

IBM Cloud Pak for Business Automation Demos and Labs 2021

IBM Automation Decision Services
Manage Decisions and infuse Machine Learning

V 1.0

Decisions
dba_decisions_I3_services@wwpd1.vnet.ibm.com

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2020.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

1 Introduction	4
1.1 IBM Automation Decision Services.....	4
1.2 Lab Overview.....	5
1.3 Lab Setup Instructions.....	5
1.3.1 Prerequisites	5
1.3.2 Log in to the environment	6
1.3.3 Navigating through the environment	6
2 Exercise 1: Modeling Decisions.....	9
2.1 Introduction	9
2.2 Exercise Instructions.....	9
2.2.1 Creating a project and importing a decision service.....	9
2.2.2 Exploring the decision service: data model and decision model	11
2.2.3 Updating a decision logic: adding a node, a rule, editing a decision table	18
2.2.4 Validating the decision service	26
2.3 Summary	27
3 Exercise 2: Adding machine learning in the decision model.....	28
3.1 Introduction	28
3.2 Exercise instructions	28
3.2.1 Log in to your project.....	28
3.2.2 Creating a project and importing a Decision service	28
3.2.3 Testing the model before changes.....	29
3.2.4 Defining a Machine learning provider	31
3.2.5 Creating and connecting the Predictive Model	33
3.2.6 Defining the mapping	37
3.2.7 Validating the predictive model	45
3.2.8 Use the predictive model in the scoreboard decision model	48
3.2.9 Validating the final decision service.....	50
3.3 Summary	50
4 Exercise 3: Sharing and publishing decision services	52
4.1 Introduction	52
4.2 Exercise instructions	52
4.2.1 Log in to your project.....	52
4.2.2 Creating a project and importing a Decision service	52
4.2.3 Sharing your decision service	54
4.2.4 Creating a version	56
4.2.5 Creating a Git repository	57
4.2.6 Connecting your project to the Git repository	57
4.2.7 Deploying your project	58
4.2.8 Publishing your decision service through Decision Designer	59
4.3 Summary	60
5 Additional information.....	61

1 Introduction

In this Lab, you build business operational decisions to implement services evaluation and client risk assessment for a Client Onboarding solution. You capture and automate these decisions using IBM Automation Decision Services.

It includes three exercises. Each exercise can be done separately.

Duration: About 3 hours (each exercise is about 1 hour).

Audience: Anyone who wants to learn how to use Automation Decision Services.

1.1 IBM Automation Decision Services

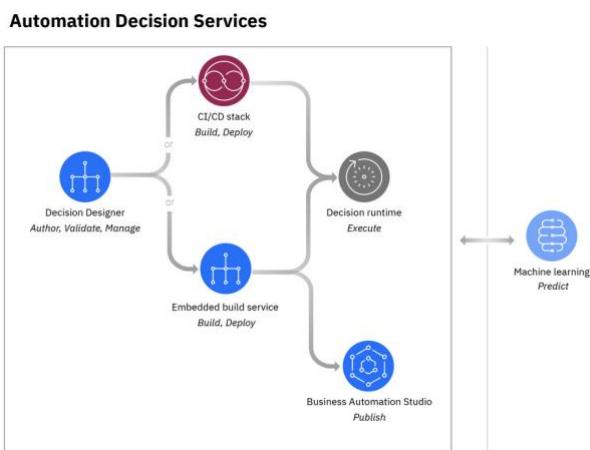
IBM Automation Decision Services provides a comprehensive environment for authoring, managing, and running decision services.

Operational decisions apply business policies, which are often influenced by numerous factors that can be both internal and external to an organization.

Part of the IBM Cloud Pak® for Business Automation platform, Automation Decision Services delivers advanced decision automation capabilities. Using Decision Designer in Business Automation Studio, business experts can model, author, and validate decisions in one development environment. They can also infuse intelligence into business decisions by combining decision models and predictive models into decision services. They can collaborate with others in their organization by sharing projects through a central Git repository.

Automation Decision Services integrates with a continuous integration and delivery (CI/CD) stack.

You can build and deploy decision services directly from Decision Designer. Deployed decision services can then either be published as automation services in Business Automation Studio, or invoked through the decision runtime.



For more information, see IBM Documentation

[IBM Automation Decision Services](#)

[What is Automation Decision Services](#)

1.2 Lab Overview

The end-to-end Client Onboarding solution explores a generic use case for onboarding new services to an existing client. Automation Decisions Services provides two decisions services that are integrated in the solution.

The first decision is to define the fees for the services the client is being onboarded to, and to suggest additional services the client could be interested in, based on the information gathered. It is consumed in the end-to-end solution via an automation service from a Business Automation Application in the front-office in-take app.

The second decision provides input to determine the onboarding approval, the scoreboard. It implements a client risk evaluation based on multiple information such as the client profile, industry and a client classification based on the client revenue. It combines descriptive and predictive decisions. It is consumed in the end-to-end solution as an automated service in Workflow, during the approval step.

In this Lab you learn how to:

1. Manage business decisions based on multiple data: client, industry information.
2. Infuse intelligence into business decisions by adding a predictive model into the decision service for the scoreboard of the client.
3. Collaborate by sharing projects through a GitHub repository, build and deploy decision services from Decision Designer. Learn how to publish an automation services in Business Automation Studio.

Integrating the automated services in other applications is not covered in this Lab, look at the Workflow and Business Automation Application labs for this.

1.3 Lab Setup Instructions

1.3.1 Prerequisites

1. If you are performing this Lab as a part of an IBM event, access the document that lists the available systems and URLs along with login instructions. For this lab, you will need to access:

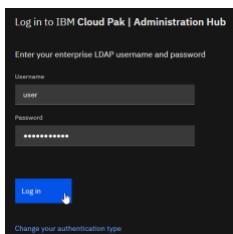
- **IBM Business Automation Studio** to access Decision Designer.
- **ADS ML Service** (Machine learning providers details).

2. Download the file **ClientOnboardingLab.zip** from the Lab Data folder onto your computer.

GitHub account is needed for Exercise 3. You use a GitHub account to create a repository. If you do not have an account, create an account in [GitHub](#). Your work will be saved in your repository.

1.3.2 Log in to the environment

1. Launch Business Automation Studio (URL listed in the Lab access document, see prerequisites)
2. Select your authentication type: **Enterprise LDAP**
3. Enter your *Username* and *Password* then click on **Log in**



1.3.3 Navigating through the environment

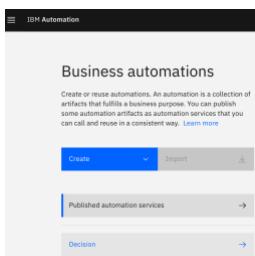
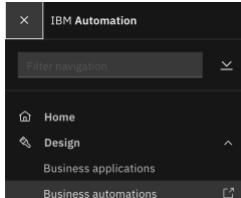
In the Lab you navigate in Business Automation Studio and Decision Designer. You work with decision projects and decision services. In each exercise, you create and/or work with your project and decision services and discover the interface in context.

This section is an introduction to the navigation. You look at the screens to discover the navigation, there are no steps to take. **This is for reading only and there are no steps to be performed.** It is provided as a reference for the Lab while working the exercises, once you work with a project.

1. Navigate to Business automations from the Navigation Menu on the top left corner.



2. Open Business automations from Design -> Business automations by expanding the menu.



3. Open a decision automation by selecting Decision (either Create or hover over and open existing Decision automations) to open a project. **Note:** This is for reading, to discover the navigation. You will open the decision automation once working the exercises.

The breadcrumb shows Business Automation and below the name of a project (Myuser Client Onboarding).



Business Automations
Myuser Client Onboarding :

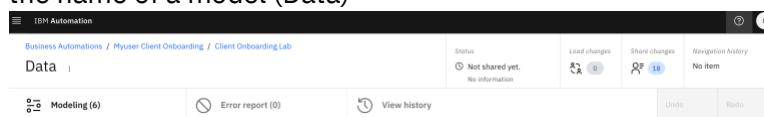
4. Open the decision services that are in the project by clicking the tile of the decision service.

The breadcrumb shows Business Automations / Project name and below the name of a decision service (Client Onboarding Lab).



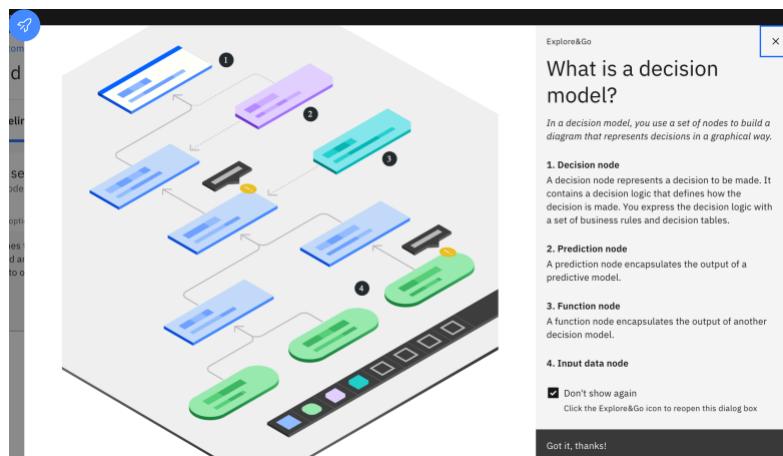
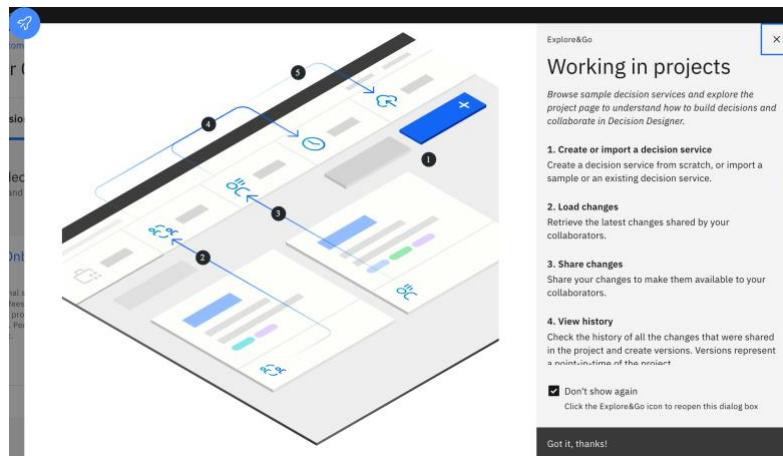
5. Open a model in the decision service by clicking one tab, then selecting a model.

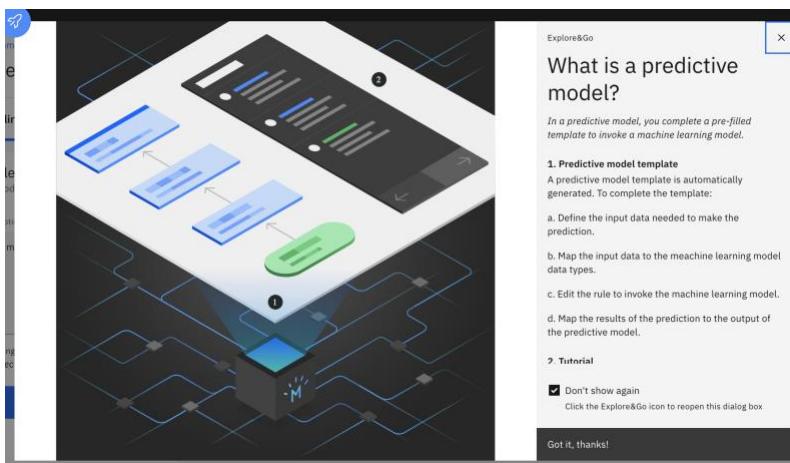
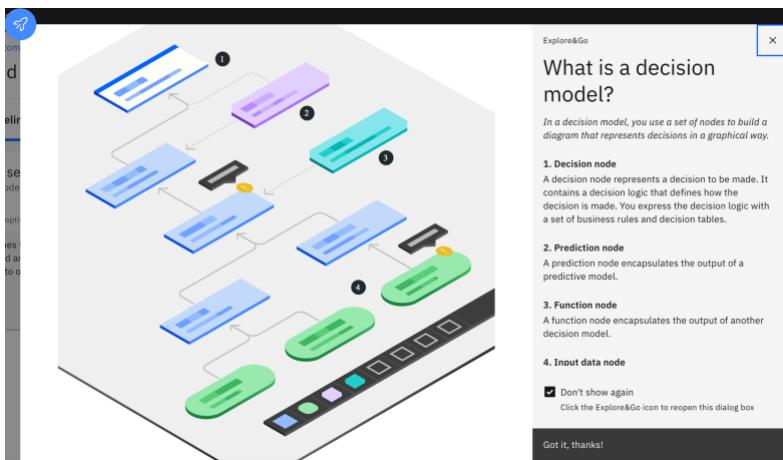
The breadcrumb shows Business Automations /Project name / Decision service name and below is the name of a model (Data)



6. Explore&Go

The Explore&Go dialog box may open. It provides contextual help on Automation Decision Services concepts. You can dismiss if it shows up.





You can open the contextual dialog box anytime by clicking the Explore&Go icon at the down left corner of the screen.

2 Exercise 1: Modeling Decisions

2.1 Introduction

This exercise is an introduction to the authoring environment of Automation Decision Services. You navigate in Decision Designer which is the development environment for creating decision services. You discover the main concepts of Automation Decision Services by exploring and modifying a given decision service. The scenario of this decision service is to define the fees for services the client is being onboarded to and to suggest additional services the client could be interested in, based on the information gathered.

2.2 Exercise Instructions

In this exercise you import a decision service and explore it:

- Create a **project** and import a **decision service**
- Explore the **data model** and the **decision model**
- Add a business rule and edit a decision table to update the **logic** used in the model
- Validate the decision service

Prerequisites: you need to download the provided file **ClientOnboardingLab.zip** from Lab data folder.

2.2.1 Creating a project and importing a decision service

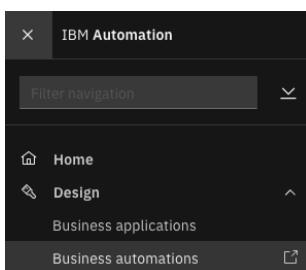
In this step, you create a project in Business Automation Studio. You import a decision service from the zip prepared for this exercise.

A project is a set of artifacts that share the same lifecycle and are grouped to solve a particular business problem. In this Lab you work with a decision automation project, it provides decision-modeling capabilities to help business experts to capture and automate repeatable decisions.

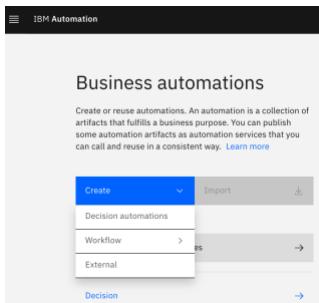
If you are already connected to Automation Decision Services and have your project opened following a previous exercise, you can skip this section and pursue in next section [Exploring the decision service: data model and decision model](#).

If you begin the Lab start with this exercise, follow the steps below.

1. Log in Business Automation Studio
2. Click on Navigation Menu on the top left corner
3. Expand Design, click on Business automations

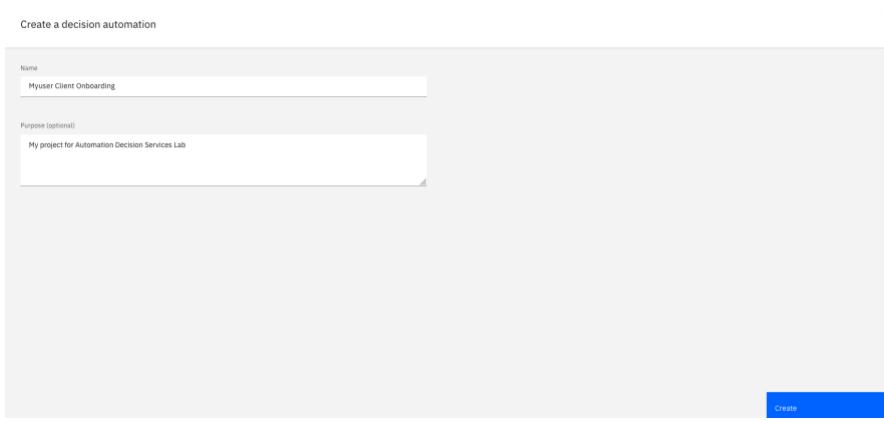


4. Click on Create and select Decision automations



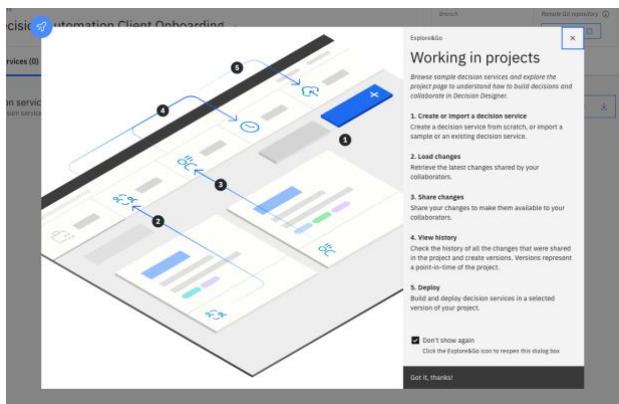
5. Enter a name for your project. Enter **UsrNNN Client Onboarding Decision** where *UserNNN* is your assigned username (a name starting by your username to avoid conflicts with other projects).

6. Click on **Create**

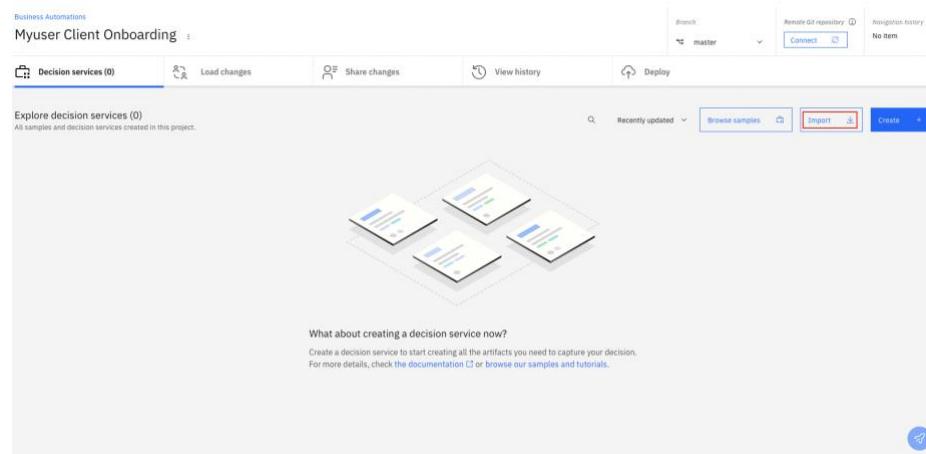


7. **Explore&Go** dialog box may open. You can dismiss it if it shows up.

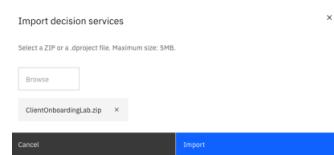
It provides contextual help on Automation Decision Services concepts. You can open contextual dialog box anytime by clicking the  Explore&Go icon at the down left corner of the screen.



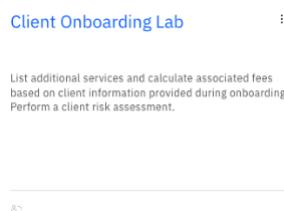
8. Click on **Don't show again**, Close the dialog box.
9. Your new project opens in Decision Designer. Click on **Import** to import the decision service provided by the Lab team.



10. Browse to select the project prepared for the Lab **ClientOnboardingLab.zip** and click on **Import**



11. A tile of a decision Service named **Client Onboarding Lab** appears on the project page.



A decision project can be composed of several decision services. You create them or import them from the samples or from a zip file. A project can be shared with other users and can be connected to a Git repository. These features will be covered in Exercise 3.

2.2.2 Exploring the decision service: data model and decision model

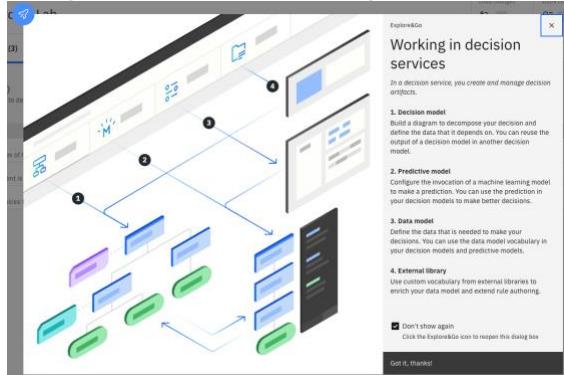
In this step, you explore the decision service you imported. A decision service uses decision artifacts to define the business decision.

- **Decision models:** represent a diagram that expresses the business logic. You can reuse the output of a decision model in another decision model.
- **Predictive models:** apply data from a machine learning model.
- **Data models:** represent the data structure used by the business logic. You can use the data model vocabulary in your decision models and predictive models.
- **External libraries:** to extend the data models using java code or contain data models and functions that you can use in decision models

For more information, see IBM Documentation [Modeling decisions](#)

1. Click on **Client Onboarding Lab** to open your decision service.
2. **Explore&Go** dialog box may open. You can dismiss it if it shows up.

It provides contextual help on Automation Decision Services concepts. You can open contextual dialog box anytime by clicking the  Explore&Go icon at the down left corner of the screen.



3. Click on **Don't show again**, Close the dialog box.
4. Your decision service is displayed.

Name	Last updated
Fee and services	usr100 6/2/2021, 2:19:32 PM
Scoreboard	usr100 6/2/2021, 2:19:32 PM
Services subset	usr100 6/2/2021, 2:19:32 PM

2.2.2.1 Exploring the data model

A data model defines the data that is needed to make your decisions. In a data model, you use predefined and custom data types to create a vocabulary that you can use to populate decisions. Basic data types, including string and integer are predefined by default in Automation Decision Services. You create a data model to define custom data types to match specific needs for your service. These types are defined in a data model and can be used in one or more decision models in a service.

1. Open the **Data models** tab and click on **Data**

Name	Last updated
Name	Me Never shared
Data	

The data model is a collection of **Data types** that represents the data needed to make the decision. This collection is used by all the decision models defined in the decision service. This model for Client Onboarding contains 6 **Data types**: Client Information, Industry, Scoreboard, Service, Service Information, Service Request.

```

graph TD
    ClientInformation[Client Information] --- Industry[Industry]
    ClientInformation --- Scoreboard[Scoreboard]
    ClientInformation --- Service[Service]
    ClientInformation --- ServiceInformation[Service Information]
    ClientInformation --- ServiceRequest[Service Request]

```

In the decision model for this exercise, 5 of these data types are used: Client Information, Industry, Service, Service Information and Service Request.

Two categories of data types can be defined:

- the **enumeration type** such as Industry and Service that contains a list of possible values.
- the **composite type** such as Client Information, Service, Service Information, Service Request, that contain a set of attributes with a name and a type. The type can be a basic type predefined in Automation Decision Services such as string or integer or a custom data type.

The types defined as input of the model for this exercise (Fee and services) are:

- the enumeration **Industry**: lists the available industries
- the enumeration **Service**: lists the available services
- the composite type **Service request**: describes in which industry the client is and what are the services requested
- the composite type **Client Information**: describes the characteristics of the client: annual revenue, number of employees, company age, defaulted payment.

The type defined as output of the model for this exercise (Fee and services) is:

- Service Information: gives a list of additional services suggested to the client and the fee for the services requested.

2. Click on **Client Information** to explore the data types. It is a Composite type composed of 4 **attributes**. Attributes and values allow you to define the characteristics of a data type

Name	Type	List
Annual Revenue	integer	<input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/>
Company Age	integer	<input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/>
Defaulted Payment	boolean	<input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/>
Number of Employees	integer	<input type="button"/> <input type="button"/> <input type="button"/> <input type="button"/>

The **verbalization** defines how a data type is referenced in business rules. It depends on the type of element. Each attribute comes with a set of automatically generated expression and action phrases.

For more information, see IBM Documentation [Creating a data model, Default verbalization](#)

3. Explore each **data type** and **attributes** to learn more about the data types.

For more information, see IBM Documentation [Defining a data model](#)

4. Click on **Client Onboarding Lab** in the breadcrumb to navigate back to your decision service.

The screenshot shows the IBM Automation interface with the following details:

- Header: IBM Automation
- Breadcrumb: Business Automations / Myuser Client Onboarding / Client Onboarding Lab
- Top navigation tabs: Decision models (3), Predictive models (0), Data models (1), External libraries (0)
- Section title: Decision models (3)
Description: Decision models allow you to decompose the decisions that you want to automate.
- Table listing three decision models:

Name	Last updated
Fee and services	Me Never shared
Scoreboard	Me Never shared
Services subset	Me Never shared

2.2.2.2 Exploring the decision model

You define the structure of a decision in a diagram, in a model. The primary part of the diagram are nodes:

- **Decision nodes:** represent the decision and contain a decision logic that defines how each decision is made.
- **Input data nodes:** represent the data used to determine the decision output. They are associated with a data type.
- **Function nodes:** represent values that are computed from other decision models.
- **Prediction nodes:** represent values that are computed in machine learning predictive models (prediction nodes are not used in this exercise, they are used in Exercise 2)
- **Links** represent the relationship between the decisions and input data, the invocation of a function or prediction

The rules define the logic of the decision and are expressed in business rules or decision tables. The rules determine the output of the model.

1. In the Decision models tab, click on **Fee and services**.

The screenshot shows the IBM Automation interface with the following details:

- Header: IBM Automation
- Breadcrumb: Business Automations / Myuser Client Onboarding / Client Onboarding Lab
- Top navigation tabs: Decision models (3), Predictive models (0), Data models (1), External libraries (0)
- Section title: Decision models (3)
Description: Decision models allow you to decompose the decisions that you want to automate.
- Table listing three decision models:

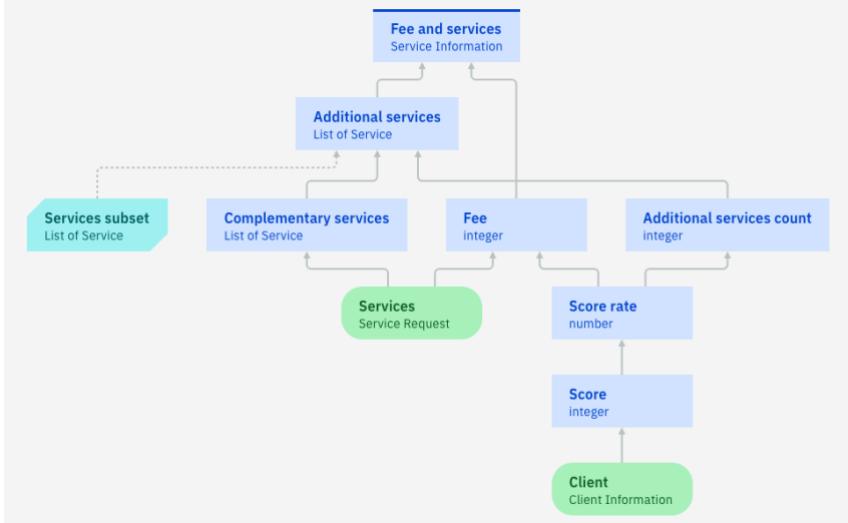
Name	Last updated
Fee and services	Me Never shared
Scoreboard	Me Never shared
Services subset	Me Never shared
- Create button: Create +

It opens the Decision model Fee and services. Explore&Go window may open, you can dismiss it.

In Automation Decision Services, you implement your decision by building a decision model diagram. The diagram shows how the decision depends on several sub-decisions.

The modeling diagram is based on the [Decision Model and Notation \(DMN\) standard](#).

The diagram below shows the decision model for **Fee and services**. As explained in the [Exploring the decision mode introduction](#), the diagram is composed of nodes and links. The decision nodes are blue, input nodes are green and function nodes cyan. The links represent the relationship between the decisions and the inputs. A link shows the input that is available in a node.



In this exercise, the top decision **Fee and services** is the final decision. It calculates the fees for the services requested by the client and suggests additional services to offer. It is based on the **Client** information and the **Services** requested as input. To provide the outcome, it depends on intermediate steps, sub-decisions.

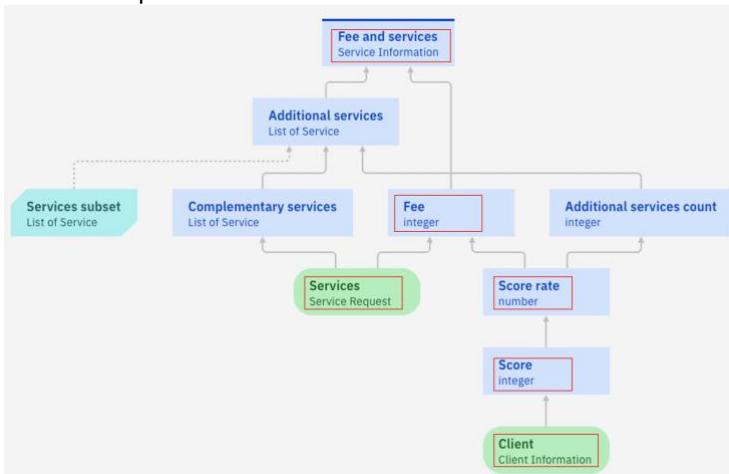
The final decision node **Fee and services** depends on two sub-decisions: **Fee** and **Additional services**.

Fee

The fee is computed by the **Fee** decision node, based on the service list requested by the input node **Services** and the **Score rate** decision node.

The **Score rate** decision node computes a rate to apply to the default fee for the client based on a score computed by the **Score** decision node.

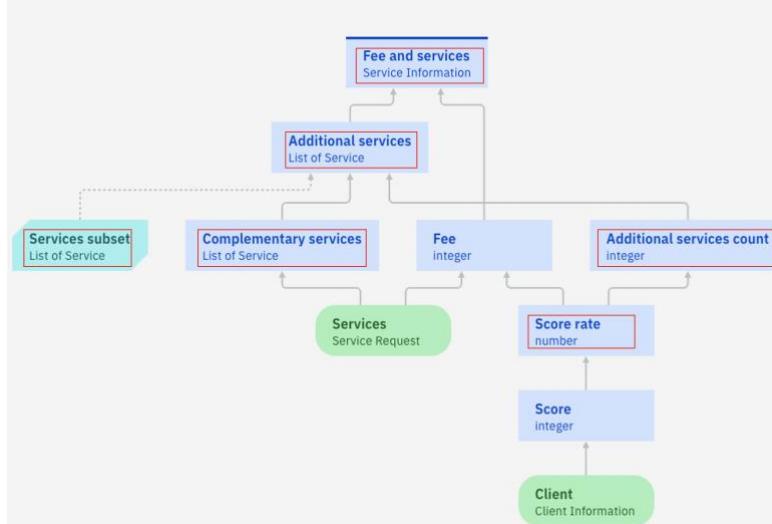
The **Score** decision node computes the client score from its different characteristics described by the **Client** input node.



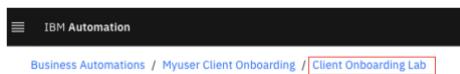
Additional services

The additional services list is computed by the **Additional services** decision node. It takes a subset of the list computed by the **Complementary services** node.

To extract the subset of services it depends on the function node **Services subset**. The subset of the list is computed by taking the count of elements from the list. In the exercise, this computation is arbitrarily simplified. The count is computed by **Additional service count** decision node from the **Score rate** decision node.



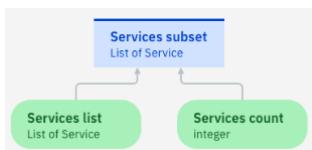
2. Services subset is a function node. It is computed from another decision model named Services subset. Click on **Client Onboarding Lab** in the breadcrumb to navigate back to the **Decision models** tab.



3. To see its diagram, select **Services subset** in the Decision models list:

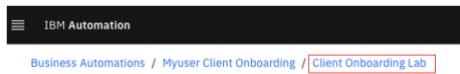
The screenshot shows the 'Decision models' list. There are three items: 'Fee and services', 'Scoreboard', and 'Services subset'. The 'Services subset' item is selected, as indicated by a red arrow. The list includes descriptions and last update information for each model.

Name	Last updated
Fee and services	Me Never shared
Scoreboard	Me Never shared
Services subset	Me Never shared



For more information, see IBM Documentation [Creating decision model](#)

4. Return to the Decision model Fee and services. Click on **Client Onboarding Lab** in the breadcrumb to navigate back to the Decision models tab.



5. In the Decision models tab, click on **Fee and services**

The screenshot shows the 'Decision models' list page with three items: 'Fee and services', 'Scorecard', and 'Additional services'. The 'Fee and services' item is selected. A red arrow points to the 'Name' column for the 'Fee and services' entry.

6. Click on the **Fee** node, then select the **Logic** tab to explore the decision logic. It displays the decision table **Default services fees** and the rule **apply score rate** that implement the business logic of the fee calculation. In Automation Decision Services, you express the decision logic with a set of business rules and/or decision tables.

The left side shows the 'Fee' decision node details and logic tabs. The logic tab is selected, and a red arrow points to the 'Rules are applied in sequence' dropdown. The right side shows a flowchart of the decision logic:

```

graph TD
    FS[Fee and services] --> AS[Additional services]
    AS --> FI[Fee integer]
    FI --> SR[Score rate number]
    SR --> SC[Score integer]
    SC --> CI[Client Client Information]
    CI --> SS[Services subset]
    SS --> CS[Complementary services]
    CS --> FI
    FI --> AS
    FI --> ASC[Additional services count]
    ASC --> SR
    
```

7. Click on Default services fees in the **Logic** tab. It opens the decision table.

The screenshot shows a decision table with two columns: 'Services count' and 'Fees'. The 'Services count' column has values 0, 1, 2, 3, 4, and 5. The 'Fees' column has corresponding values 0, 15,000, 25,000, 38,000, and 50,000. A red arrow points to the first row of the table.

Services count	Fees
0	0
1	15,000
2	25,000
3	38,000
4	50,000
5	
6	
7	
8	
9	
10	

Each row represents a single decision where the **Services count** column represents a condition parameter and the **Fees** column the value of the action when the conditions are met.

8. Click on apply score rate in the **Logic** tab. It opens the business rule.

The screenshot shows the 'apply score rate' rule editor. It contains a code editor with the following content:

```

1 set decision to round( decision + 'Score rate', 0 );
2

```



It is a simple rule statement that determines a rate to apply to the fee calculation.

9. Click on **Inputs**, then **Output** at the bottom of the screen.

The rule uses the vocabulary of the input data **Services** and **Score rate** and selects a value for the output of the decision **Fee**.

In this decision logic, the rules are applied in sequence. A fee is assigned according to the number of services. Then, a rate is applied to calculate the final fee for the requested services.

To define how the rules interact with each other, you select a **rule interaction policy**.

10. On the **Logic** tab, expand **Rules are applied in sequence**.

The screenshot shows a user interface for defining decision logic. At the top, there are two tabs: 'Details' and 'Logic'. The 'Logic' tab is selected. Below the tabs, a section titled 'Rules are applied in sequence' is expanded. This section contains five items, each with a brief description:

- First rule applies**: When a rule is applicable, the decision is taken and cannot be changed by other rules.
- Choose greatest value**: When one or more rules are applicable, the decision is set to the greatest value.
- Choose smallest value**: When one or more rules are applicable, the decision is set to the smallest value.
- Sum all values**: When one or more rules are applicable, the decision is the sum of all values.
- Rules are applied in sequence**: When a rule is applicable, it can modify the decision made by the previous rule. This option is currently selected.

The interaction policy choices differ according to decision node type (number, list, integer...) In this exercise, the decision node type Fee is an integer, the options above are displayed for the integer type.

For more information, see IBM Documentation [Interaction policies](#)

2.2.3 Updating a decision logic: adding a node, a rule, editing a decision table

You change the decision logic for the Score computation to take into account the number of services requested by the client. A higher number of services leads to a higher score. The change in the decision logic implies:

- knowing the number of requested services from the score node. That requires adding a link from the services input and the score node.
- adding the computation from the number of services. That requires adding a rule to compute the new score.
- adapting the score rate computation to the new score range.

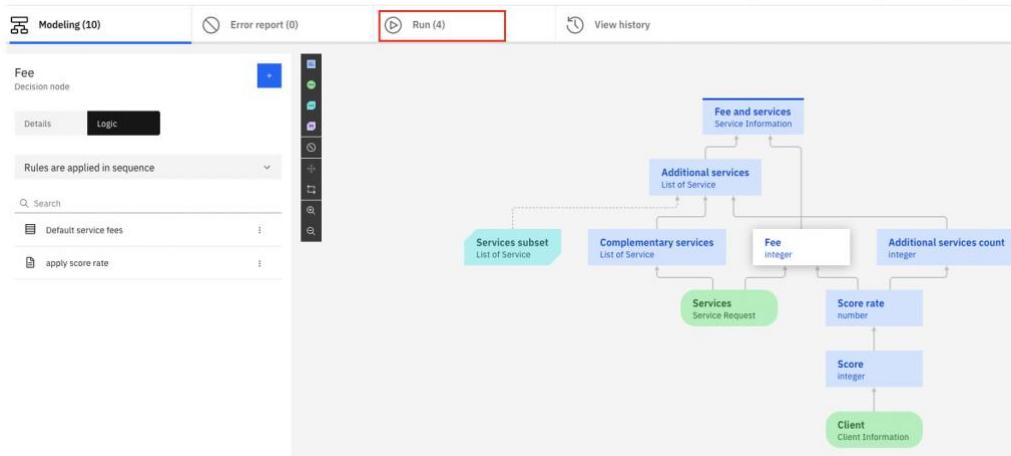
Before modifying the decision logic, you check the current behavior of the rule execution.

2.2.3.1 Test the decision logic

1. Click on **Back to the diagram** to return to the decision diagram



2. Select Run tab



On the left side of the screen, the **Test data** pane allows to select the set of data to submit to execute the rules. In this Lab, 4 sets of data are pre-defined: All Services, Federal 2, Federal 3, Telecom 1.

3. Select All Services and click on Run.

Node Name	Result
Fee and services	{ "extendedServices": [], "servicesFee": 50000 }

4. Expand **Run history**, then expand **Score** node

Run history			
Node	Rules	Rule Interaction	Output
Fee and services	1	Sequence	{ "extendedServices": [], "servicesFee": 50000 }
Fee	2	Sequence	50000
Additional services	1	Sequence	0
Additional services count	1	Sequence	2
Score rate	1	Sequence	1
Complementary services	2	Sequence	0
Score	3	Sum	2

Triggered rules	Number of runs	Output
revenue score	1	1
size score	1	1
age score	1	2

The output displays the result for the decision node Score, it is a consolidated result. In this example, Score decision node output is 2:

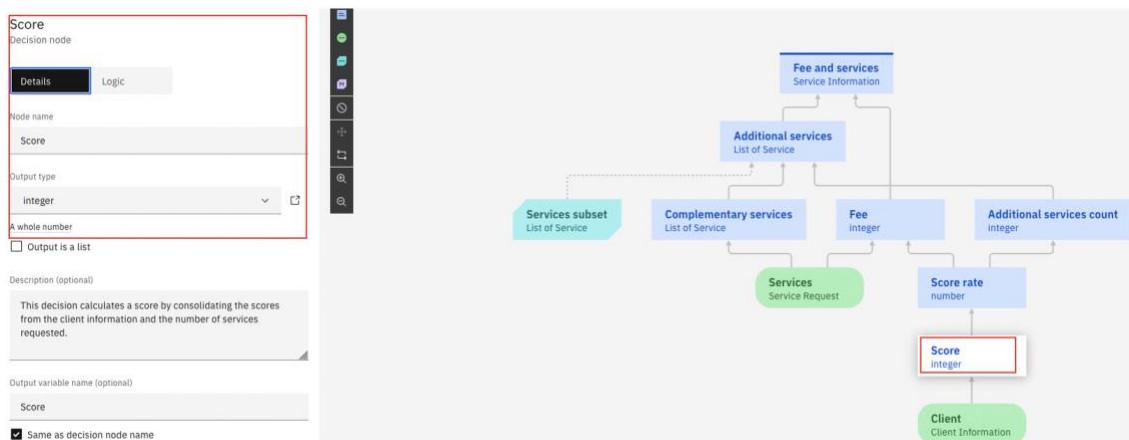
- revenue score output is 1,
- then added to size score (size score is 0) it remains 1,
- then added to age score result is 2 (age score is 1).

5. Click on the **Modeling** tab to return to your decision

2.2.3.2 Exploring the current logic

The score node returns an integer that is used by the score rate node to compute a rate from the score.

1. Click on the **Score** node, then the **Details** tab



It returns an integer computed by adding all the values returned by its defined rules and decision tables.

2. Click on the **Logic** tab to explore the business rules logic.

The screenshot shows the Logic tab for a 'Score' decision node. On the left, there's a sidebar with 'Details' and 'Logic' tabs. Under 'Logic', there's a dropdown menu set to 'Sum all values'. Below it is a search bar and a list of four items: 'revenue score', 'size score', 'defaulted score', and 'age score'. Each item has a small icon and a dropdown arrow.

The logic contains one rule (defaulted score) and 3 decision tables (revenue score, size score, age score).

3. Click on **defaulted score** to explore the logic.

The defaulted score rule adds -1 to the decision in case of a defaulted payment (defaultedPayment).

This screenshot shows the Logic tab for the 'Score' decision node, specifically focusing on the 'defaulted score' rule. The sidebar shows 'Details' and 'Logic' tabs, with 'Logic' selected. A dropdown menu is set to 'Sum all values'. The search bar is empty. The list of items includes 'revenue score', 'size score', 'defaulted score' (which is highlighted), and 'age score'. The 'defaulted score' row has a dropdown arrow. To the right, a modal window titled 'defaulted score' displays the rule: 'If Client is Defaulted Payment then add -1 to decision ;'.

4. Click on the **revenue score** decision table

The revenue score decision table computes a score from the annual revenue.

This screenshot shows the 'revenue score' decision table. The sidebar shows 'Details' and 'Logic' tabs, with 'Logic' selected. A dropdown menu is set to 'Sum all values'. The search bar is empty. The list of items includes 'revenue score' (highlighted), 'size score', 'defaulted score', and 'age score'. The 'revenue score' row has a dropdown arrow. To the right, a table titled 'Annual Revenue' shows the mapping between annual revenue ranges and scores:

	min	max	Score
1	0	1,000,000	0
2	1,000,000	50,000,000	1
3	≥ 50,000,000		2
4			
5			
6			
7			
8			

5. Click on the **size score** decision table

The size score decision table computes a score from the number of employees.

This screenshot shows the 'size score' decision table. The sidebar shows 'Details' and 'Logic' tabs, with 'Logic' selected. A dropdown menu is set to 'Sum all values'. The search bar is empty. The list of items includes 'revenue score', 'size score' (highlighted), 'defaulted score', and 'age score'. The 'size score' row has a dropdown arrow. To the right, a table titled 'Number of Employees' shows the mapping between employee counts and scores:

	min	max	Score
1	100	3,000	0
2	3,000	8,000	1
3	≥ 8,000		2
4			
5			
6			
7			
8			

6. Click on **age score** decision table

The age score decision table computes a score from the company age.

Company Age		Score
min	max	
1	2	0
2	15	1
3	≥ 30	2
4		0
5		0
6		0
7		0
8		0

The score output is used by the score rate node to compute a rate from the score.

7. Click on **Back to the diagram** to return to the decision diagram

8. Click on the **Score rate** node, then **Logic** tab

9. Open the **compute rate** decision table

You see the score value varies between -1 and 7.

Score	Score rate
≤ -1	1.7
0	1.4
1	1.2
2	1
3	0.95
4	0.9
5	0.8
≥ 6	0.7

In the next step, you change the decision logic.

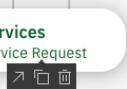
10. Click on **Back to the diagram**

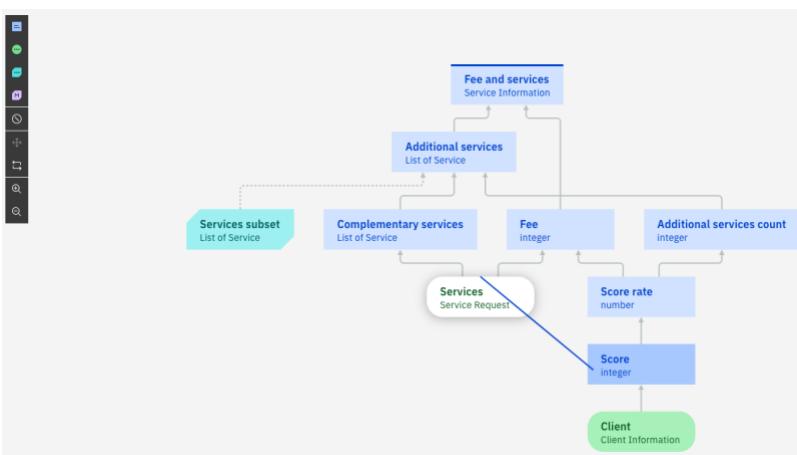
2.2.3.3 Adding a business rule

The score of the client is calculated based on 3 decision tables: revenue score, size score and age score and 1 rule: defaulted score. The score should also be evaluated based on the number of services the

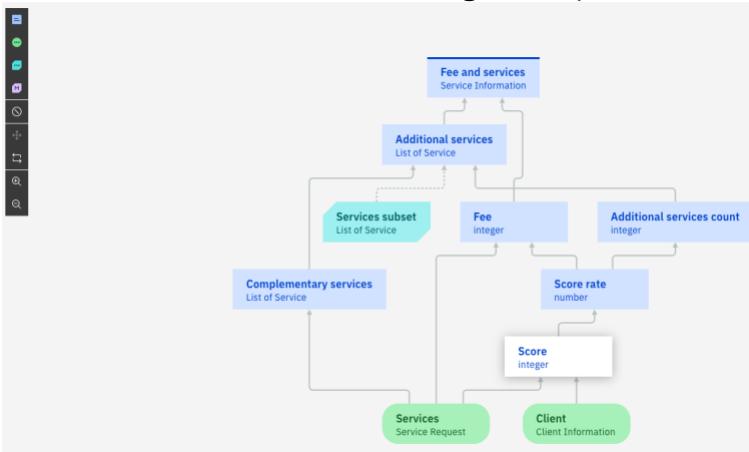
customer requested to be onboarded. You modify the business rule that calculates the score to add the number of services to its logic. A higher number of services results in a higher score.

To add the number of services as an input to the score computation you update the decision node **Score** to add **Services** as input and add a new rule to determine the score.

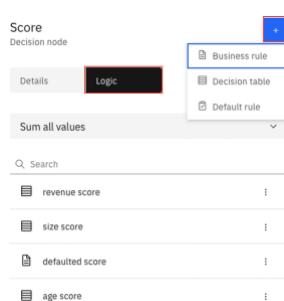
1. Click on **Services** input node, then hover over the icon (arrow) , click on **Connect to another node** to add a link to connect to another node. Drag the link to the **Score** decision.



It creates a link, and the decision diagram is updated to reflect the new dependency:



2. Click on **Score** decision node, then select the **Logic** tab in the left pane, and click on  button, **Business rule**



It opens a wizard to help you select the criteria of your rule.

3. Enter **services score** as the name for the rule, scroll down to select the criteria **the Services requested of 'Services'**.

4. Click on **Create** and update your rule. You see a red icon that indicates an error. You can hover over the icon to see the details.



5. In rule editor, update the rule. Replace the Services requested of Services contain <a Service> by the number of elements in the Services requested is more than 3 and add 1 to the decision.

```
if
  the number of elements in the Services requested of Services is more than 3
then
  add 1 to decision ;
```

While you are typing the auto-completion guides you writing your rule. You can trigger the auto-completion by typing Ctrl-Space in the editor.

Once finished the rule is as below:

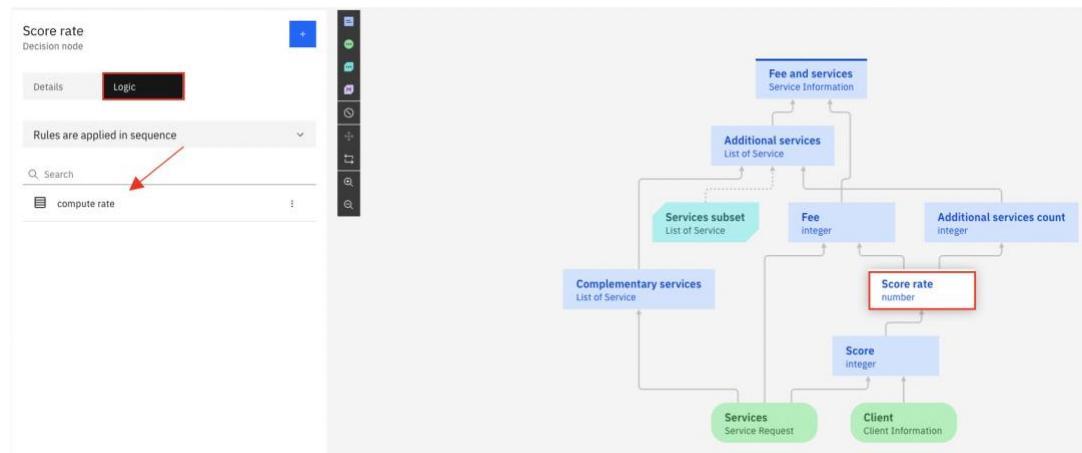
In the decision logic, the **Score** node is now the sum of 5 rules: revenue score, size score, defaulted score, age score and services score.

6. Return to the Decision model by clicking on [Back to the diagram](#) above the rule.

2.2.3.4 Updating a decision table

You update the decision table for the score rate because the values change following the update of the rule to compute the score. If needed, return to the diagram by clicking on **Modeling** tab.

- Click on **Score rate** node. In the **Logic** tab, click on compute rate.



It displays the Decision table that contains the decisions to determine the score rate.

	Score	Score rate
1	≤ -1	1.7
2	0	1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	≥ 6	0.7
9		
10		
11		

In this table, each row represents a single decision where the Score column represents a condition parameter and the Score rate the value for the action.

- Hover over the **1** in the first row to display the text as shown below:

	Score	Score rate
1	≤ -1	1.7
	<pre> if all of the following conditions are true : - ('Score' is at most '-1'), then set 'decision' to 1.7; </pre>	
2	0	1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	≥ 6	0.7
9		
10		
11		

3. Select the **row 8**, right click to open the contextual menu. Select **Insert row -> Above**. Enter 6 in Score and 0.7 in Score rate.

	Score	Score rate
1	-1	1.7
2	0	1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	6	0.7
9	≥ 6	0.7
10	Errors Line 8 overlaps with line(s) 9	
11		
12		

The warning icon appears in lines 8 and 9 to outline an error. The error indicates an overlap of row 8 and 9. You have to change the values in row 9.

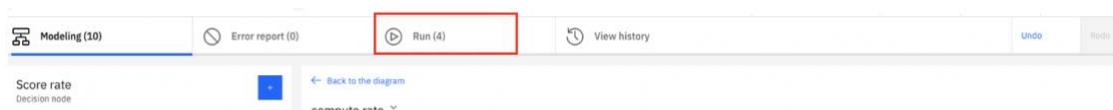
4. Change to 7 for Score and 0.6 for **Score rate**.

	Score	Score rate
1	-1	1.7
2	0	1.4
3	1	1.2
4	2	1
5	3	0.95
6	4	0.9
7	5	0.8
8	6	0.7
9	≥ 7	0.6
10		
11		
12		

2.2.4 Validating the decision service

Now that you have modified the decision model and updated the decision logic, the decision service should be tested before deploying to production. Automation Decision Services integrates a test environment allowing you to fine tune your rules and verify their behavior.

1. Click on **Back to the diagram** to return to your decision service. Select the **Run** tab.



On the left side of the screen, the **Test data** pane allows you to select the set of data to submit to execute the rules.

2. As you explore the behavior after the changes, you select the same data set as you did before the changes: **All Services** and click on **Run**.

Test data
Run your model with test data

All Services

Client
Annual Revenue
27500000
Company Age
25
Defaulted Payment

Run

The decision output for servicesFee is now 47500.

Decision output	
Node Name	Result
Fee and services	{ "extendedServices": [], "servicesFee": 47500 }

3. Expand **Run history**, then expand Score node.

It shows that the rule you added **services score** has been triggered.

4. The output displays the result for the decision node **Score**, it is a consolidated result. It is the sum of the rules that triggered the result, applied in sequence.

Each displayed output is the sum, including the previous output.

Run history			
Node	Rules	Rule Interaction	Output
Fee and services	1	Sequence	{ "extendedServices": [], "servicesFee": 47500 }
Fee	2	Sequence	47500
Additional services	1	Sequence	[]
Additional services count	1	Sequence	2
Score rate	1	Sequence	0.95
Complementary services	2	Sequence	[]
Score	4	Sum	3

Triggered rules

Triggered rules	Number of runs	Output
revenue score	1	1
size score	1	1
age score	1	2
services score	1	3

In this example, Score decision node output is 3:

- revenue score output is 1
- then added to size score (size score is 0) it remains 1,
- then added to age score result is 2 (age score is 1)
- then added to services score it is 3 (services score is 1).

For more information, see IBM Documentation

[Modeling decisions](#)

[Creating decision model](#)

2.3 Summary

You have completed the Exercise 1 - Modeling Decisions. You created a project and a decision service to define the fee and services for a client being onboarded to a set of services.

- You explored the decision model diagram and its elements.
- You updated the business logic.
- You created a rule.
- You added a node in the decision diagram.
- You edited a decision table.
- You tested and validated the decision service.

3 Exercise 2: Adding machine learning in the decision model

3.1 Introduction

In this exercise you leverage a machine learning model to improve the quality of the decision this model can evaluate. This model can evaluate a risk based on information provided during the request. Adding a predictive model allows you to benefit from the previous customer's experience to estimate the risk level.

This model provides a prediction for the client risk, based on the client information (annual revenue, company age, number of employees, and industry). If the client onboarding is risky, the prediction returns 1. If it is not, it returns a value of 0. It also provides the prediction probability.

3.2 Exercise instructions

In this exercise, to use the machine learning model deployment into Automation Decision Services you:

- Define a **provider** in the project to access the machine learning deployment
- Create a **predictive model** connected to the machine learning deployment
- Use this **predictive model** into a **decision service**
- Validate a decision service

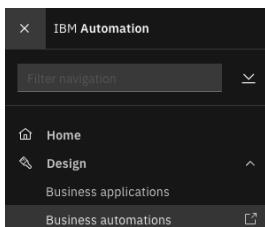
3.2.1 Log in to your project

If you are already connected to Automation Decision Services and have your project opened you can skip this section and pursue in next section [Testing the model before changes](#).

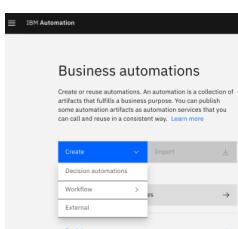
If you begin the Lab start with this exercise, follow the steps below.

3.2.2 Creating a project and importing a Decision service

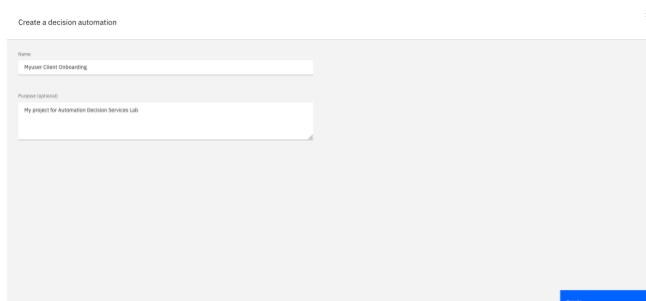
1. Log in Business Automation Studio
2. Click on the Navigation Menu on the top left corner .
3. Expand **Design**, and click Business automations



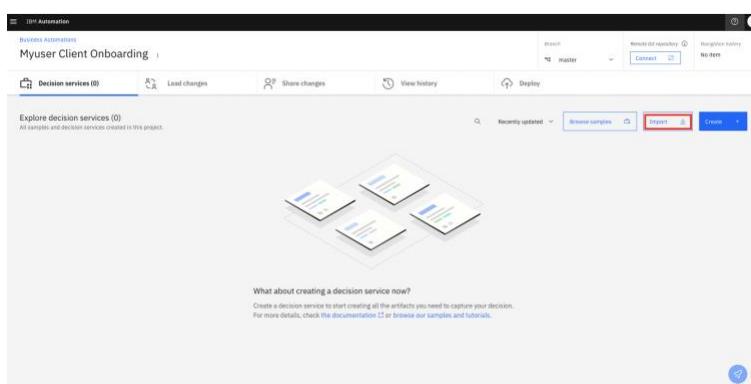
4. Click on Create and select Decision automations



- Enter a name for your project. Enter **UsrNNN Client Onboarding Decision** where *UserNNN* is your assigned username (a name starting by your username to avoid conflicts with other projects).
- Click on **Create**



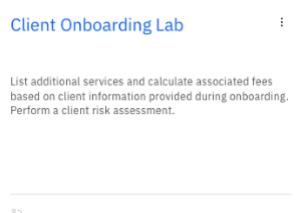
- Your new project opens in Decision Designer. Click on **Import** to import the Decision Service provided by the Lab team.



- Browse to select the project prepared for the Lab **ClientOnboardingLab.zip** and click on **Import**.



- A tile of a decision service named **Client Onboarding Lab** appears on the project.



- Click on **Client Onboarding Lab** to open your decision service.

3.2.3 Testing the model before changes

The Client Onboarding Lab decision service contains several decision models. In this exercise, you work on the Scoreboard decision model. It uses one data model: Data.

If you are not familiar with data model, you can check [Exercise 1 Exploring the decision service: data model and decision model](#).

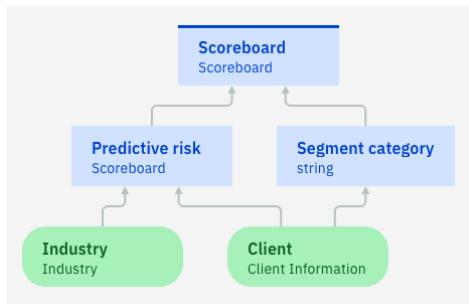
1. In the Decision models tab, click on Scoreboard.

The screenshot shows the 'Decision models (3)' tab selected in the top navigation bar. Below it, a table lists four decision models:

Name	Last updated
Fee and services	Me Never shared
Scoreboard	Me Never shared
Services subset	Me Never shared

2. It opens the Decision model **Scoreboard**.

The decision model diagram is a dependency diagram that shows the steps to make the final decision. The top decision **Scoreboard** is the final decision. It calculates a client risk and categorizes the client in a business segment. The segment is computed by the decision node **Segment category** from the input node **Client**. The risk is computed from the **Predictive risk** node with the input nodes **Industry** and **Client information**. This model is not complete, currently the Predictive risk node has no business logic defined, you will complete it in this exercise.



3. Select the **Run** tab to test the model

The screenshot shows the 'Run' tab selected in the top navigation bar of the Scoreboard model. It displays 4 runs.

On the left side of the screen, the **Test data** pane allows to select the set of data to submit to execute the rules. In this Lab, 4 sets of data are pre-defined: May be risky, Not so risky, Risky, Safe.

4. Select one data set of your choice and click on **Run** to test your model. You can test another data set.

The screenshot shows the 'Test data' pane. A dropdown menu is open, listing four categories: 'May be risky', 'Not so risky', 'Risky', and 'Safe'. The 'May be risky' option is selected and highlighted with a red box.

It generates a report and displays the input and output of each node in the decision model.

5. Expand **Decision output** and **Run history**

The screenshot shows the 'Test data' section on the left with various client details like Annual Revenue, Company Age, and Industry. The 'Decision output' and 'Run history' sections on the right are highlighted with red boxes. In 'Decision output', there's a table with one row for 'Scoreboard' where the result is 'null'. In 'Run history', there are four rows: 'Scoreboard' (Rules: 0, Rule Interaction: Sequence, Output: null), 'Predictive risk' (Rules: 0, Rule Interaction: Sequence, Output: null), 'Segment category' (Rules: 1, Rule Interaction: Sequence, Output: "Segment 2"), and 'Industry' (Rules: 0, Rule Interaction: Not applicable, Output: null).

The final decision is displayed in the top of the report: for Scoreboard the result is "null".

The decision result provides the segment 2 category based on the Annual revenue specified in the data set. In this exercise, if the revenue is below 50000000 it categorizes the client in segment 2, if the revenue is above the client is set in segment 1. Here the risk is not yet computed, you need to add the logic of the decision node Predictive risk.

3.2.4 Defining a Machine learning provider

A machine learning provider gives access to Machine learning deployment. Two types of providers are currently supported in Automation Decision Services: IBM Watson® Machine Learning and IBM Open Prediction Service.

In this exercise, the Machine learning model has been deployed on an instance of IBM Open Prediction Services (OPS).

This machine learning model has been defined using a Jupyter notebook and saved into a Predictive Model Markup Language file (PMML). This PMML file has been used to deploy the machine learning model into IBM Open Prediction Services (OPS).

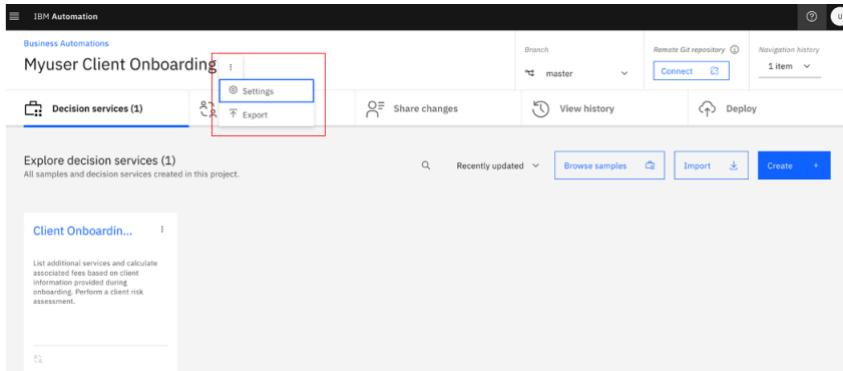
For more information, see IBM Documentation [Managing machine learning providers](#)

For more information about ADS ML Service, see the Open Prediction Service Hub repository [available on GitHub](#).

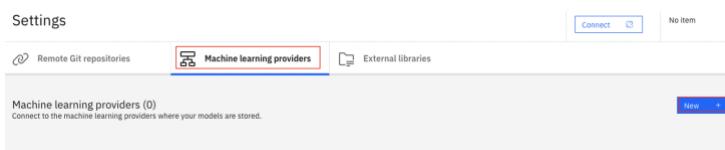
1. Click on your project Client Onboarding in the breadcrumbs to navigate back to your project.



2. Open the **Settings** menu from your project, click on the three dots menu ⋮ next to the name of the project and click on **Settings**



3. Open the **Machine learning providers** tab and click on **New**



4. Select the provider type **Open Prediction Service**



5. Enter **OPS** in the **name** field, and optionally a description

6. Enter the URL of the Open Prediction Service Instance. The URL you need is the URL of ADS ML Service (see [Prerequisites](#) if you need the URL)

Ensure it has no *docs* at the end of the url. For example: `http://ads-ml-service-service-ads-ml-service.cp4ba-jam-americas-464887bc828751e1b00625ca9211fbca-0000.us-south.containers.appdomain.cloud`

Name	OPS
Description	(empty)
URL	<code>http://ads-ml-service-service-ads-ml-service.swat-dev-01-464887bc828751e1b00625ca9211fbca-0000.eu-de.containers.appdom</code>
Test connection	
Cancel	Save

- Click on **Test connection** button to verify you access OPS provider. Upon successful connection click on **Save**



- In the **Machine learning providers** tab, you have now the OPS provider you defined.

For more information, see IBM Documentation

[Integrating machine learning](#)

[Managing machine learning providers](#)

- To go back to your decision service, click on your project name in breadcrumb.

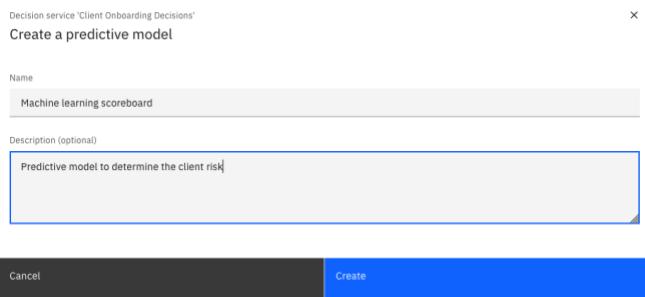
3.2.5 Creating and connecting the Predictive Model

Now that the provider is defined, you can create a predictive model to encapsulate the machine learning model deployment. You connect it to the appropriate machine learning model deployment and define the input and the output of this predictive model.

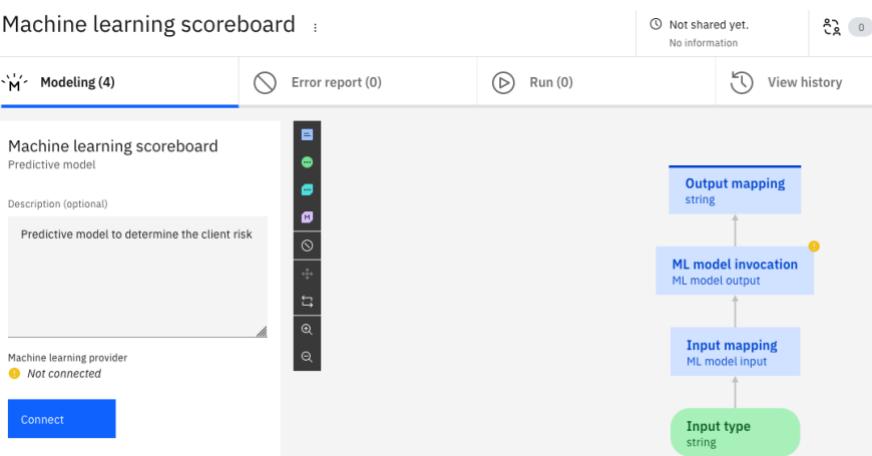
- Click on the tile **Client Onboarding Lab** to open the decision service

- Click on the **Predictive models** tab and click on **Create**.

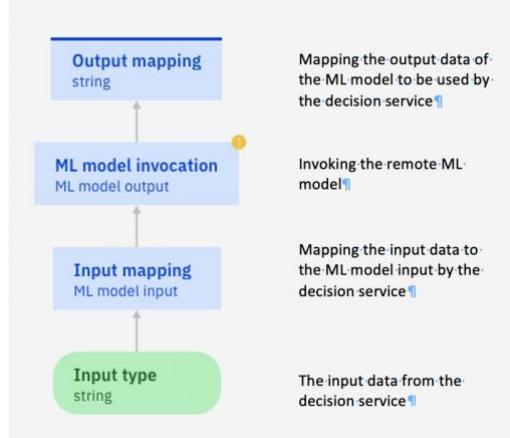
3. In the creation wizard, enter a name for your model (Machine learning scoreboard), a Description optionally and click on **Create**



4. The Predictive model **Scoreboard** is opened.



A predictive model is a simple Decision Model diagram. It has one or multiple input nodes, two mapping nodes and one node that invokes the Machine Learning model.



5. In the left-hand pane click on **Connect**.

6. Expand **Select a provider**, select OPS.

Configure predictive model

Select provider and ML deployment

Select provider
Select the provider where your deployed model is stored.

Machine learning provider

Select a provider OPS

New provider

7. In **ML model name** list, expand service-payment-default-risk.

The list displays all the machine learning models deployed in the environment. Expand the model with the most recent Last modified date.

ML model name	Training date	Last modified
<input checked="" type="checkbox"/> service-payment-default-risk	5/12/2021, 1:58:38 PM	5/12/2021, 1:58:38 PM
Deployment name	Status	Deployment date
<input type="radio"/> service-payment-default-risk	in_service	5/12/2021, 2:04:07 PM

8. Choose service-payment-default-risk, click on **Next**.

Back to Machine Learning scoreboard
Configure predictive model

Select provider and ML deployment

Define input schema
Optional

Test invocation
Optional

Define output schema
Optional

Select provider
Select the provider where your deployed model is stored.

Machine learning provider

OPS New provider

Select machine learning model deployment
Select the deployment you want to use to generate the predictive model.

Show deployed models only

ML model name	Training date	Last modified
<input checked="" type="checkbox"/> service-payment-default-risk	5/12/2021, 1:58:38 PM	5/12/2021, 1:58:38 PM
Deployment name	Status	Deployment date
<input checked="" type="radio"/> service-payment-default-risk	in_service	5/12/2021, 2:04:07 PM

9. Look at the **input schema**: you have to provide the values to the machine learning model from the input nodes defined in your predictive model. Click on **Next**.

Back to Machine learning scoreboard
Configure predictive model

Select provider and ML deployment

Define input schema
Optional

Test invocation
Optional

Define output schema
Optional

Define input schema
Define the input parameters needed to make the prediction.

Name	Type	↑ ↓ ↻
clientAnnualRevenue	float64	↑ ↓ ↻
clientExistenceDuration	float64	↑ ↓ ↻
clientEmployeeNumber	float64	↑ ↓ ↻
clientIndustry	float64	↑ ↓ ↻

Form JSON Generate from payload Add

10. Enter some values to test in the **Test invocation**, then click on **Run**.

clientAnnualRevenue 15708854
clientExistenceDuration 12
clientEmployeeNumber 3
clientIndustry 0

Select provider and ML deployment Define input schema Test invocation Define output schema

Test invocation
Use test data to make sure the model works as expected.

clientAnnualRevenue
15708854

clientExistenceDuration
12

clientEmployeeNumber
3

clientIndustry
0

Output
Click Run to invoke the ML model.

Run

11. Click on **Run**

Back to Machine learning scoreboard Configure predictive model

Select provider and ML deployment Define input schema Test invocation Define output schema

Test invocation
Use test data to make sure the model works as expected.

clientAnnualRevenue
15708854

clientExistenceDuration
12

clientEmployeeNumber
3

clientIndustry
0

Output
{ "result": { "predictions": 1, "scores": [0.014675209287711932, 0.9853247907122881] } }

Back Next

Run

The output returns: it is a risky client as the prediction is 1

```
{  
  "result": {  
    "predictions": 1,  
    "scores": [  
      0.014675209287711932,  
      0.9853247907122881  
    ]  
  }  
}
```

12. Look at the **output schema**: you have to map it to the scoreboard type in your predictive model.
Click on **Next**.

Back to Machine learning scoreboard Configure predictive model

Select provider and ML deployment Define input schema Test invocation Define output schema

Test invocation
Use test data to make sure the model works as expected.

clientAnnualRevenue
15708854

clientExistenceDuration
12

clientEmployeeNumber
3

clientIndustry
0

Output
{ "result": { "predictions": 1, "scores": [0.014675209287711932, 0.9853247907122881] } }

Back Next

Run

13. Select **Generate from test output**.

The screenshot shows the 'Configure predictive model' interface. In the 'Define output schema' section, there are four tabs: 'Form', 'JSON', 'Generate from test output' (which is highlighted in red), and 'Generate from payload'. Below these tabs, there is a table with columns 'Name' and 'Type'. Under 'Name', there is a row with 'attributes'. Under 'Type', it is set to 'object'. At the bottom of the 'Generate from test output' dialog, there is a 'Payload' section containing JSON code and a 'Generated schema' section showing the generated schema. At the bottom right of the dialog, there are 'Cancel' and 'OK' buttons, with 'OK' being highlighted in blue.

14. Click **OK**

15. Finally click on **Apply** to finish the configuration of your predictive model.

The screenshot shows the 'Configure predictive model' interface. In the 'Define output schema' section, there are four tabs: 'Form', 'JSON', 'Generate from test output', and 'Generate from payload'. Below these tabs, there is a table with columns 'Name' and 'Type'. Under 'Name', there are two rows: 'predictions' (Type: double) and 'scores' (Type: [double]). At the top right of the interface, there is an 'Apply' button highlighted in blue.

3.2.6 Defining the mapping

You map the data types of the input and output of the machine learning model to the data types of the input and output of the Decision Service data model.

3.2.6.1 Defining the input mapping

In 3.2.5.8 you have seen the input mapping of the ML model, it requires:

- clientAnnualRevenue as a number
- clientExistenceDuration as a number
- clientEmployeeNumber as a number
- clientIndustry as a number

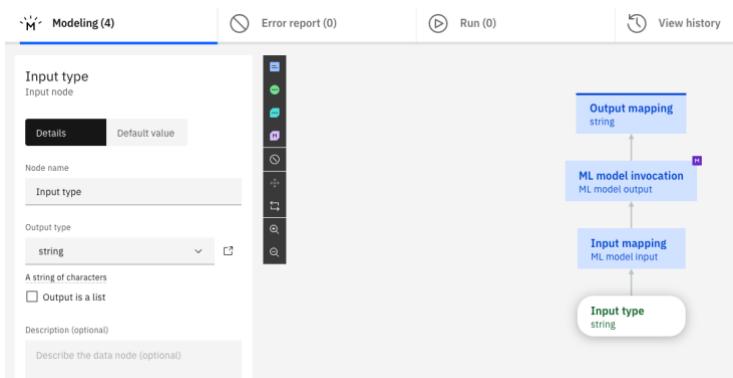
You get the three first values from the Client information type. The last one is taken from Industry enumeration type. You map each value to a number. You have to:

- define an input node for the Industry
- add an input node for the Client Information
- define the rules to map those input nodes to the values expected by the ML model. For the Industry, you need a decision table to map the enumeration type to an integer.

1. Go back to **Machine Learning scoreboard** model.

The predictive model Machine learning model is based on the industry and the client information as input. You update the model to map your data to the required input.

2. Select the **Input type** node.



3. In the left pane **Details** edit the Node name and Output type. Enter Industry as the **Node name** and expand **Output type** to select Industry as the type.

Industry
Input node

Details Default value

Node name
Industry

Output type
Industry

Federal, Finance, Healthcare, Insurance,...

Output is a list

Description (optional)

Describe the data node (optional)

Machine learning scoreboard

Modeling (4) Error report (0) Run (0) View history

Machine learning scoreboard
Predictive model

Description (optional)
Predictive model to determine the client risk

Machine learning provider
OP5

Machine learning model
service-payment-default-risk

Deployment
service-payment-default-risk

Edit Configuration

```

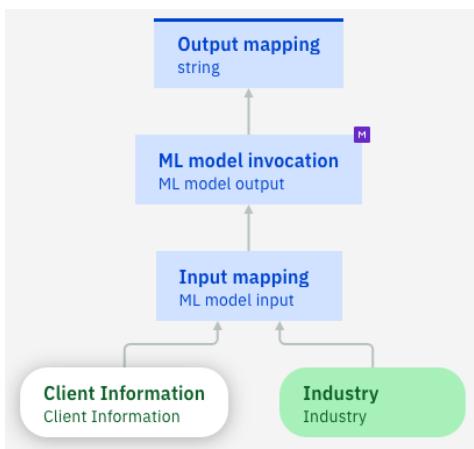
graph TD
    Industry((Industry)) --> InputMapping[Input mapping<br/>ML model input]
    InputMapping --> ModelInvocation[ML model invocation<br/>ML model output]
    ModelInvocation --> OutputMapping[Output mapping<br/>string]
  
```

You add a node in the model for Client information.

- Select Industry node and click on the **Copy node** icon to duplicate the node.



- Select the Copy of Industry node and update the **Node name** to **Client Information** and Output type list to select **Client Information**.



You have added the Input nodes. Now, you map the input with the Decision Service data model.

- In the Predictive model diagram, select the **Input mapping** node, click on the **Logic** tab and select the + sign to add a **Business rule**

Input mapping
Decision node

Details Logic

No decision logic defined yet
Create rules and decision tables to define how your decision is made.
For more details, check the documentation

7. In the wizard, notice the **preview** of the generated rule. It matches the input of the Machine Learning Service (Industry, Client Information). Enter a rule name **set client input** in the Name field and click on **Create**.

8. Edit the rule with the following statement

```
set decision to a new ML model input where
    the clientannualrevenue is the Annual Revenue of 'Client Information',
    the clientemployeenumber is the Number of Employees of 'Client Information',
    the clientexistenceduration is the Company Age of 'Client Information' ,
    the clientindustry is 0 ;
```

The rule above always sets the client industry to 0. As next step, you add a new Decision table to update the business logic by defining the industry from the industry provided as the input.

9. In the **Logic** tab, select the **+sign** and select **Decision table**.

10. Enter a name **set industry** in the Name field. Select Industry in the **condition columns** list. Click on **Create**.

	Industry	Input mapping
1		
2		
3		

11. Right-click on the **Input mapping column to update the condition, select **Define column**.**

You need to update the column to set the industry only. The other attributes were set in earlier step by the rule set client input.

A screenshot of a table titled 'set industry'. The columns are 'Industry' and 'Input mapping'. The 'Input mapping' column has a context menu open with options: 'Define column...', 'Format column...', 'Cut', 'Copy', 'Paste', 'Insert column', 'Delete', and 'Clear'. The 'clientannualrevenue' row is selected in the main table.

	Industry	Input mapping
1		clientannualrevenue
2		
3		
4		
5		
6		
7		
8		
9		
10		

12. Update the rule in the Editor and replace with the following rule statement and click the **Update button in the Update Action Column dialog.**

```
set the clientindustry of decision to <a number>
```

A screenshot of the 'Update Action Column' dialog. It contains a text area with the rule statement: 'set the clientindustry of decision to <a number>'. Below the dialog is the 'set industry' table, which is currently empty.

	Industry	Input mapping
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

13. Double click in **Row 1 and add select **Federal** in the drop-down list for the Industry. Add **0** for Input mapping.**

14. Pursue for Row 2 to 5. Enter the following values:

A screenshot of the 'set industry' table. The data is as follows:

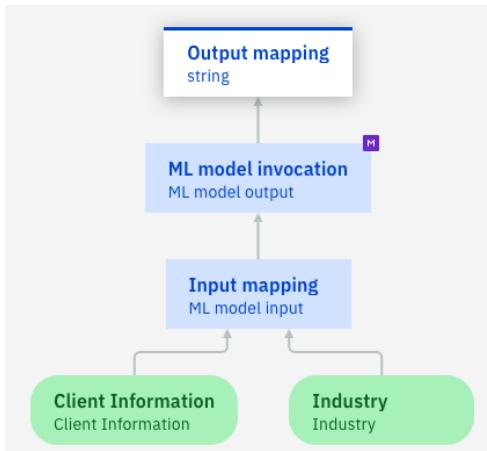
	Industry	Input mapping
1	Federal	0
2	Finance	1
3	Healthcare	2
4	Insurance	3
5	Telecom	4
6		
7		
8		
9		
10		

3.2.6.2 Defining the output mapping

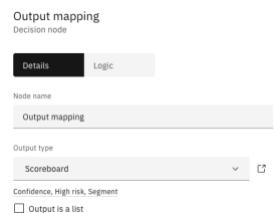
In 3.2.5.11 you have seen the output of the machine learning model. You have to map it to the output of your predictive model. Your predictive model will return a scoreboard in which it predicts the score and gives the probability. To do this you:

- change the output type of the Output mapping node of the predictive model to scoreboard
- define the rule to build this scoreboard:
 - write a default rule for the case of the machine learning model in error. This rule has to be the only one to be executed in case of an error. You use the rule policy First rule applies and ensure that this rule is on the top.
 - write a rule to map the output of the machine learning model to the scoreboard. The segment remains Unknown since it is not computed by the machine learning model.

1. Click on [← Back to the diagram](#) to return to the Predictive model diagram.

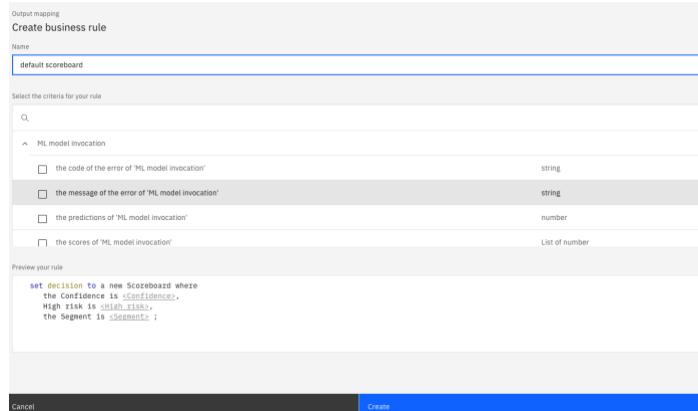


2. In the Predictive model diagram, select the **Output mapping** node, in **Details** tab, Expand **Output type** and select **Scoreboard** in the list.



You define the business logic by adding two rules to feed the output type from the machine learning model invocation. First, a rule to define an output when the machine learning model is in error (**default scoreboard**). Second, a rule to map the machine learning model output (**get machine learning output**). The two rules should not be applied in sequence, only the first rule applies if there is an error in the machine learning model.

3. Select the **Logic** tab and select the + sign to add a **Business rule**.
4. Enter a name for the rule: **default scoreboard**. Click on **Create** to edit your rule.



5. In the editor, a rule is predefined. An icon indicates an error as the rule is incomplete. Hover over the error icon in line 1 to see the details:

default scoreboard ▾
Type your rule using the list below as reference

```
1 set decision to a new Scoreboard where
2   confidence,
3   • The rule is incomplete,
4   fill all the placeholders. ;
```

6. Edit the **rule** statement as below

```
if 'ML model invocation' is in error
then
set decision to a new Scoreboard where
    the Confidence is 0,
    High risk is true,
    the Segment is "Unknown" ;
```

default scoreboard ▾

Type your rule using the list below as reference

```
1 if 'ML model invocation' is in error
2 then
3 set decision to a new Scoreboard where
4   the Confidence is 0,
5   High risk is true,
6   the Segment is "Unknown" ;
```

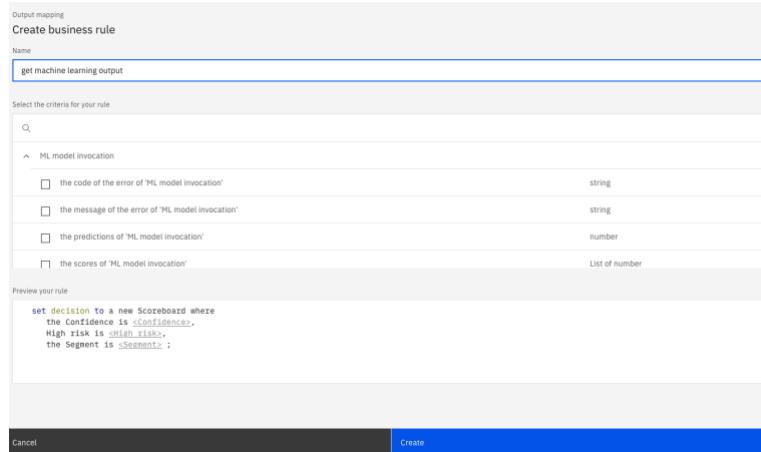
7. Click on **Inputs** and **Output** below the rule to see the data types of the input and output.

Inputs (1)	Output (1)
^ ML model invocation	ML model output
error	Output error
predictions	number
score	number

Inputs (1)	Output (1)
Output mapping	Scoreboard
Confidence	number
High risk	boolean
Segment	string

Now, you add the second rule.

8. Click on the **Logic** tab, select the + sign to add a **Business rule**.
9. Enter a name for the rule: **get machine learning output**. Click on **Create** to edit your rule.



The rule is incomplete as you can see from the icon in line 1.

10. Edit the rule with the following statement:

```
definitions
set percent to the maximum score of 'ML model invocation';
if the predictions of 'ML model invocation' is 1
then
set decision to a new Scoreboard where
    High risk is true ,
    the Confidence is percent,
    the Segment is "Unknown";
else
set decision to a new Scoreboard where
    High risk is false ,
    the Confidence is percent,
    the Segment is "Unknown";
```

[← Back to the diagram](#)

get machine learning output

Type your rule using the list below as reference

```
1 definitions
2 set percent to the maximum score of 'ML model invocation';
3
4 if the predictions of 'ML model invocation' is 1
5 then
6 set decision to a new Scoreboard where
7     High risk is true ,
8     the Confidence is percent,
9     the Segment is "Unknown";
10 else
11 set decision to a new Scoreboard where
12     High risk is false ,
13     the Confidence is percent,
14     the Segment is "Unknown";
15
```

11. In the **Logic** tab in the left pane you can see the **Rules are applied in sequence**, first the default scoreboard, then get machine learning output.

The screenshot shows the Logic tab of a decision node configuration. At the top, it says "Output mapping" and "Decision node". Below that is a navigation bar with "Details" and "Logic" tabs, where "Logic" is selected. A dropdown menu titled "Rules are applied in sequence" is open, showing two items: "default scoreboard" and "get machine learning output".

How the rules interact with each other is defined by a **rule interaction policy**.

12. Expand **Rules are applied in sequence**. You see two policies First rule applies and Rules are applied in sequence. The choices for the rule interaction policy depend on the Decision type. Here, the decision node **Output mapping** is a custom type.

The screenshot shows the Logic tab with the "Rules are applied in sequence" dropdown expanded. It contains two options: "First rule applies" and "Rules are applied in sequence". The "First rule applies" option is highlighted with a blue border. A tooltip for "First rule applies" states: "When a rule is applicable, the decision is taken and cannot be changed by other rules." A tooltip for "Rules are applied in sequence" states: "When a rule is applicable, it can modify the decision made by the previous rule."

13. Change the rule policy to **First rule applies**

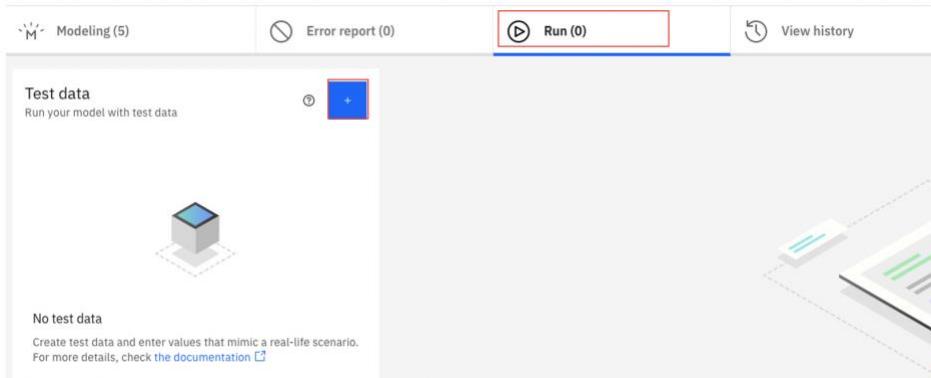
The screenshot shows the Logic tab with the "First rule applies" policy selected. The dropdown menu now only shows "First rule applies". The search bar and rule list below are identical to the previous screenshot.

For more information, see IBM Documentation [Interaction policies](#)

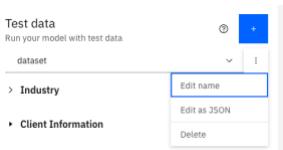
3.2.7 Validating the predictive model

Now that you have created your Predictive model and defined the predictive decision logic, you test it to verify. IBM Automation Decision Services integrates a test environment allowing you to fine tune your rules and verify their behavior online against the test data. You add datasets to define your data to test.

1. Select the **Run** tab.



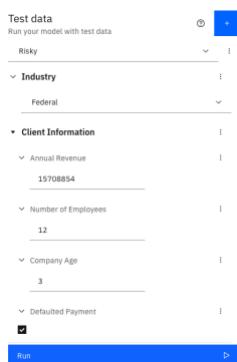
2. In the **Test data** left-side pane, click on + sign, select the 3 dots to edit and add test data.



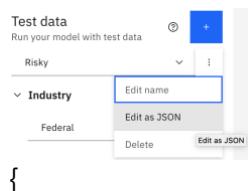
As next step, you add four data sets: Risky, May be risky, Not so Risky, Safe.
To add test data, you enter a name and define the values.

3. Select Edit name, enter **Risky**. Expand **Industry** and **Client Information** to enter the test data

Name: **Risky**
 Industry: Federal
 Annual Revenue: 15708854
 Number of Employees: 12
 Company Age: 3
 Defaulted Payment: true (checked)



4. You can see as JSON content by clicking on the 3 dots near the name and select Edit as JSON.



```

"industry": "Federal",
"clientInformation": {
  "annualRevenue": 15708854,
  "numberOfEmployees": 12,
  "companyAge": 3,
  "defaultedPayment": true
}
}

```

You can edit dataset as FORM by entering the data in the form as you did in this step or as JSON. To enter a new data set you can paste a JSON content.

5. (Optional) Repeat the steps to add 3 additional datasets. Copy/Paste the JSON content

Name: **Not so risky**

```

{
  "clientInformation": {
    "numberOfEmployees": 10,
    "annualRevenue": 61399457,
    "companyAge": 4
  },
  "industry": "Healthcare"
}

```

Name: **May be risky**

```

{
  "industry": "Federal",
  "clientInformation": {
    "annualRevenue": 15708854,
    "companyAge": 3,
    "numberOfEmployees": 12
  }
}

```

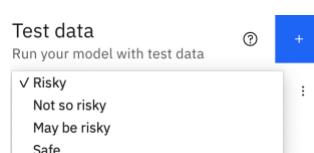
Name: **Safe**

```

{
  "industry": "Healthcare",
  "clientInformation": {
    "annualRevenue": 75314927,
    "numberOfEmployees": 67,
    "companyAge": 26
  }
}

```

You now have 4 data sets (or less depending on the number of data sets you added previously).



- Select a data set of your choice and click on **Run** in the left-pane.



- See the Decision output for the data set Risky.

Risky
5/20/2021, 1:50:13 AM

[Show JSON output](#)

Decision output

Node Name	Result
Output mapping	<pre>{ "confidence": 0.9999937559038077, "highRisk": true, "segment": "Unknown" }</pre>

- You can explore the output in JSON format, by clicking on **Show JSON output**

Risky
5/20/2021, 1:50:13 AM

[Show formatted output](#)

```
[
  "type": "test",
  "value": "Risky",
  "id": "0",
  "value": "Risky",
  "intos": [
    {
      "executedRules": [
        {
          "executionCount": 1,
          "nodeKind": "Decision"
        }
      ],
      "result": "\\"confidence\\":0.9999937559038077,\\\"highRisk\\\":true,\\\"segment\\\":\\\"Unknown\\\"",
      "ruleName": "\\ge machine learning output"
    }
  ],
  "nodeKind": "Decision mapping",
  "result": "\\"confidence\\":0.9999937559038077,\\\"highRisk\\\":true,\\\"segment\\\":\\\"Unknown\\\""
},
{
  "executedRules": [
    {
      "executionCount": 1,
      "nodeKind": "Decision"
    }
  ],
  "result": "\\"error\\":null,\\\"predictions\\":1.0,\\\"scores\\":{\\"0.24409619232047E-6\\":0.9999937559038077}",
  "ruleName": "\\maxRiskHighScoreout"
},
{
  "nodeKind": "Decision",
  "result": "\\"error\\":null,\\\"predictions\\":1.0,\\\"scores\\":{\\"0.24409619232047E-6\\":0.9999937559038077}"
},
{
  "executedRules": [
    ...
  ]
}
```

3.2.8 Use the predictive model in the scoreboard decision model

You add a prediction node to the scoreboard decision model connected to this predictive model. You connect this prediction node to the predictive risk input node and create a rule to define the risk.

3.2.8.1 Add a prediction node

- Go back to your **Scoreboard** decision model.

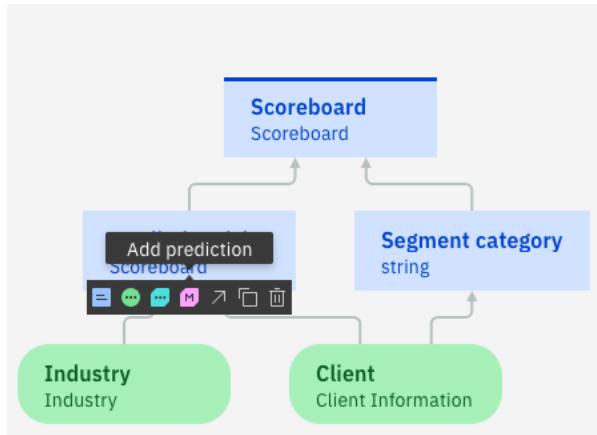
You can click in the navigation history on the top-right corner of the screen



or click on **Modeling** tab.



2. It shows the decision model Scoreboard. Hover over **Predictive risk** node and click on **Add prediction** icon.



The prediction node shows an error as it needs to be connected to a predictive model.

3. In the left-side pane **Prediction node**, select the predictive model you created. Expand the list of Predictive models and select **Machine Learning scoreboard**.

Prediction node

Predictive models

Select a predictive model ▾

Machine learning scoreboard

Machine learning scoreboard

Prediction node

Predictive models

Machine learning scoreboard ▾

Input name clientInformation Input type Client Information
industry Industry

Output name Machine learning scoreboard Output type Scoreboard

Invocation example

```
set decision to the machine learning scoreboard computed from
Client Information being <Client information>,
Industry being <Industry>;
```

3.2.8.2 Editing the logic definition of the predictive risk

You update the decision logic by adding a predictive rule. You add a rule calling the predictive model with the appropriate input, to define the risk and the confidence of the Scoreboard decision node.

1. Select the decision node **Predictive risk**, in the **Logic** tab click the **+** sign to create a Business rule.
2. Enter a name for the rule: **predictive risk**.
3. Click on **Create**.
4. Edit the rule statement as below:

```
set decision to the machine learning scoreboard computed from
Client Information being Client ,
Industry being Industry ;
```

predictive risk

Type your rule using the list below as reference

```

1 set decision to the machine learning scoreboard computed from
2   Client Information being Client ,
3   Industry being Industry ;

```

3.2.9 Validating the final decision service

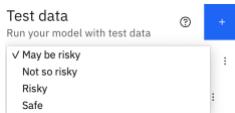
Now that you created the predictive model and updated the decision model Scoreboard to integrate the prediction to compute the risk for the client onboarding, you can validate the changes. You use the four datasets previously created to validate:

Risky, May be risky, Not so Risky, Safe

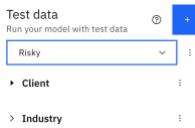
1. Select **Run** tab. In the **Test data** left-side pane.



2. In the **Test data** tab, select the dataset of your choice.



3. For this example, Risky is selected. Click on **Run** and check the results.



The run history shows that the rule **Predictive risk** ran and computed the **confidence** level 99,99% and **high risk** value (true) for the **Scoreboard** decision node.

3.3 Summary

You have completed the Exercise 2 - Adding machine learning in the decision model.

- You added a predictive service provider and connected your project to it.

- You modified the scoreboard for the client onboarding by combining descriptive rule and predictive rule. By adding predictive model to your project decision, you infused machine learning to evaluate the client risk based on a trained model.
- You added data sets to verify that your model is running correctly against the business rules policies defined for the Client Onboarding scenario.

4 Exercise 3: Sharing and publishing decision services

4.1 Introduction

In this exercise, you learn how to collaborate on your decision services and how to make it ready to be executed by the other components of the platform. You can directly collaborate in Automation Decision Services by sharing your decision service and giving appropriate access. Then you connect your project to a Git repository to be able to build and deploy a decision service as an archive. You publish this archive as an automated service in Business Automation Studio.

This exercise is dedicated to integrators and anyone who wants to understand how to execute a decision service.

Integrating the automated services in other applications is not covered in this Lab, look at the Workflow and Business Automation Application Labs for this.

For this exercise, you can either work with the project you created or with the decision services prepared by the Lab team.

4.2 Exercise instructions

In this exercise you prepare a decision service for collaboration:

- **Share changes** in a decision service to make them visible to collaborators
- Connect a decision service to a GitHub repository
- **Deploy** the decision service as **an automated service**

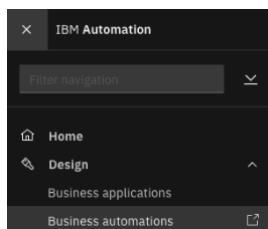
4.2.1 Log in to your project

If you did the previous exercises, and have your decision service you can pursue with your project. In Decision designer, open your project. Then move to the step [Sharing your decision service](#).

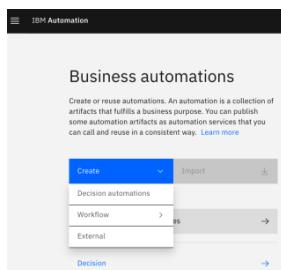
If you begin the Lab start with this exercise, follow the steps below.

4.2.2 Creating a project and importing a Decision service

1. Log in Business Automation Studio
2. Click on the Navigation Menu on the top left corner  .
3. Expand Design, and click on Business automations

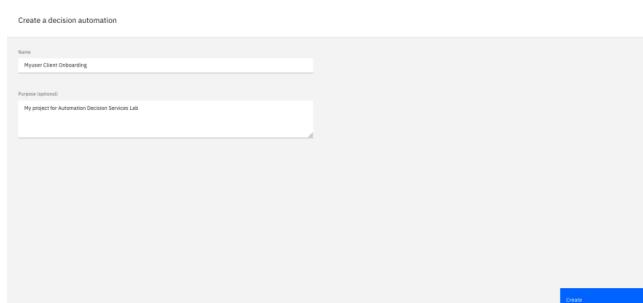


4. Click on Create and select Decision automations

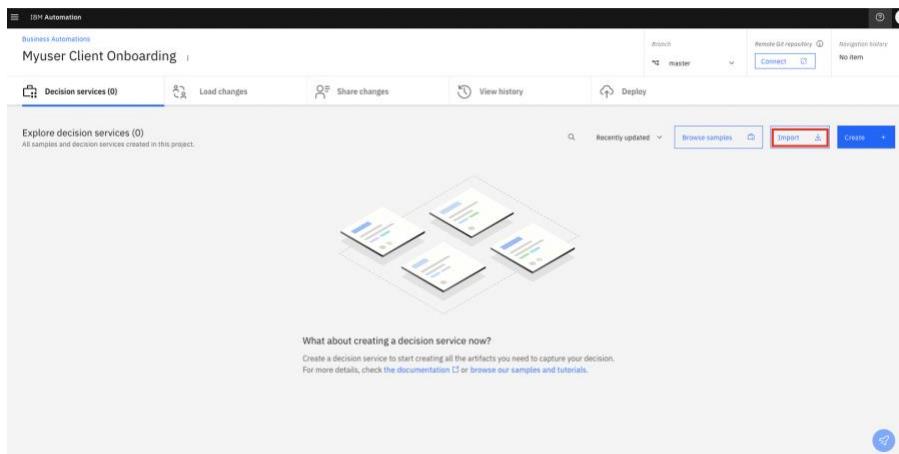


5. Enter a name for your project. Enter **UsrNNN Client Onboarding Decision** where *UserNNN* is your assigned username (a name starting by your username to avoid conflicts with other projects).

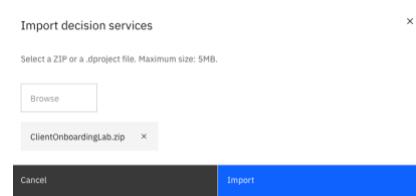
6. Click on **Create**



7. Your new project opens in Decision Designer. Click on **Import** to import the Decision Service provided by the Lab team.



8. Browse to select the project prepared for the Lab **ClientOnboardingLab.zip** and click on **Import**.



9. A tile of a decision Service named **Client Onboarding Lab** appears on the project.

[Client Onboarding Lab](#)

List additional services and calculate associated fees based on client information provided during onboarding. Perform a client risk assessment.

4.2.3 Sharing your decision service

The changes made on a decision service become visible when you share them. You give the access rights in Business Automation Studio to your collaborators.

1. Click on the **Share changes** tab



It displays the changes you made to the Decision service. It shows the number of changes. You see artifacts of your decision service Keep all selected. In your project the number of changes may be different.

Share changes (12)			
Share changes that you made locally with your collaborators.			
	Decision service name	Updated artifacts	Details
^	Client Onboarding Lab	6	Decision service added
	Name	Type	Artifact added
	Data	Data model	Artifact added
	Fee and services	Decision model	Artifact added
	Machine Learning scoreboard	Predictive model	Artifact added
	Scoreboard	Decision model	Artifact added
	Services subset	Decision model	Artifact added
^	Client Onboarding Decisions	6	Decision service added
	Name	Type	Artifact added
	Data	Data model	Artifact added
	Fee and services	Decision model	Artifact added
	Machine Learning scoreboard	Predictive model	Artifact added
	Scoreboard	Decision model	Artifact added
	Services subset	Decision model	Artifact added
	Client Onboarding Decisions	6	Decision service added
	Name	Type	Artifact added
	Data	Data model	Artifact added
	Fee and services	Decision model	Artifact added
	Machine Learning scoreboard	Predictive model	Artifact added
	Scoreboard	Decision model	Artifact added
	Services subset	Decision model	Artifact added

2. Click on **Share**

3. Enter a comment to describe the changes i.e My user first Client Onboarding version then click on **Share**

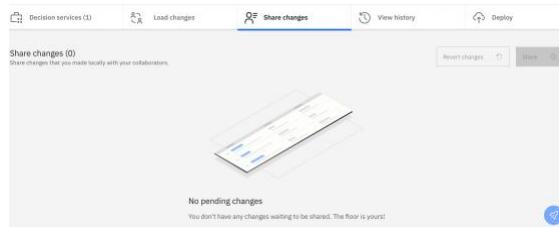
A screenshot of the 'Share' dialog box. It contains a table showing the updated artifacts from the previous screenshot, followed by a text area for a comment, and finally a 'Share' button at the bottom.

Name	Type
Data	Data model
Fee and services	Decision model
Machine Learning scoreboard	Predictive model
Scoreboard	Decision model
Services subset	Decision model
Client Onboarding Decisions	
Name	Type
Data	Data model
Fee and services	Decision model
Machine Learning scoreboard	Predictive model
Scoreboard	Decision model
Services subset	Decision model

Describe the changes (optional)
My user first Client Onboarding version

Cancel Share

- As you see there are no pending changes now

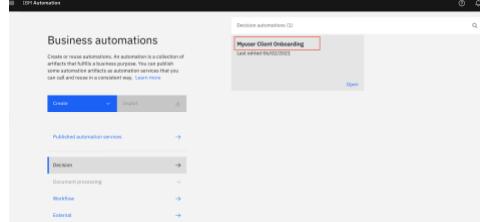


Sharing changes means the updates done locally are published and visible to other users provided that you give them access.

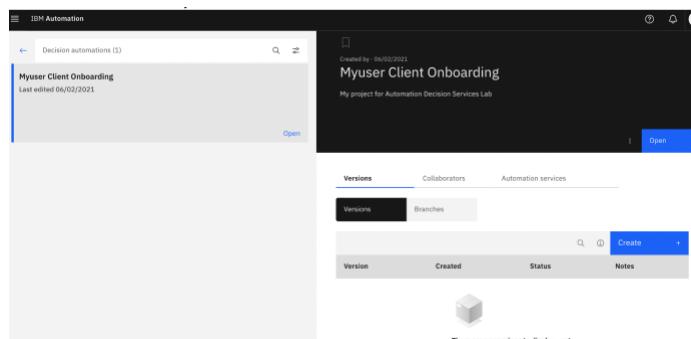
In this Lab, adding other users is not covered, however below is the procedure you would follow.

To share a project with other users, you go to **Business automations** from the upper left menu (IBM Automation, Expand Design, Select Business automations, then select Decision).

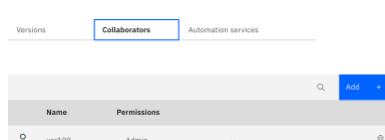
- In Business automations you select your project by clicking on your project name.



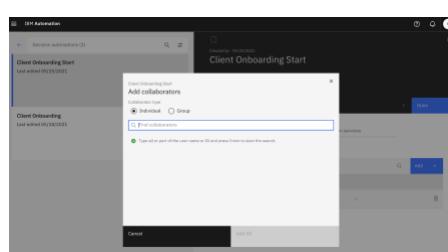
- It opens.



- Then, you select the **Collaborators** tab and click **Add +**. Add the name of the users you want to add and select the permissions (Admin, Edit or Read).



- Add the name of the users you want to add and select the permissions (Admin, Edit or Read).



4.2.4 Creating a version

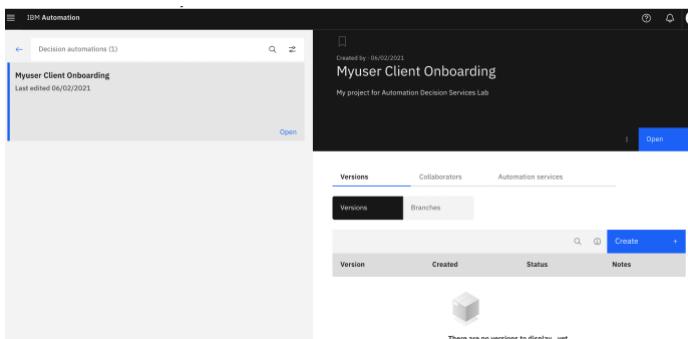
To deploy your decision service, you need to create a version of the current content. Versions correspond to tags in Git. A version is a snapshot of the project and records a point-in-time of the decision services within the project.

Creating versions requires following permission types for the project:

- **Admin** - Administrative privileges
- **Edit** - Write permission

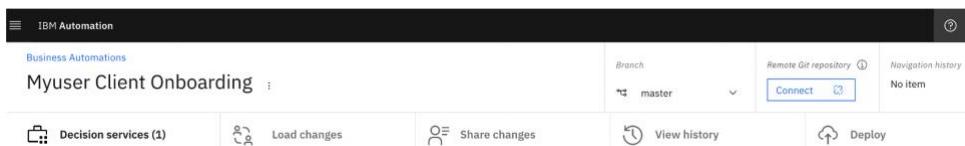
For more information, see IBM Documentation [Managing access to projects](#)

1. Return to your project by clicking on **Open** if you see the screen below

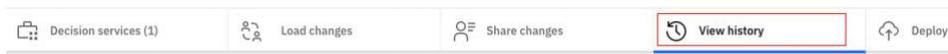


2. or navigate from the **navigation menu** in the upper-left corner to show (IBM Automation, Expand Design, Select Business automations, Select Decision, Select your project and click Open)

3. The project opens as below



4. Open the **View History** tab.



It shows all the changes made to the project.

View history (2)
View all the changes you and your collaborators shared.

Last shared	Shared by	Versions	
5/20/2021, 12:32:30 PM Myuser first Client Onboarding version	usr001	No version is created	Restore Version +
5/19/2021, 10:51:15 AM initial commit	usr001	No version is created	Restore Version +

5. Select the most recent version and Click on **Version +** button to create a version of your project
6. Enter a name for this version (add your user number in the name i.e v1.1usr100) and a description

7. Click on **Create**

A new version is created

View history (2)		
Last shared	Shared by	Versions
5/20/2021, 12:32:30 PM My user first Client Onboarding version	usr001	v1.1
5/19/2021, 10:51:15 AM Initial commit	usr001	No version is created

To deploy your version, you project must be connected to a Git repository. As next step you create a Git repository.

4.2.5 Creating a Git repository

As a prerequisite you must have a GitHub account to do this step.

1. Open [GitHub](#) in your browser, Sign-in by using your GitHub credentials.
2. Select **New** to create a new repository
3. Give the repository a unique name, and add the following description
Git repository for the Automation Decision Services Lab
4. Select Private and Click on the **Create repository** button
5. Click on the Copy button to copy the HTTPS URI. It has the following format:
`https://github.com/<yourAccountName>/<yourRepoName>.git`

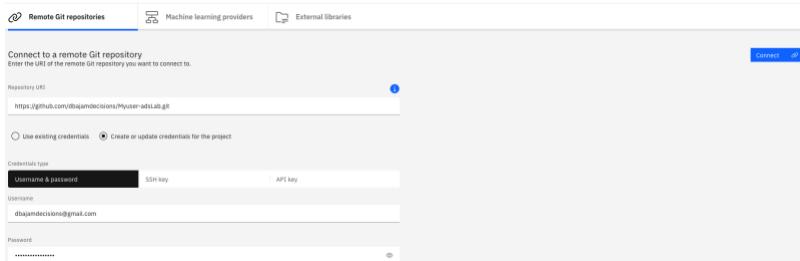
4.2.6 Connecting your project to the Git repository

You can manage your project on a Git repository where you will get all changes history from Automation Decision Services. Connecting to a Git repository is required to be able to build and deploy archives.

1. Check the status of **Remote Git repository** in the upper right corner of Decision Designer. It shows that the project is not connected.



2. Click on **Connect** to make a connection.
3. Enter the Git URI and the security credentials that you used to create the repository, and then click on **Connect**.

A screenshot of a "Connect to remote Git repository" dialog box. It has a "Repository URI" field containing "https://github.com/dbajanddecisions/Myuser-adslab.git". Below it is a radio button group for "Credentials type": "Username & password" (selected), "SSH key", and "API key". Under "Username", there is a field with "dbajanddecisions@gmail.com". Under "Password", there is a field with "*****". At the bottom right is a "Connect" button.

Upon successful connection Decision Designer displays the following messages and updates the status of Remote Git repository.

The screenshot shows the 'Remote Git repositories' tab selected in the navigation bar. A message box at the top right indicates 'Connected'. Below, a success message states 'The credentials are created or updated successfully.' and 'The remote Git repository is connected successfully.' Both messages are highlighted with a red border.

4. Refresh your repository page in GitHub. You can now browse the files that describe your decision service.

4.2.7 Deploying your project

You build and deploy your decision service directly in Automation Decision Services. A decision service archive is built from your decision services. This archive is deployed on an embedded repository for runtime archives. It is ready to be published as an automation service. Once deployed, an archive can be executed with a curl command.

1. Click on the project in the breadcrumb to navigate back to your decision project.

The screenshot shows the 'Business Automations' section with 'Myuser Client Onboarding' selected. The 'Settings' tab is active. The navigation bar includes 'Remote Git repositories' (selected), 'Machine learning providers', and 'External libraries'.

2. Open the **Deploy** tab and expand the **Version** you created in step 4.2.4.

Note: in this example the name of the version is 1.1, in your exercise add your user number as the name of the version.

The screenshot shows the 'Deploy' tab selected. A message at the top says 'Build and deploy a decision service. When it's done, you can publish it as an automation service.' A search bar 'Filter versions' is present. A table lists a single version entry:

Version	Shared on	Shared by
1.1 First version	5/20/2021, 12:53:43 PM	usr001
Decision service	Deployment status	DecisionId
Client Onboarding Lab	Not deployed	

At the bottom, there is a 'Deploy' button.

- Click on **Deploy** in the confirmation window. That triggers a build and deployment through the embedded repository for runtime archives. Wait for the deployment to finish.
- Once completed, you can check the logs by clicking the **View logs button** in the Deployment status.

The screenshot shows a table with columns: Version, Shared on, and Shared by. The first row shows 'v1.1' (First version), '5/20/2021, 12:53:43 PM', and 'usr001'. The second row shows 'Decision service' and 'Deployment status'. Below the table, it says 'Deployed on 5/20/2021, 1:31:24 PM'. The 'DecisionId' field is highlighted with a red box and an arrow points to it from the left. The 'Deploy' button is located at the bottom right of the table area.

Note that a **DecisionId** has been added. The decisionID parameter is required to call the decision service in the runtime service. It provides the decision path to the generated decision service archive. The decisionID can be used in the swagger UI for the runtime service. It is not covered in this Lab.

For more information, see IBM Documentation
[Executing decision services](#)
[Calling decision services in the embedded runtime service](#)

4.2.8 Publishing your decision service through Decision Designer

Note: The procedure is described as a reference information only.
Do not publish for shared environment to prevent mixing with other pre-defined automation services.

To publish the version of your decision deployed into the catalog of **Automation Services**, this is the procedure.

- Click on **Business Automation** in the breadcrumb



[Business Automations](#)

- Do not perform this step in a shared environment.** Click on the 3 dots menu to select **Publish**.

The screenshot shows a project named 'usr001 Client Onboarding'. The 'Versions' tab is selected. A table lists one version: 'v1.1' (Created: 5/20/2021, Status: First version). A red box highlights the 'Publish' button in the table. Below the table, a message says 'Publish the automation services from version v1.1.' and lists two items: 'Version v1.1 is deployed to the runtime environment.' and 'These automation services are now available.'

The **Status** is updated to Published

Version	Created	Status	Notes
v1.1	5/20/2021	Published	First version

3. Navigate back to **IBM Automation page** and check **Published automation services**, you see your decision service.

Published automation services (3)		
client_onboarding_decisions	Decision	Published 05/18/2021
client_onboarding_lab	Decision	Published 05/20/2021
Client_Onboarding_Workflows	Workflow	Published 05/18/2021

Once the archive is published as an automation service, you can execute it in other Cloud Pak for Business Automation capabilities. Look at the Workflow and Business Automation Application labs to learn how to work with pre-published decisions.

4.3 Summary

You have completed the Exercise 3 - Sharing and publishing decision service.

- You made updates to your decision services visible for other collaborators by sharing the changes.
- You connected your decision project to a Git repository.
- You created a version and explored the procedure to deploy and publish a decision archive.

Congratulations on completing the lab!

5 Additional information

IBM Documentation

[IBM Automation Decision Services](#)

[Building decisions](#)

[Glossary](#)

IBM Business Automation Community

[Decision Management](#)

Sample and tutorials for Automation Decision Services

[Samples and tutorials in GitHub](#)

DMN

[Decision Model and Notation](#)