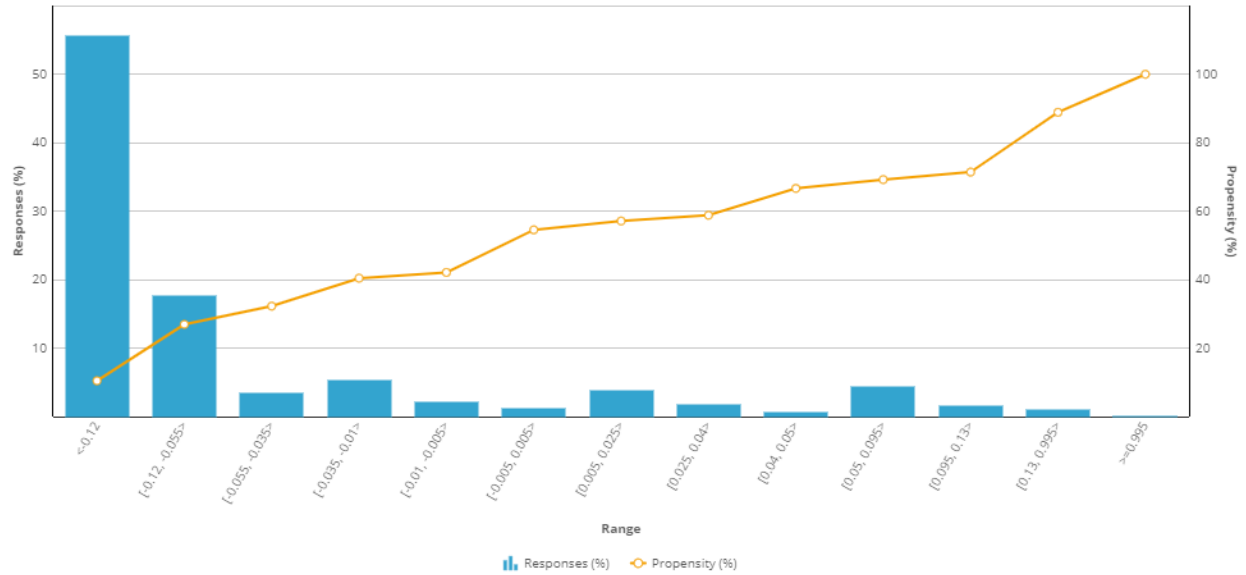
In this example, the adaptive system established that the number of days since the offer was accepted is a well-performing predictor.



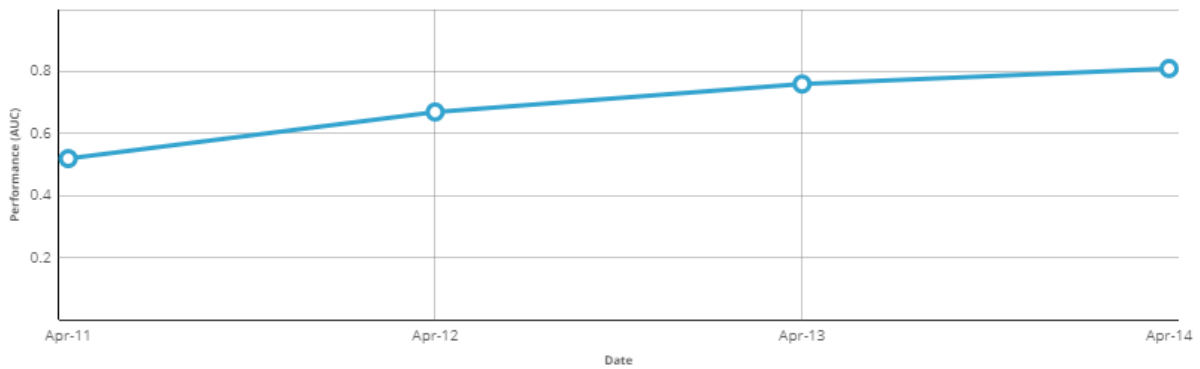
The system grouped three predictors that are correlated. It then marked two of them as inactive. Inactive predictors are not used in the propensity calculation.

▼ 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

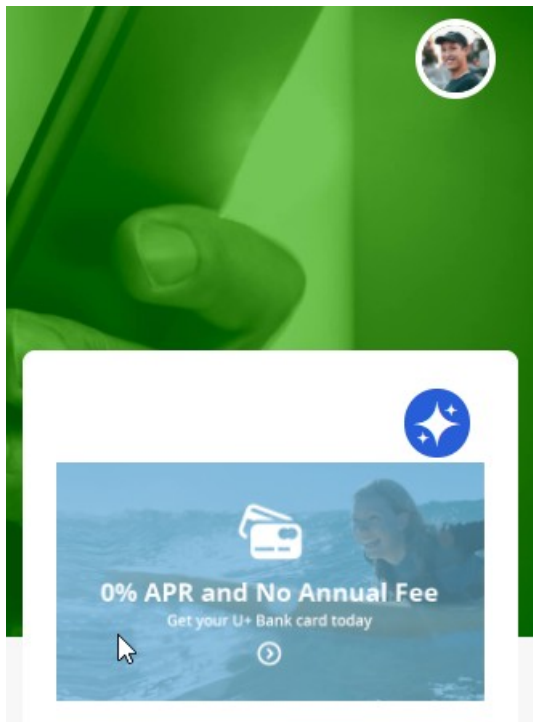
## Introduction

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 as negative behavior.

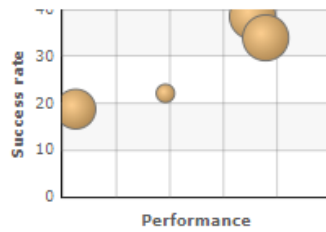
When the customer clicks on the banner, the model regards this as positive behavior.

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.



[Web\\_Click\\_Through\\_Rate](#) ⋮

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.

### Recording historical data

Save historical data in a repository to use for offline analysis.

You can find an overview of the historical data in

[Historical data overview](#).

Record historical data

Clicked      Sample percentage  
100.00%

NoResponse, Impression      Sample percentage  
1.00%

The sample percentages determine the likelihood that a customer response 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.

### Select repository type



Artifactory



S3



File system



Azure

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

### New data set



Name \*

Type \*

Apply to \*

Development branch

Add to ruleset

Ruleset version

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

**File** Mapping

---


**Data source**

Files on repository  
 Embedded file

---

**Connection**


Repository configuration\*


defaultstore 

---

**File configuration**

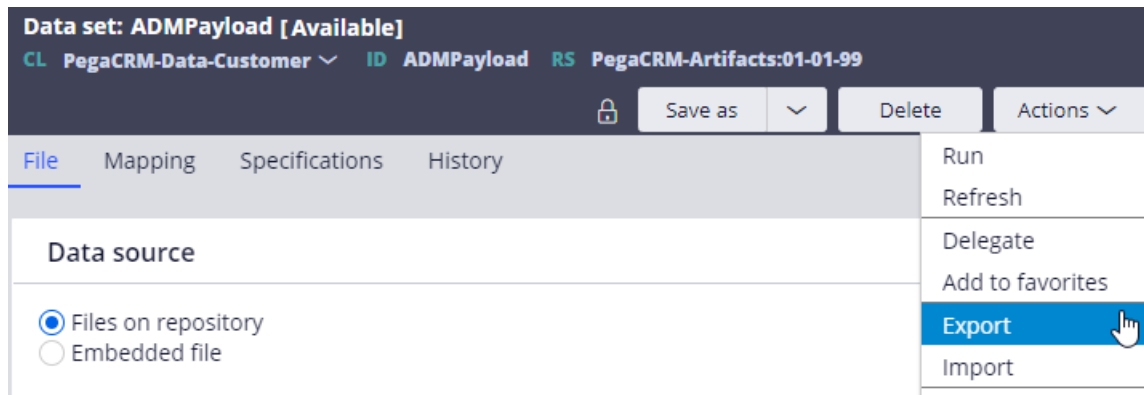
Use a file path  
 Use a manifest file

File path\* 

ADM/Rule-Decision-AdaptiveModel/Data-Decision-Request  Preview file

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

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



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.

The screenshot shows a software interface with a 'Fields (7)' list on the left and a 'Details' panel on the right. The 'Fields (7)' list includes: Customer.Age, Customer.AverageBalance, Customer.AverageSpent, Customer.BranchCode, Customer.ChurnScore, and Customer.City. The 'Details' panel shows a JSON object for 'Web\_Click\_Through\_Rate'. The 'Customer.Age' field is highlighted in red in both the list and the JSON data.

```

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

```

To use the JSON file for further analysis, import the file into a third-party analytics tool.

Keep in mind that when many customers visit the website, the file size becomes very large in a short time. To limit the storage space needed, you can lower the sample percentages.

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

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