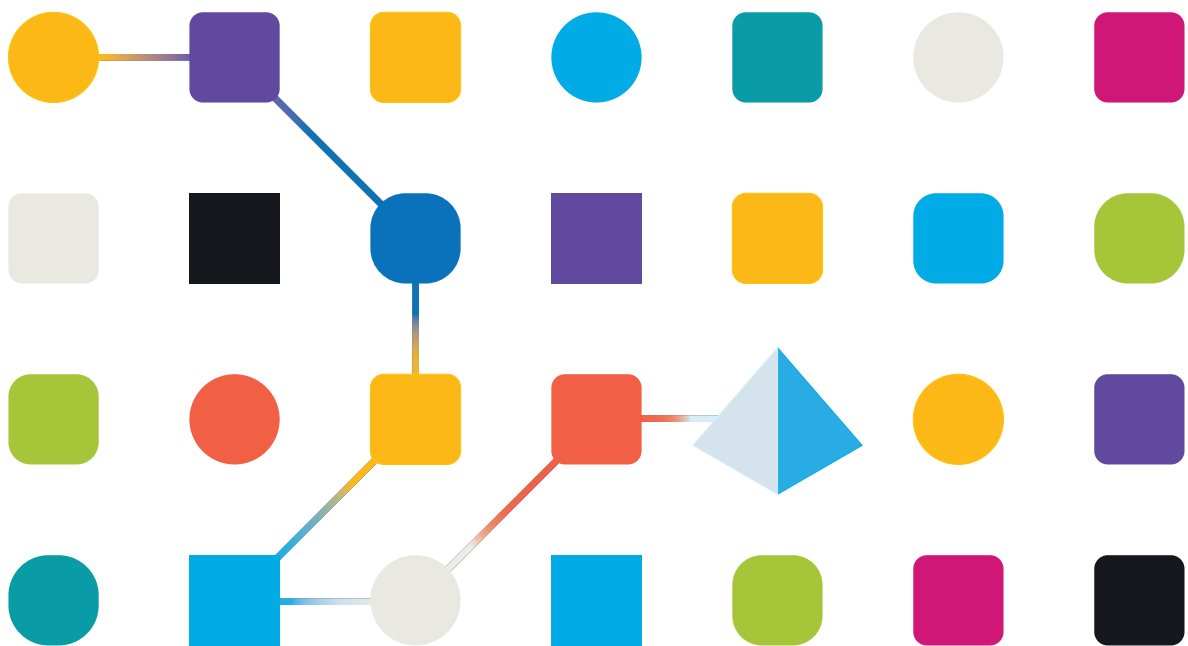




Decision 4.5

User Guide

Document Revision: 2.0



Trademarks and Copyright

The information contained in this document is the proprietary and confidential information of Blue Prism Limited and should not be disclosed to a third-party without the written consent of an authorized Blue Prism representative. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying without the written permission of Blue Prism Limited.

© Blue Prism Limited, 2001 – 2022

“Blue Prism”, the “Blue Prism” logo and Prism device are either trademarks or registered trademarks of Blue Prism Limited and its affiliates. All Rights Reserved.

All trademarks are hereby acknowledged and are used to the benefit of their respective owners. Blue Prism is not responsible for the content of external websites referenced by this document.

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom.
Registered in England: Reg. No. 4260035. Tel: +44 370 879 3000. Web: www.blueprism.com

Contents

Blue Prism Decision	4
Intended audience	4
Dependencies	4
Compatibility	4
Disclaimer	4
Getting started with the Decision plugin	5
Log into Hub	5
Open the Decision plugin	6
Library overview	7
Creating a new model	8
Defining your model	8
Training your model	13
Calibrating your model	16
Using a model in an automation	19
Use your Decision model in a process	19
Include a human reviewer	22
Maintaining the models	26
Editing a model	26
Deleting a model	26
Viewing a model's usage	27
Glossary of terms	28

Blue Prism Decision

Blue Prism® Decision provides Blue Prism users with the ability to train and deploy intelligent machine-learning decision-making models within their Digital Workforce, without requiring any data science expertise. Blue Prism Decision's simple and intuitive user interface allows users to:

- **Define a model** – Add a title, description, decision type, and decision variables.
- **Train the model** – Use active learning based on defined input variables specified by the model service, without needing any data science knowledge to produce statistically accurate machine-learning models.
- **Calibrate the model** – Review model inputs and outcome predictions, and overwrite model decisions if required.

This guide outlines the functionality and usage of the Decision plugin available in Blue Prism® Hub.

Intended audience

This guide is intended to be used by anyone who wants to use Decision to create a model that can be used in an automation and any users who create processes in Blue Prism that will use these models.

Dependencies

The Decision plugin requires access to the **Blue Prism Decision container** – the Decision plugin's Machine Learning API is packaged in the Blue Prism Decision container. For information on installing the container, see [Install Blue Prism Decision](#).

Compatibility

Blue Prism Decision is compatible with Blue Prism 6.4 or later.

Disclaimer

It is your organization's responsibility to implement Decision in accordance with laws applicable to your organization. For example:

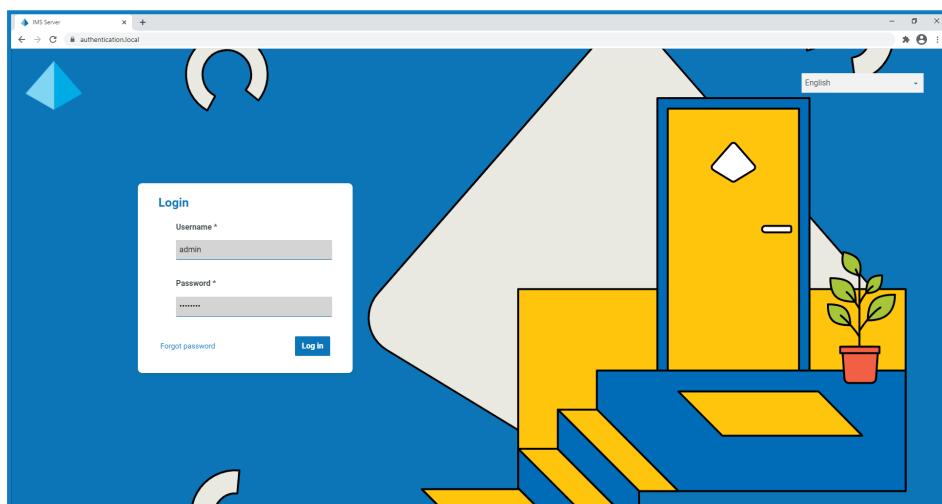
1. Your organization may be subject to laws that prohibit or restrict it from making solely automated decisions that have a legal or similarly significant effect on an individual (such as a decision about access to credit or shortlisting for a job). You can build in human involvement in reviewing decisions by building a process with human interaction by adding a human-in-the-loop after the decision is made, in the exact same way they would do now in a process using tools like Blue Prism® Interact, to review the decision if necessary.
2. Certain data privacy laws require that an organization using a statistical model to make a decision/prediction about people ensure that the model is sufficiently statistically accurate and avoids discrimination in order for the processing of personal information to be fair. The accuracy of the model is continually updated as the user continues to train the model, and it is solely at the discretion of the user to continue to train a model (by supplying enough samples) to the sufficient accuracy they wish to get to, before utilizing the decision capabilities in a production process.
3. To comply with the principle of transparency, certain data privacy laws require that an organization informs people about how it processes personal data in a machine learning system, which includes being able to explain the basis for any decisions. Decision is built on the principles of simplicity and auditability. Once the model is utilized in a process, you can view an audit log that records every input, output, confidence score, and output decision made. This information is available in the [prediction list](#).

Getting started with the Decision plugin

Your system administrator will create an account for you to use to access Blue Prism® Hub and grant you permissions to use the Blue Prism Decision plugin. When your account has been created, you will receive an email with details of your account and access information. You should follow the instructions in the email to log in for the first time.

Log into Hub

1. Launch a browser and go to the Authentication Server website. Your system administrator will provide you with the website address. Our example shows <https://authentication.local>



2. Log in using your credentials.

The application selection screen displays. If you only have access to Hub, you will only see the Hub tile. If you also have access to the Interact web-based application, you will see a tile for Hub and a tile for Interact, as shown below:

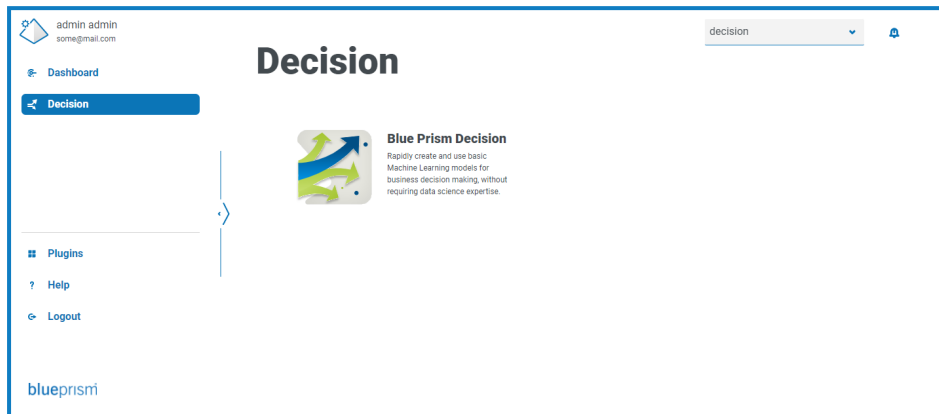


3. Click **Hub** to launch the Hub website.

The Hub Dashboard displays.

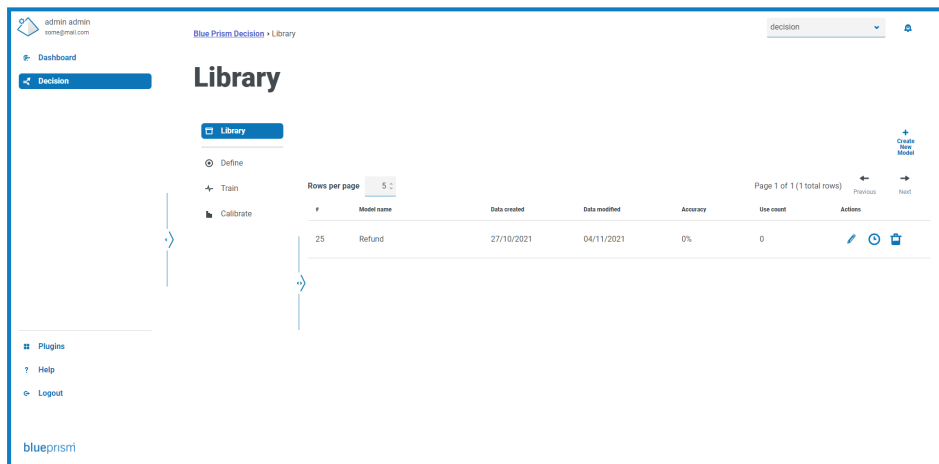
Open the Decision plugin

1. Click **Decision** on the sidebar menu to open the Decision plugin.



2. Click **Blue Prism Decision**.

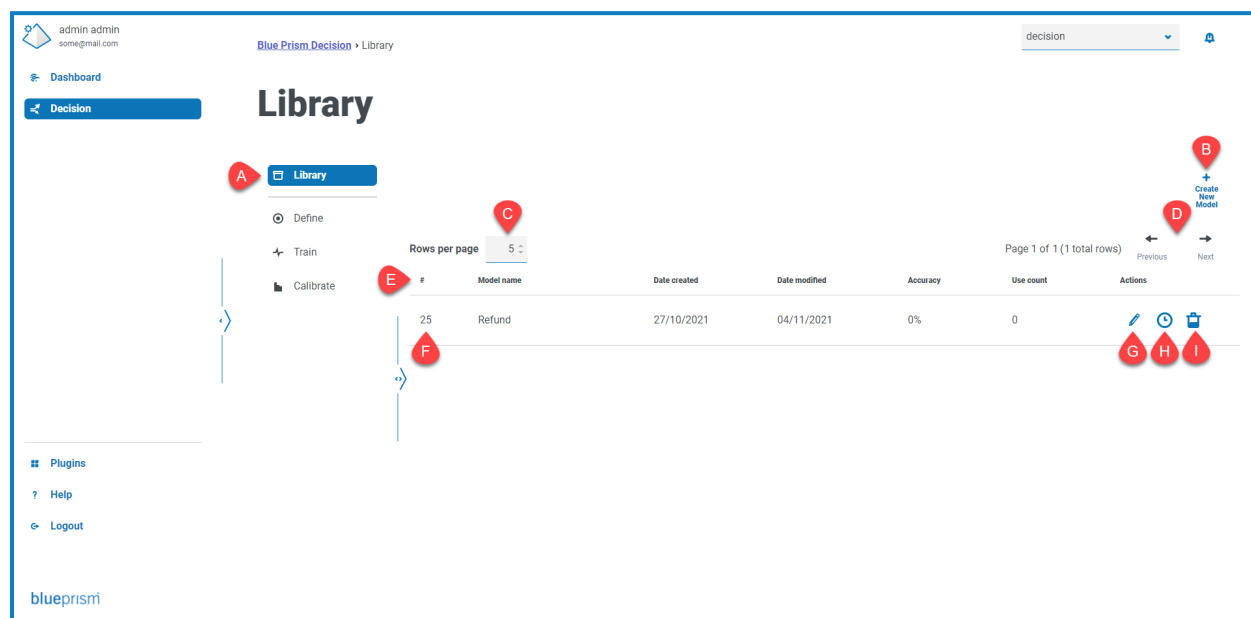
The Decision plugin displays the Library page.



The example above shows a single model that is in the process of being created and will be used as our example throughout this guide.

Library overview

The Library page displays all the models that have been created in Decision by you and the other Decision users in your system. From here, you can create new models and maintain any of the existing models.



The Library page provides you with the following information and functions:

- A. **Side menu** – Indicates which page in the Decision plugin you are viewing. This example shows Library selected. The other menu options are only available when you are creating or editing a model.
- B. **Create New Model** – Add a [new model](#).
- C. **Rows per page** – Enter a number, or use the up and down arrows, to change the number of rows seen on a page.
- D. **Previous and Next** – Click **Previous** or **Next** to move through the pages.
- E. **Column headings** – Click a column heading to change the sort order. For example, click **#** to sort the model identifier by ascending number, click **#** again to sort by descending.
- F. **Model identifier** – The **#** column displays the model identifier. This unique number is used when calling the model from a Blue Prism automation.
- G. **Edit** – Click to [edit the associated model](#).
- H. **Predictions list** – Click to view a [summary of the model's usage](#) by automations.
- I. **Delete** – Click to [delete the associated model](#).

Creating a new model

Blue Prism® Decision is designed to enable you to automate decision making by utilizing machine learning model-based training. Decision is highly flexible, enabling you to create scenarios to meet your organization's requirements.

When you create a decision model, you need to:

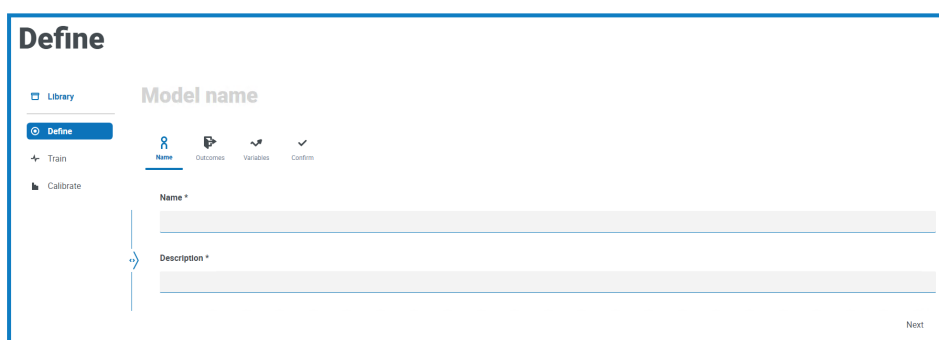
1. [Define your model](#) – Add a title, description, decision type, and decision variables.
2. [Train your model](#) – Add rules and samples that will be used to form a decision.
3. [Calibrate your model](#) – Review your model decision outcomes and make any corrections.

The information below uses the example of a simple decision model called Refund. This model will determine whether or not a refund should be given based on the entered criteria.

Defining your model

1. On the Blue Prism Decision [Library page](#), click **Create New Model**.

The Define page displays with the **Name** icon active.



2. On the **Name** tab:
 - a. Enter a name and description for your model.
As you type in the name, the **Model name** above the icons changes to reflect what you have entered.
 - b. Click **Next**.

The **Outcomes** tab displays.

3. On the **Outcomes** tab:
 - a. In **Question**, enter the main question that this decision model is intended to answer. For example, *Should I give a refund?*
 - b. Select whether the answer should be **Numerical** or **Categorical**. For more information, see [What are Numerical and Categorical answers? on page 11](#).

- c. Enter the required values based on the type selected. For example:

Refund

Name

Outcomes

Variables

Confirm

Question *

Should I give a refund?

Numerical

Categorical

Value

Add new

+ Yes/No

+ Maybe

+ Low/Medium/High

+ Small/Medium/Large

+ Reject

☐ Is ordinal

Edit

Delete

Value	Selected (0)
Yes	<input type="checkbox"/>
No	<input type="checkbox"/>

Back

Next

- d. Click **Next**.

The **Variables** tab displays.

4. On the **Variables** tab:

- Click **Add** to add a new variable to the model.
- In **Variable name**, enter information that is relevant to the model you are creating. You may find it useful to structure this as a question, for example, *What is the value of the item?* or *Have they returned the item?*
- Select whether the variable is **Numerical** or **Categorical**. For more information, see [What are Numerical and Categorical answers? on page 11](#).

- d. Enter the required values based on the type selected. For example:

The screenshot shows a form titled "Refund" with four tabs: Name, Outcomes, Variables, and Confirm. The "Variables" tab is active. The form contains the following fields:

- Variable name ***: A text input field containing "What is the value of the item?".
- Variable type**: Two radio buttons, "Numerical" (selected) and "Categorical".
- Lower value ***: A numeric input field containing "10".
- Upper value ***: A numeric input field containing "200".
- Increment value ***: A numeric input field containing "10".

At the bottom right of the form are "Save" and "Cancel" buttons.

- e. Click **Save**.
- f. Repeat this process until you have defined all the variables that need to be considered in order to make a decision.
- g. Click **Next**.

The **Confirm** tab displays showing a summary of all the information you have entered.

5. On the Confirm tab, click **Next** to save the model.

The **Rules** tab of the Train page displays. You are now ready to [train your model](#).


What are Numerical and Categorical answers?

When you are defining your model, you can determine whether the answer to your model question and the individual variables is:

- **Numerical** – Defines that the answer has a numerical value. You must enter a range along with an increment. An example of fields that display when **Numerical** is selected is shown below:

To define your answer, in:

- **Lower value** – Enter the minimum number that can be accepted.
- **Upper value** – Enter the maximum number that can be accepted.
- **Increment value** – Enter the expected increment.

 When entering your range and increment, you should consider the number of possible data points this will generate. The greater the number of data points, the longer Decision will take to process the information when training the model. For example, if you have a lower value of 0 and an upper value of 50,000 with an increment of 1, this will generate 50,000 data points. If you change the increment to 2, this will generate 25,000 data points and reduce the processing time.

For variables, Decision has a limit of approximately 70,000 data points (the total sum over all input variables). This many data points can take up to 10 minutes to return a test question when training the model.

For outcomes, Decision has a limit of approximately 100,000,000 data points.

If you exceed these limits, your model will time out and cause an error.

You must use a range and increment that will minimize the processing time, whilst providing the level of information that you need.

- **Categorical** – Defines that the answer has a text-based categorical answer. You must enter a selection of valid answers to the question or variable. Decision has a number of predefined options that you can select, or you can defined your own answers. An example of the fields and options that display when **Categorical** is selected is shown below:

To define your answer:

- a. You can do one of the following:
 - Select one of the predefined options below the Value field.
 - Enter your own entry in **Value** and click **Add new**.
 - Combine these methods – select a predefined option and add some additional values.
- b. If required, select **Is ordinal** – the answer values have a priority based on the order you have entered them in.

As you add the values, they display in a summary below the fields, for example:

Value

Add new

+ Yes/No

+ Maybe

+ Low/Medium/High

+ Small/Medium/Large

+ Reject

☒ Is ordinal


Value	Actions
Small	<div><div></div><div></div></div>
Medium	<div><div></div><div></div></div>
Large	<div><div></div><div></div></div>
X-Large	<div><div></div><div></div></div>

You can edit or delete any of these values.

Training your model

When you have defined your model, you are ready to train your model. When you first open the Train page, the Rules tab displays with the model name and rule question visible:

1. On the **Rules** tab of the Train page, you can define hard-coded rules. Rules are defined as conditions on input variables and override all other outcomes. Multiple rules can be created and they will be applied in the order they are created, with the oldest first.

 Rules are not required for most models. Training the model and creating samples should be enough to achieve the desired outcome. Only add a rule if you have a definitive outcome that will always be true based on a specific condition.

If you require any rules, follow the steps below. Otherwise, click **Next** and move to the **Initialise** tab in [step 2](#):

- a. In **Variable**, select the variable you want to define an answer for.
- b. Select the appropriate validator. The options available depend on whether the values are categorical or numerical. They are:
 - = Equal to
 - != Is not equal to
 - < Less than
 - <= Less than or equal to
 - > More than
 - >= More than or equal to
- c. In **Value**, select the appropriate value based on the variable and the validator.


- d. In **Answer**, select the appropriate answer for the model question based on the variable criteria you have defined, for example:

Variable	Value	Question	Answer
IF What is the value of the item?	<= 50	THEN Should I give a refund?	= Yes

+ Add rule

- e. Click **Add rule**.

The rule appears in the summary at the bottom of the Rules tab.


 Rules appear in the order newest to oldest – they are applied in the order oldest to newest.

- f. Repeat the process building up any other rules that are needed.
- g. Click **Next**.

The **Initialise** tab displays.

2. On the **Initialise** tab, if you created a model that has a:

- Categorical outcome – you are prompted to create a sample for each outcome. This is required so that the model is aware of which values can be returned. If a possible outcome value does not have at least one sample, then the model will be unable to predict this value.
- Numerical outcome – you do not need to initialize your model, click **Next** and move to the **Teach** tab in [step 3](#).

 You can only perform the **Initialise** steps once for each outcome. When you click **Create Sample**, the information is committed and cannot be edited for the chosen outcome. Ensure you set the values correctly.

For a categorical model:

- a. Adjust the values of the variables until they are appropriate for the shown outcome, for example:

Refund

Rules
Initialise
Teach

Before teaching Blue Prism Decision you must first create a sample for each possible outcome. For example: If the outcome can be Small, Medium, or Large you would need to adjust the variable values that would lead to each of those 3 outcomes.

Should I give a refund? = Yes

Adjust the variable values to give an outcome of Yes

Variable	Value
What is the value of the item?	100 ← →
Have they returned the item?	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
Can it be resold?	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
How big is the item?	<input checked="" type="checkbox"/> Small <input type="checkbox"/> Medium <input type="checkbox"/> Large

[Create Sample](#)

[Back](#)
[Next](#)

- b. Click **Create Sample**.

The values are stored and a new sample outcome displays.

- c. Repeat the process until new sample outcomes are no longer displayed.
- d. Click **Next**.

The **Teach** tab displays.

3. On the **Teach** tab, you are presented with a generated scenario that you need to provide a response to, for example:

Refund

Should I give a refund?

Variable	Value
What is the value of the item?	80
Have they returned the item?	Yes <input checked="" type="checkbox"/> No
Can it be resold?	Yes <input checked="" type="checkbox"/> No
How big is the item?	Small <input checked="" type="checkbox"/> Medium Large

Answer: Yes No

Create Sample

Back Next

- a. Click the appropriate answer.
- b. Click **Create Sample**.

The values are stored and a new sample displays.

- c. Repeat the process until you have created enough samples to cover different scenarios. The more samples you create, the more accurate your model will be.

Consider the types of variables your model contains when deciding on the number of samples you want to use to train your model. For models with variables which have:

- Multiple values, you could start with around 15 samples.
- Values with two options (binary values), such as Yes/No or True/False, you may only need a handful of samples as the number of value combinations will be less. For example, if you had three variables all with binary values, you can only have eight possible combinations.

You can create more samples later if needed.

If required, you can change the generated scenario before you answer it. To do this:

- For Categorical variables, click the appropriate value.
- For Numerical variables, either use the drop-down list or use the slider to change the number shown.

You should not type a number directly into Numerical variable fields. If you do, this number will not be recognized and the previous value will be applied.

- d. Click **Next**.

The **Ask** tab of the Calibrate page displays. You are now ready to [calibrate your model](#).

Calibrating your model

When you have trained your model, you are ready to calibrate your model. You can review the model's predicted outcomes for your chosen inputs and correct as needed. When you first open the Calibrate page, the Ask tab displays with the model name, the model variables, the model's prediction, and the basis that this has been made on – a rule, matching sample or a decision based on a match to a nearest sample from a selection of similar samples. For example:

Calibrate

Library Define Train **Calibrate**

Refund

Ask Samples Weighting

Should I give a refund?

Variable	Value	Weight
What is the value of the item?	10	
Have they returned the item?	Yes	No
Can it be resold?	Yes	No
How big is the item?	Small	Medium Large

Prediction Confidence Your Correction

Yes 89% Yes Save correction

Neighbours/points used

What is the value of the item?	Have they returned the item?	Can it be resold?	How big is the item?	Should I give a refund?	Confidence
100	Yes	Yes	Medium	Yes	89%

Next

The format of Ask tab is similar to the Teach tab. On the **Teach** tab, Decision was asking you for the outcome based on the variable values it provided. On the **Ask** tab, Decision provides you with the outcome based on the variable values you set.

- On the **Ask** tab of the Calibrate page, you must review and correct the predicted outcomes from the model:
 - Review the outcome under **Prediction** based on the variable values that display.

The **Confidence** of the model also displays – the higher the percentage, the more likely it is that the model provides you with the answer you expect.

- b. If the **Prediction** is based on a match to a similar sample (shown under **Neighbours/points used**), and you disagree with the outcome, you can enter the required value in **Your Correction** and click **Save correction**.



You cannot correct a prediction that is based on a matching sample or a rule. The **Neighbours/points used** section is replaced with **Sample used** when a matching sample is found, or **Rule used** when a rule is applied.

If you disagree with the prediction outcome that is based on a rule or matching sample, you will need to re-train your model. You will either need to correct:

- A specific [rule](#).
- The order of the [rules](#) – they are applied oldest to newest.
- The samples – this may require you to delete the samples that are incorrect and create new samples using the [Teach tab](#).

- c. Change the values of the variables and repeat the steps above. To do this:
- For Categorical variables, click the appropriate value.
 - For Numerical variables, either use the drop-down list or use the slider to change the number shown.



You should not type a number directly into Numerical variable fields. If you do, this number will not be recognized and the previous value will be applied.

- d. When you feel that you have reviewed enough combinations, click **Next**.

The **Samples** tab displays.

2. On the **Samples** tab, you can see a summary of the samples that were created when you trained your model, for example:

Refund

?

Ask

✓

Samples

Weighting

Rows per page

5






Page 1 of 11 (53 total rows)

←

Previous

Next

→

What is the value of the item?	Have they returned the item?	Can it be resold?	How big is the item?	User decision	Actions
100	Yes	Yes	Small	Yes	
20	No	No	Small	No	
80	No	No	Medium	No	
120	No	No	Medium	No	
60	No	No	Large	No	

Back

Next

- a. Review the samples.



You can see more samples on a page by increasing the value in **Rows per page**.

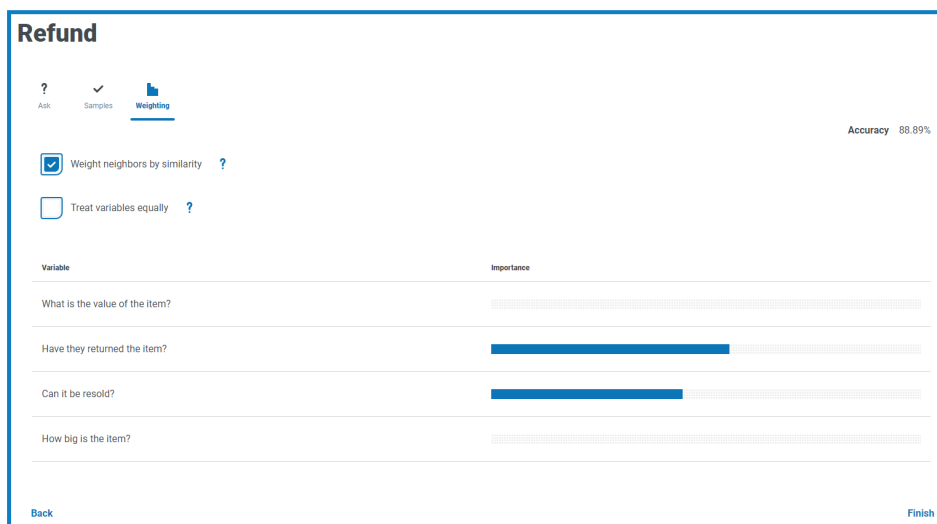
- b. If required, delete any obsolete or incorrect samples by clicking the associated delete icon in

the **Actions** column.

c. Click **Next**.

The **Weighting** tab displays.

- On the **Weightings** tab, the model displays its assessment of the importance of the variables based on the answers you provided when training the samples, for example:



It is recommended that the default values are used. However, if required, you can adjust these:

- If required, clear **Weight neighbors by similarity**. This is selected by default.

Neighbors are a sub-set of the samples that are considered to be similar to the values that have been entered for the variables. When this option is:

- Selected – Neighbors that are most similar to the values supplied will have a greater influence on the outcome than other neighbors.
- Unselected – All neighbor samples will have the same influence on the outcome.

- If required, select **Treat variables equally**. This is unselected by default. When this option is:

- Selected – All the variables will have the same level of influence on the outcome.
- Unselected – The model dynamically determine each variable's influence on the outcome, and this is demonstrated by the importance bars in the summary displayed.

- Click **Finish**.

Your model is now ready to use.



If required, you can re-train and re-calibrate your model at any point. The more you train your model, the greater the accuracy and confidence of the model.

However, if you create samples for all possible combinations of the variables on the Teach tab on of the Train page, you will not be able to create any samples to calibrate your model.

Using a model in an automation

The trained and calibrated Decision models can be used by Blue Prism to automate the decision-making process.



Depending on the type of decision-making process you are automating, you should consider building the requirement for a human reviewer into your process.

As part of the Decision installation process, the Decision API VBO is imported into the Blue Prism interactive client which sets the details for the Decision Web API Service and provides the action to use your Decision model. This is required to be able to use Blue Prism with Decision. For more information, see the [Decision install guide](#).

Use your Decision model in a process

Including your Decision model in a process enables the process to reach a conclusion based on a number of inputs, without the need to build a more complex process in Blue Prism.

To use your Decision model in a process:

1. In the Blue Prism interactive client, open the required process in Process Studio.
2. Add an **Action** stage and double-click it to open the Action Properties window.
3. In the Action Properties:
 - a. Enter a name and an optional description for the action.
 - b. Select **Utility – Decision** from the **Business Object** drop-down list.



If you have not entered a name for your action, the name now defaults to *Utility – Decision::Get Prediction*.

- c. On the Inputs tab, enter the identification number for the model in the **Value** column for the **Model Id**. You can find the model identification number on the [Library page](#) in Decision.

- d. On the Outputs tab, click the create icon in the **Store In** column for each output. Each cell is populated, for example:

Name	Data Type	Store In
Confidence	Number	Confidence
Prediction Response	Text	Prediction Response
Explanation	Collection	Explanation
Result	Collection	Result

- e. Click **OK**.

The updated action displays with the created collections and data items for the outputs.

4. Add a new **Collection** to your process (this will be used for the inputs), and double-click it to open the Collection Properties window.
5. In the Collection Properties:
 - a. Enter a name and an optional description for your collection.
 - b. On the Fields tab, add as many fields to this collection as you have variables in your model. Copy the name of each variable into each field and set the **Type** to **Text** for each field. The names must be identical.

The Weighting tab on the Calibrate page in Decision lists all the variables.

For example:

Refund

Should I give a refund?

Variable	Value
What is the value of the item?	
Have they returned the item?	
Can it be resold?	
How big is the item?	

Collection Properties

Name: Model variable inputs
Description:

Name	Type	Description
What is the value of t.	Text	
Have they returned t.	Text	
Can it be resold?	Text	
How big is the item?	Text	

Reset to Initial Value whenever this page runs ☒ Hide from other pages in the process ☒ Single Row ☐

OK Cancel

c. If required, on the Initial Values tab, add a single row and enter any initial values.

d. Click **OK**.

- Double-click your Action stage to open the Action Properties window.
- In the Action Properties, on the Inputs tab, drag and drop the collection you have created in [step 5](#) from the navigation tree into the **Value** for **Inputs**.

The name of the collection displays between square brackets, for example:

Action Properties

Name: Utility - Decision::Get Prediction
Description:

Business Object: Utility - Decision
Action: Get Prediction

Inputs Outputs Conditions

Name	Data Type	Value
inputs	Collection	[Model variable inputs]
Model Id	Number	39

Stage logging: Errors only ☐ Don't log parameters on this stage
Warning threshold: System Default Number of minutes: 5 (0 to disable)

OK Cancel

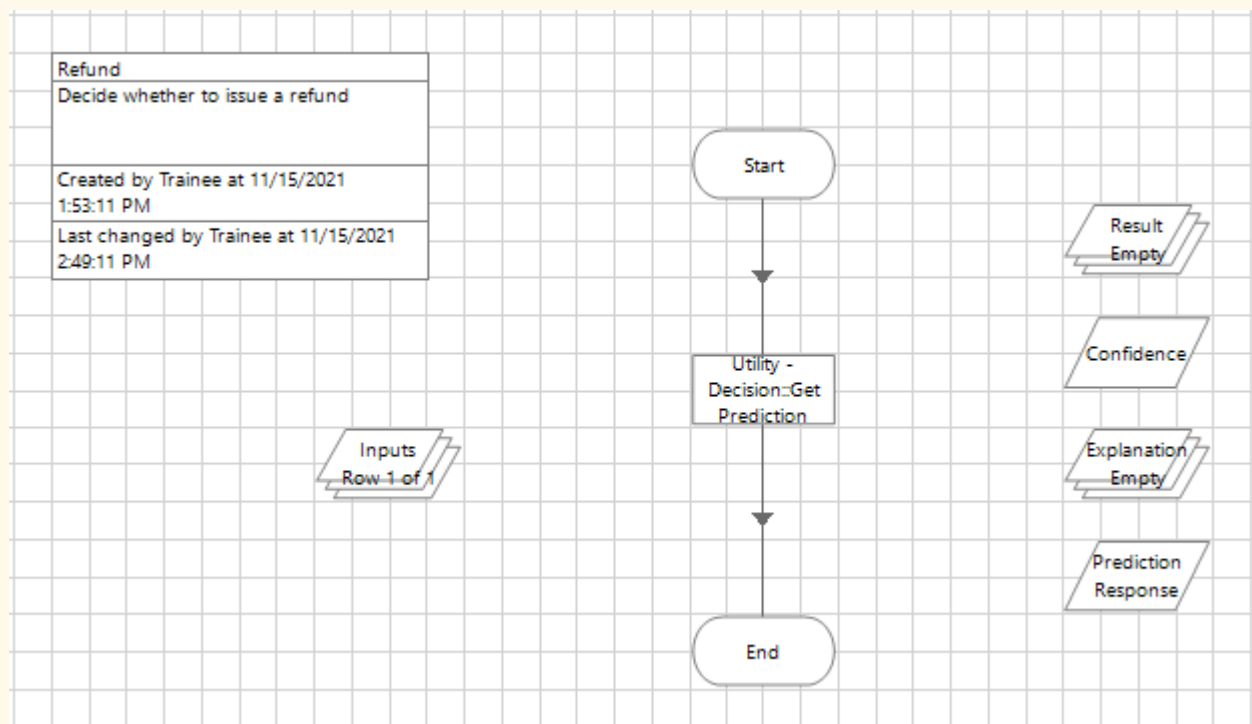
Binaries

- Collections
 - Explanation
 - Model variable inputs
 - Model variable input
 - Model variable input
 - Model variable input
 - Model variable input
 - Model variable input
 - Result
- Dates
- DateTimes
- Flags
- Images
- Numbers
- Passwords
- Text
- Times

8. Click **OK**.

You can now build the rest of your process around the decision model.

💡 If you want to test your connection and check that you can provide inputs and receive an outcome, create a new process with just a Start stage and a Stop stage. Add the Action stage as detailed above and use the Initial Values in [step 5c](#) to provide a set of inputs. Once you have completed the steps above, your process will look like this:



You can run this process and, if everything is set correctly, the output collections and data items will be populated.

If you receive an error, check that the model information is correct, and that the connection details and credentials are correctly configured.

Include a human reviewer

You can include a human reviewer or approver into the automation process. The example below is a suggested method and uses Blue Prism Interact as the interface for the user. Design an automation that suits your organization's requirements.

This example includes a human approver in the process to review the data and approve the outcome, where the model confidence level is below a defined threshold. In this case, below 80%.

To include an approver in your process:

1. In Blue Prism, open the required process in Process Studio.
2. At an appropriate place in your process, after the Action stage for your model, add a **Decision** stage and double-click it to open the Decision Properties window.

3. In the Decision Properties:

- a. Enter a name and an optional description for the stage.
- b. Enter an expression that you want to be true using the outputs that you received from the model Action stage. For example, using the Refund model shown in [Creating a new model on page 8](#), a valid expression is:

```
([Prediction Response] = "Yes" AND [Confidence] > 0.8)
OR
([Prediction Response] = "No" AND [Confidence] > 0.7)
```

In this example, two different values have been defined for the different responses. If the model's predicted response is:

- "Yes" and the confidence is greater than 0.8 (80%), the result will be true. If the confidence is 0.8 or less, the result will be false.
- "No" and the confidence is greater than 0.7 (70%), the result will be true. If the confidence is 0.7 or less, the result will be false.

Decision Properties

Name: Refund?

Description:

Expression

```
([Prediction Response] = "Yes" AND [Confidence] > 0.8)
OR
([Prediction Response] = "No" AND [Confidence] > 0.7)
```

Validate Evaluate Expression

Functions

- Conversion
- Data
- Date
- Environment
- Exceptions
- File
- Logic
- Number
- Text

Function Detail

Expression Function Builder

Use this area to compose a function statement and paste it into the expression.

Select a function from the list on the left. The function details and the required parameters will be shown here.

Complete the details either by entering values or by dragging in data items from the list on the right.

When the details are complete, send the

Data Items

Group:

☐ Page ☒ Data Type

☐ View All Items

- Binaries
- Collections
- Dates
- DateTimes
- Flags
- Images
- Numbers
- Passwords
- Text
- Times
- TimeSpans

Stage logging: Enabled

Warning threshold: System Default Number of minutes 5 (0 to disable)

OK Cancel

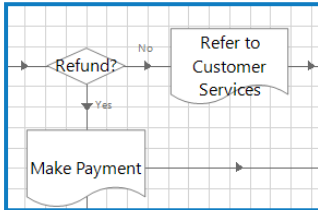
- c. Click **OK**.

The updated Decision stage displays.

4. Add the next steps in your process, for example:

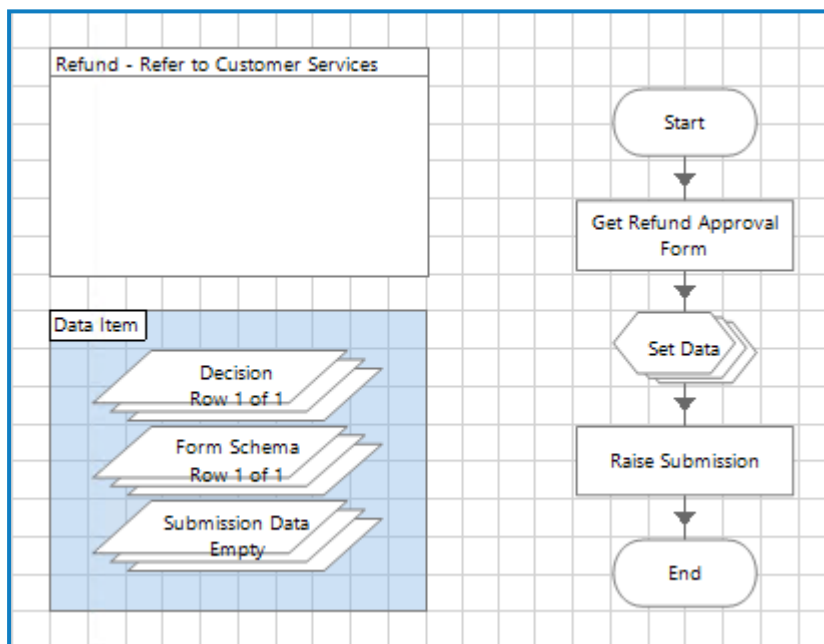
- If the result of the Decision stage is false (No in the image below), the process refers the information to an approver.
- If the result of the Decision stage is true (Yes in the image below), the process can continue following an automated path.

For example:



In this case, both routes call another page in the automation process. If the result of the Refund? Decision stage is:

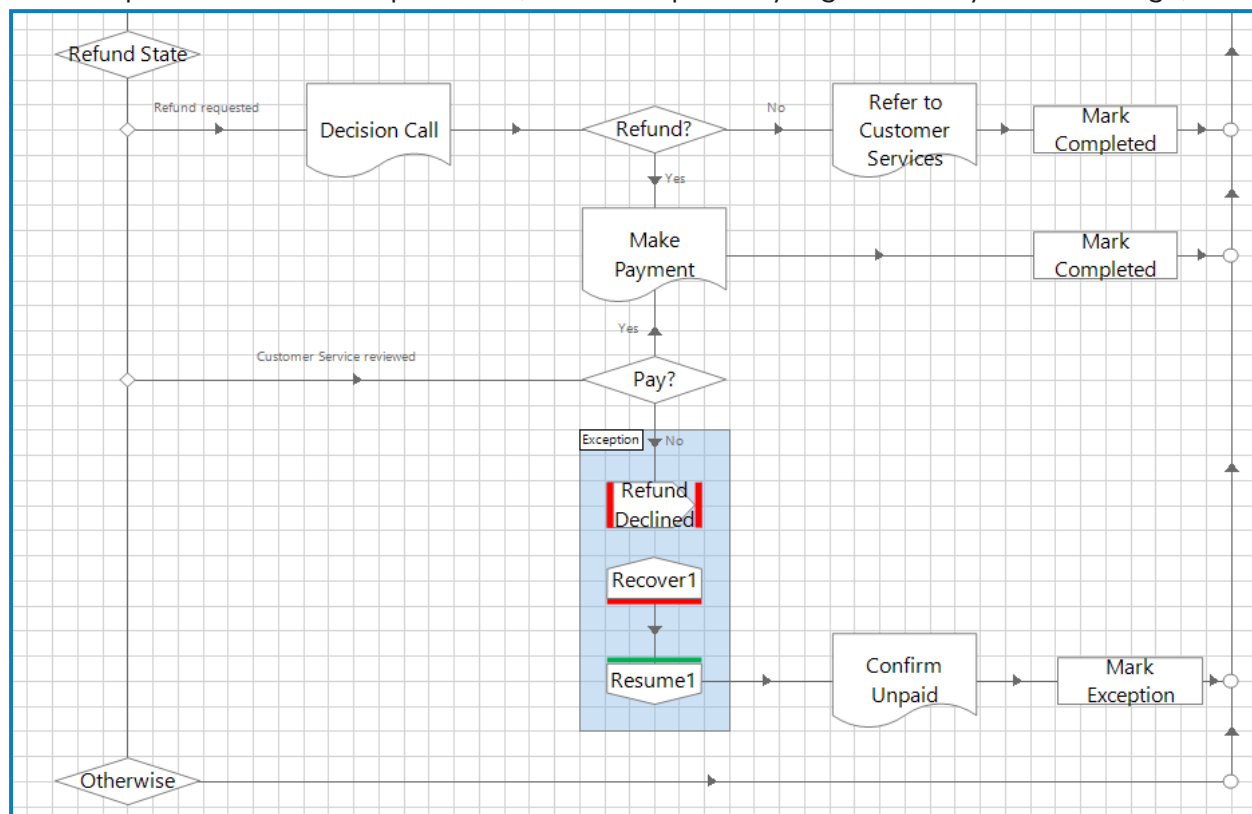
- Yes, the Make Payment page is used which triggers the refund to be issued.
- No, the Refer to Customer Services page is used, which sends the information to a human approver. The process on this page is:



This submits an Interact form to a Customer Services representative, where they can review the refund details and approve or decline the request. Their response is then submitted back to the work queue for the Digital Worker to action.

5. Add another Decision stage into your process to assess the response from the human approver and incorporate their response into the automation.

An example from the Refund process is, where the pathway is governed by a Choice stage, is:



Maintaining the models

Editing a model

You can edit any of the models that are shown on the Library page. The ability to edit a model depends on where the model is in the process of being defined, trained, and calibrated. If the model is:


- Defined but not trained or calibrated – You can change any of the fields in the model.
- Defined and trained but not calibrated – You can re-train the model by editing rules and creating samples. You cannot edit the outcomes or variables in the model definition, nor can you edit the Initialise tab on the Train page.
- Defined, trained and calibrated – You can re-train the model by editing rules and creating samples and re-calibrate the model. You cannot edit the outcomes or variables in the model definition, nor can you edit the Initialise tab on the Train page.

To edit a model:

1. On the Blue Prism Decision [Library page](#), find the model you want to edit.
2. Click the associated edit icon.
The Define page displays for the model you have selected.
3. Click **Next** until you reach the tab with the fields you want to change and edit the model as required. For more information, see [Creating a new model on page 8](#).
4. When the changes have been made, click **Next** and move through the rest of the process to create your updated model. For more information, see [Creating a new model on page 8](#).

Deleting a model

You can delete any of the models that are shown on the Blue Prism Decision Library page.

 If the model is being used by an automation, this will break that automation. Ensure the model is no longer in use before you delete it.

To delete a model:

1. On the Blue Prism Decision [Library page](#), find the model you want to delete.
2. Click the associated delete icon.
A message displays asking you to confirm that you want to delete the model.
3. Click **Yes** to permanently delete the model.

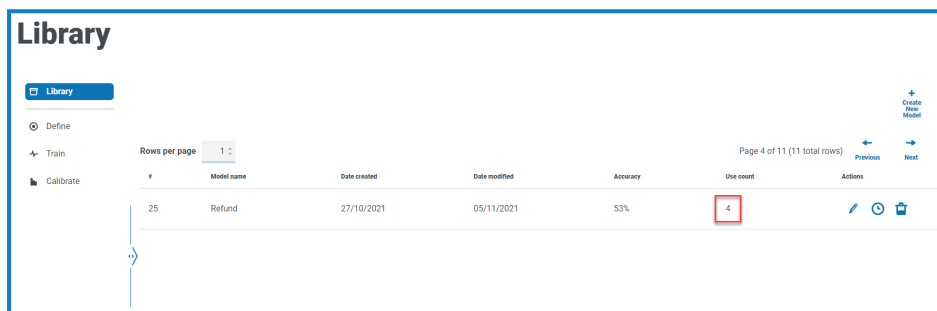
Viewing a model's usage

You can view the number of times a model has been used by an automated process, the inputs that it received, and the decision that was provided back to the automation. Models that are used by automations are indicated by the **Use Count** column on the Library page, which displays the number of times the model has been used.

To view the usage history:

1. On the Blue Prism Decision [Library](#) page, find the required model.

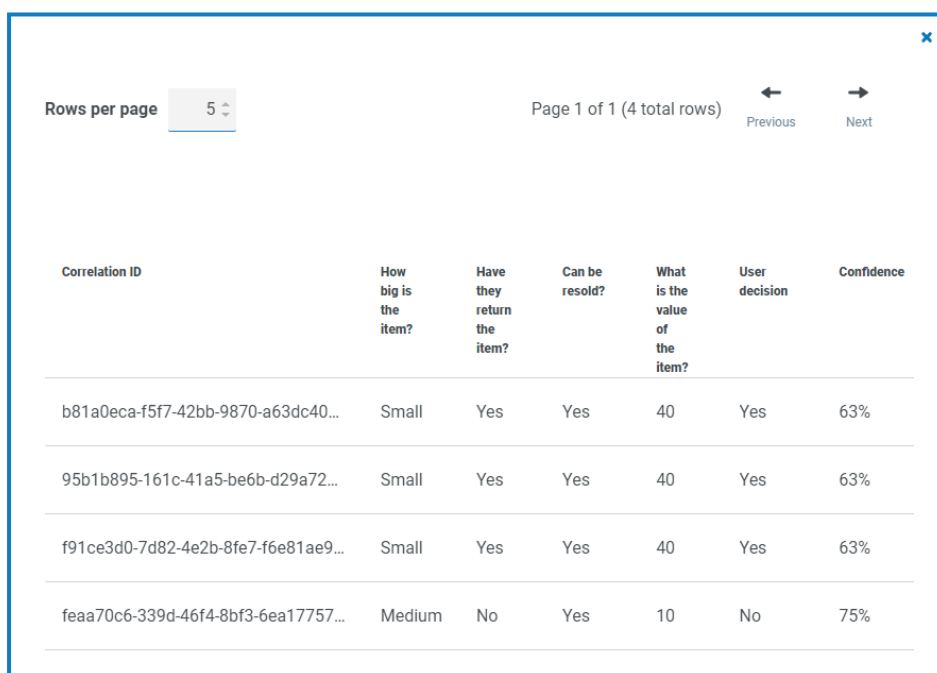
In this example the model has been used four times:



#	Model name	Date created	Date modified	Accuracy	Use count	Actions
25	Refund	27/10/2021	05/11/2021	53%	4	[Edit] [Refresh] [Delete]

2. Click the **predictions list** icon.

The list of predictions displays with the inputs and the resulting output, along with the confidence level of the output and the date and time it was generated. The Correlation ID can be traced back to the automation request that used the model.



Correlation ID	How big is the item?	Have they return the item?	Can be resold?	What is the value of the item?	User decision	Confidence
b81a0eca-f5f7-42bb-9870-a63dc40...	Small	Yes	Yes	40	Yes	63%
95b1b895-161c-41a5-be6b-d29a72...	Small	Yes	Yes	40	Yes	63%
f91ce3d0-7d82-4e2b-8fe7-f6e81ae9...	Small	Yes	Yes	40	Yes	63%
feaa70c6-339d-46f4-8bf3-6ea17757...	Medium	No	Yes	10	No	75%




Use the horizontal scroll bar to view the **Time stamp** column.



The model decision is shown under the **User decision** column.

Glossary of terms

 The terms are listed in alphabetical order.

Term	Description
Accuracy	The percentage of correct predictions a model makes for categorical decisions. This is more complex for numerical decisions.
Active learning	A machine learning (ML) technique which asks a human-in-the-loop to provide outcomes for specific inputs. These inputs are used by the machine to optimize the rate of learning and minimize the number of interactions with the human-in-the-loop. See also Guided learning .
AutoML (Automatic Machine Learning)	A machine learning system that requires no user input to fit data, analyze the relationships between the variables, and make predictions on new data. A typical example is the Automatic Statistician.
Binary variable	A categorical variable that takes one of two values, typically true or false.
Calibration	The process of reviewing model inputs and outcome predictions, and overwriting model decisions if required.
Categorical variable	A variable that takes one of a fixed set of values. If the values are unordered then it is a “nominal” categorical variable, otherwise if the values are ordered it is an “ordinal” categorical variable.
Certainty	See Confidence .
Condition	Part of a rule. All the rule's conditions must be satisfied in order for the rule to apply. For example: stock room = full. Also known as 'constraint'.
Confidence	This defines how sure a model is in its prediction, for example how many times out of 100 it expects to be correct. Also known as 'certainty'.
Data type	The type of a variable in a problem. Can be categorical or numerical.
Decision	See Outcome variable .
Exception	See Rule .
Guided learning	A machine learning technique which asks a human-in-the-loop to provide inputs that generate specific outputs. These outputs are used by the machine to optimize the rate of learning and minimize the number of interactions with the human-in-the-loop. See also Active learning .
Human-in-the-loop	A human that provides guidance to a machine learning method or automation.
Input variable	Input to the decision making process.
Input importance	A relative value that a model gives to an input variable compared to the other input variables to determine its importance. A higher value indicates that the model treats that variable as more important. Some models permit the user to alter these weights. Also known as feature importance or feature weight.

Term	Description
kNN model	A machine learning or statistical non-parametric model that makes predictions (categorical or numerical) for an input based on the k nearest neighbors (kNN) to the input in the training data. The nearest neighbors can be considered equally (an unweighted kNN model) or weighted by their inverse distance or similarity (a weighted kNN model).
Machine learning	Systems in which machines learn by themselves.
Model	A machine learning system that predicts an outcome (decision) based on a given a set of features.
Neighbors	A subset of the samples that are considered to be similar to the values that have been entered for the variables. See kNN model .
No Code	Systems that can be configured without programming. These are typically designed for a business user rather than an engineer to use.
Nominal variable	An unordered categorical variable. For example, “cat”, “horse”, “dog”.
Non-parametric model	A machine learning model that makes few assumptions of the structure of the problem and fits to the data. Typically, these models are more flexible than standard statistical parametric models.
Numerical variable	A variable that takes any value in a range. For example, 1, 1.2, 100, -3.
Ordinal variable	An ordered categorical variable. For example, “small”, “medium”, “large”.
Outcome variable	The outcome of the decision making process. Also known as ' decision '.
Rule	A user-defined exception to the Decision model. This is a list of conditions or constraints that when satisfied force the outcome to a particular value. Also known as 'exception'.
Sample	A data point that is one complete set of input variable values possibly also associated with an outcome variable value.
Unweighted	See kNN model .
Variable	Generic term for either an input or outcome (decision) variable.
Weighted	See kNN model .