Redpoint Automated Machine Learning

Version 6.2 User Guide

February 2021

Redpoint Global Inc., 36 Washington Street, Suite 120, Wellesley Hills, MA 02481
T: +1 781 725 0250 F: +1 781 235 3739 www.redpointglobal.com

Contents

1 Redpoint AML Overview

The following topics give a brief overview of machine learning and how AML implements machine learning.

1.1 Why use machine learning?

The rationale behind collecting copious amounts of customer data is to identify and leverage the information contained within it to provide the best possible, personalized customer experience. Knowing your customers' behavioral patterns allows you to make informed decisions when interacting with your customers in the future.

The amount of data collected about your customers and their behavior is huge and is constantly growing. Trying to manually find patterns within this vast amount of data can be very difficult.

Instead of trying to find data patterns yourself, you can use Machine Learning (ML, aka Artificial Intelligence/AI). In ML/AI, computers use mathematical algorithms to quickly find patterns among vast amounts of data.

One of the key concepts of ML/AI is that it allows the data to suggest the best method to recognize useful patterns instead of imposing human ideas and assumptions that may not be suited to the underlying data.

AML gives you the ability to use machine learning on your data without requiring data scientist-level expertise.

- If you have a solid understanding of the data or marketing process (are a *data citizen*), you can get involved with data modeling through AML.
- If you want to go deeper and tweak the machine learning process, AML gives you the tools to do so.
- AML hides the complexity of data modeling behind an easy-to-use, guided workflow where only critical decisions about the model need to be made by the data citizen.

1.1.1 How automated, optimized machine learning works

Machine learning algorithms build a mathematical model based on sample data (known as *training data*) in order to make predictions or decisions without being explicitly programmed to perform the task.

To start, you provide training data (which includes input and sometimes output data) to a machine learning program.

The machine learning program generates a set of *models* (mathematical functions that take input data and generate output data). You can use these models to make predictions based on input data. You can also think of a model as a potential solution, one of many.

Model changes use a genetic mutation evolutionary model, in which multiple generations of models are created and the best models of a generation are retained as parent models for the next generation.

For every generation:

1. Parents (AKA parent solutions or parent models) are created, and are the starting solutions for a generation.
2. Each parent produces one or more offspring (AKA offspring solutions or offspring models) that are variations of the parent.
3. For all the parents and offspring of a generation:
    a. Each model is trained.
    b. A fitness score is calculated for each solution.
        - A fitness score represents “goodness of fit”, and is not a percentage. It measures how much error the model has when compared to known “truth”.
        - The lower the value, the better the score. A fitness score close to zero indicates the model has a very low amount of error.
4. Out of all the generated possible solutions, the top solutions (the ones with the best scores) become parents for the next generation.

The training process ends when either a model reaches a predetermined score (measure of how accurately the model predicts results), or a calculation time period expires or a predefined number of training iterations is reached.

In a nutshell, this is a process of creating multiple generations of models. In each generation, the bad models are discarded, and the good models are retained and their traits passed on to their children. By feeding models back into the machine learning loop, successive models become more accurate. Eventually, the training process bottoms out (scores do not improve over time), and the best (top-scoring) solutions reveal themselves.

1.2 Getting help

We want to make your experience with AML as pleasant and productive as possible. Please email or call us when you need help. See Troubleshooting 8 for a list of information we'll need in order to help you.

Contact us at:

Redpoint Global Inc.
36 Washington Street, Suite 120
Wellesley Hills, MA 02481

Phone:	+1 781 725 0250
Fax:	+1 781 235 3739

Email: support@redpointglobal.com

1.3 Troubleshooting

When you contact us about a problem 7, please include the following information:

- Product version number (displayed in the lower right-hand corner of the web application login page).
- Your name, company name, telephone number, and email address.
- Operating system and version number.
- Exact wording of any error messages that appear.
- Exactly what you were doing when the problem or error occurred.
- How you tried to resolve the problem.

We may also ask you for:

- A sample of the input data.
- Your system configuration.

2 Log on to and off of AML

The following sections describe how to log on to and off of AML.

2.1 Log on to AML

For a Docker Swarm installation, every instance of AML has a unique URL:

`user/app admin: https://server:8983/#/Login`

`system admin: https://server:8993/#/Login`

/#/Login is the server path to the login page, and is optional. If not provided, the browser will redirect you to your instance server path after logging on.

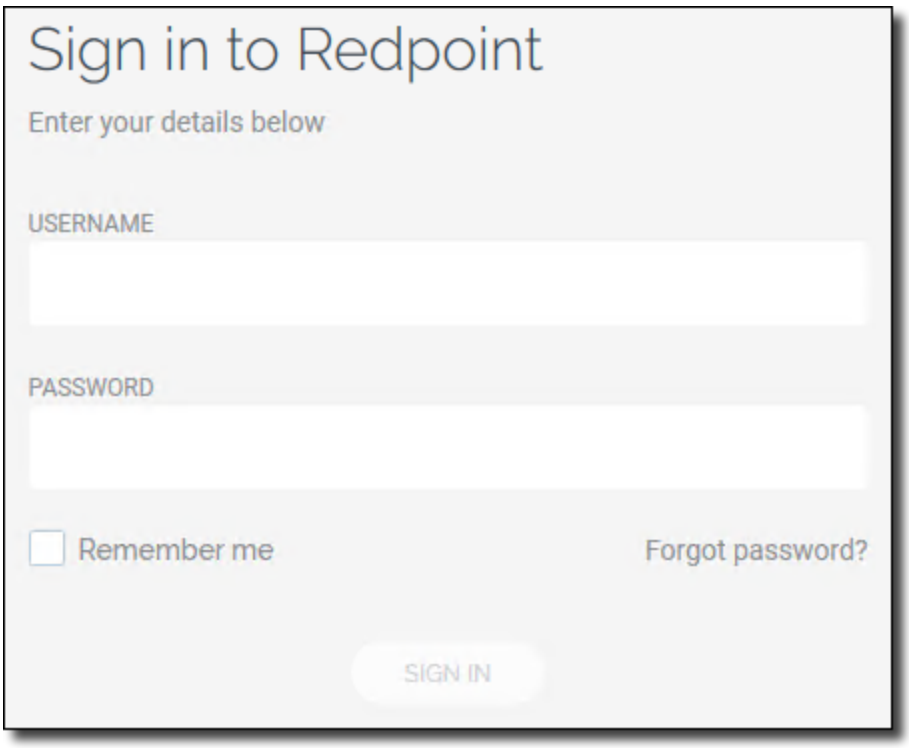
The person that installs AML will provide you with URLs and username/password pairs for user/app admins and system admins. See Installation and configuration 68 for this procedure, as well as the default installation username/password pairs.

Note

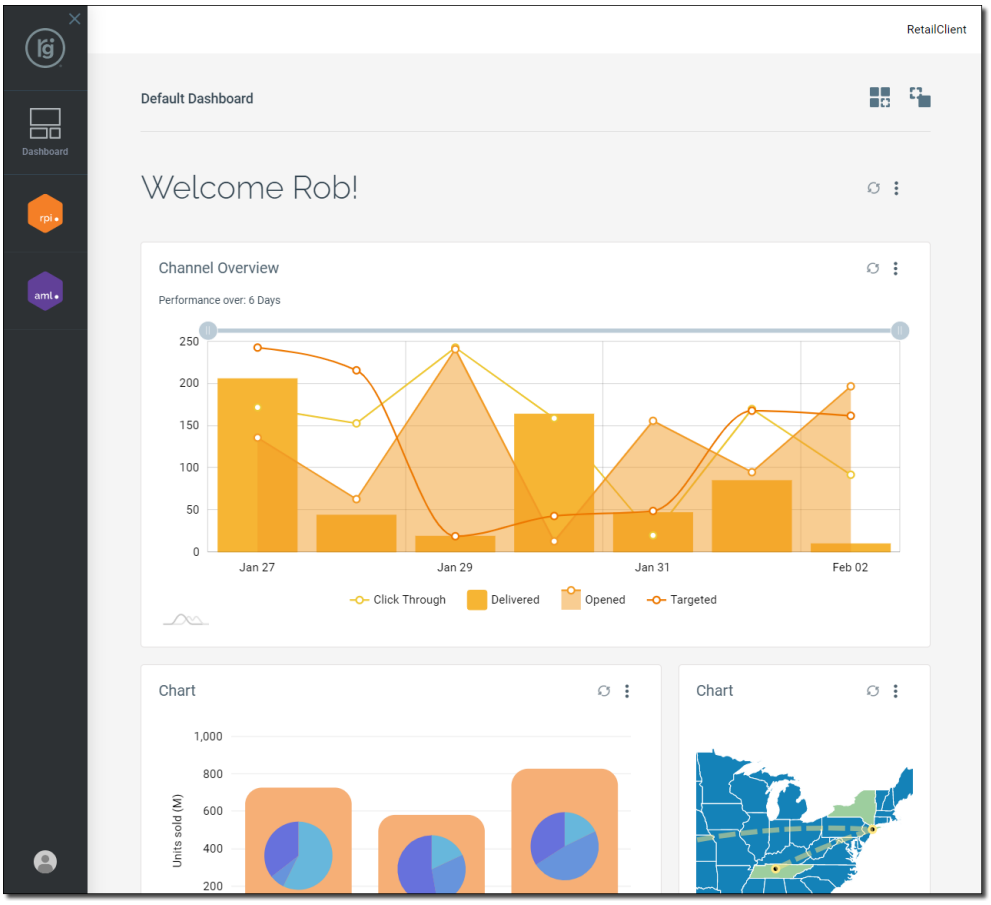
The AML web app is initially configured with two users: an app admin and a system admin. To add general user accounts, you must do so as an app admin 82.

1. Enter the appropriate user URL into your browser. The Redpoint sign-in page is displayed.
    - Use the user/app admin URL if logging on for the first time or if you are logging on as a regular user.

- If you are logging on as a system admin, use the system admin URL.



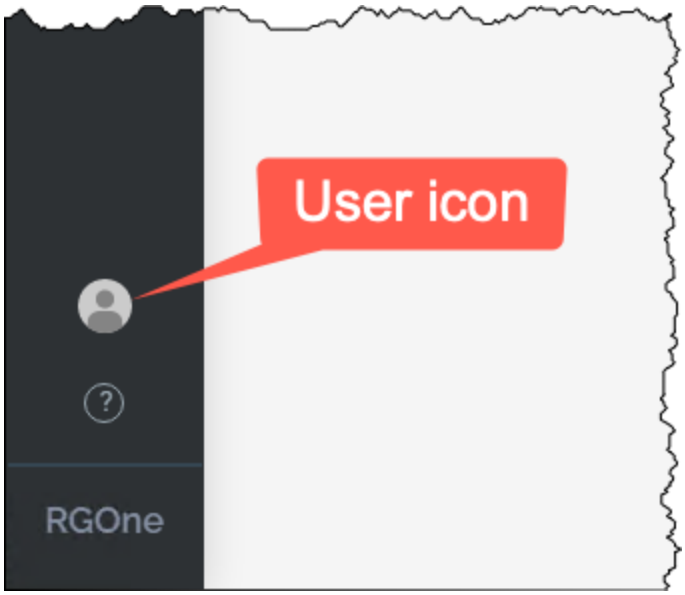
2. Enter your username and password.
   - If you select **Remember me**, the **Username** box will be auto-populated with your username when you log on in the future.
   - If you click **Forgot password?**, a dialog is displayed that allows you to enter your email address so that your password will be reset, and you will be sent a new temporary password. Once you receive your new temporary password, we recommend that you immediately change it.
3. Click **Sign In**.
4. The rgOne dashboard is displayed. It displays the set of widgets you have customized and added to the dashboard. The left vertical navigation bar displays icons of your installed Redpoint apps.



5. On the vertical navigation bar, click the AML icon to display the app section icons.
6. Click one of the AML app section icons to display the section page.

2.2 Log off of AML

On the left-side navigation menu of any web app page, click the **User** icon and on the popup menu select **Sign Out**.



3 The AML app user interface

The following sections describe the visual elements of the AML app.

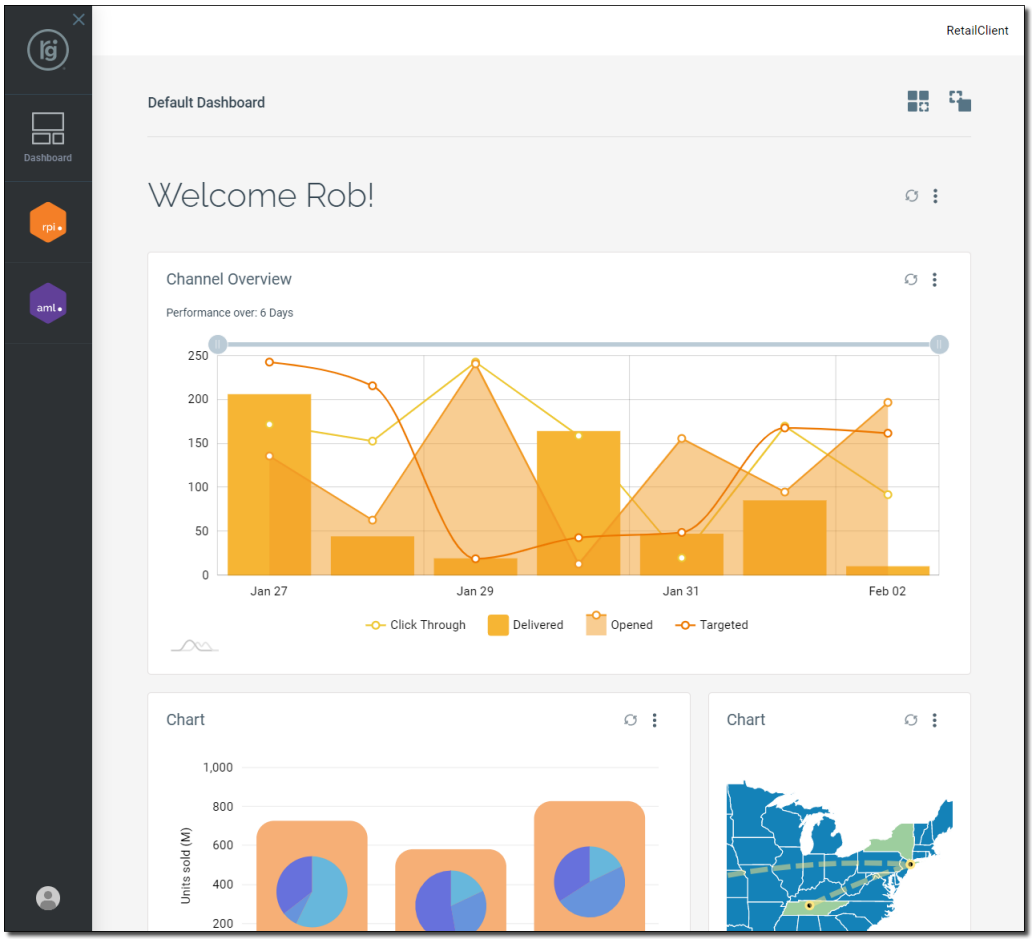
Note

All AML functionality is also available through the AML API 132.

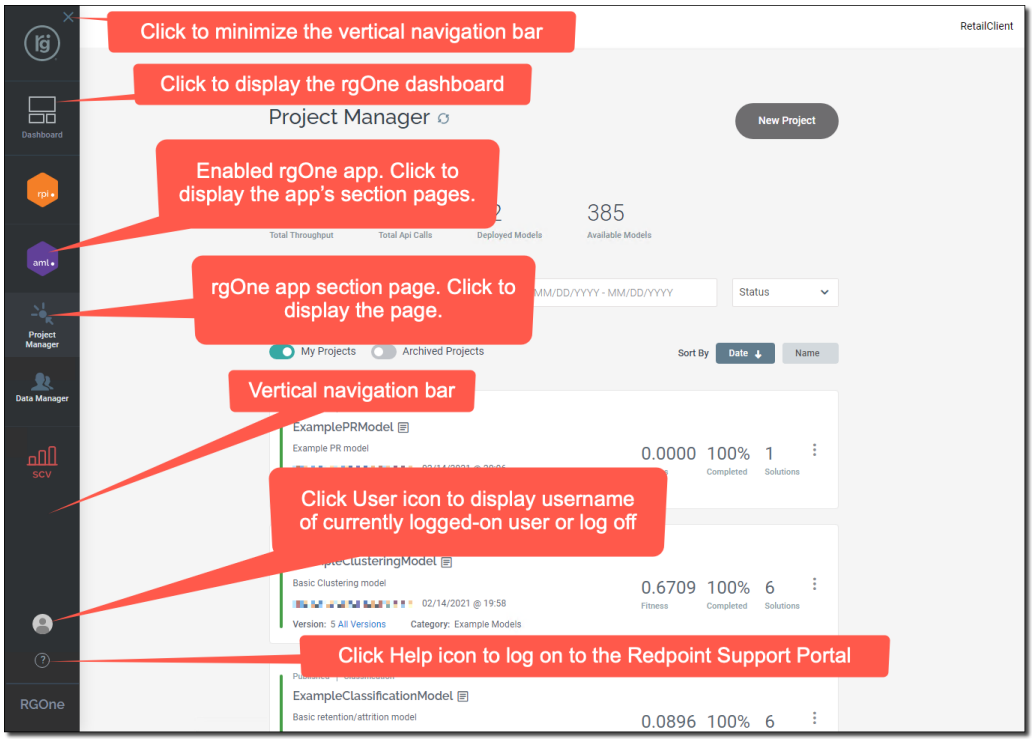
3.1 AML app navigation

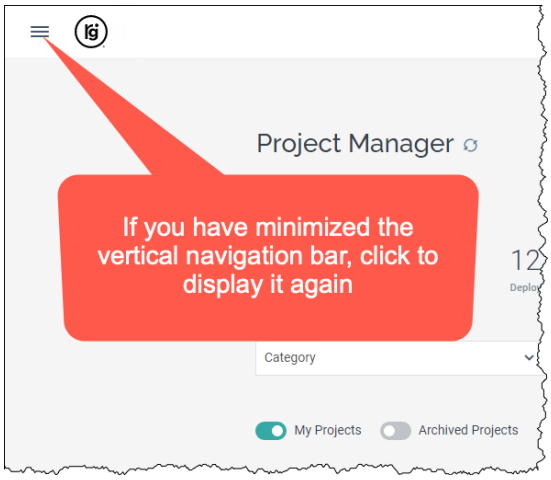
When you first log on to AML, the left vertical navigation bar displays your installed rgOne apps, as well as a link to the rgOne default dashboard.

The main viewing area displays the rgOne dashboard, which is comprised of widgets you have previously enabled and configured. See @@new widget section@@ for procedures on how to enable and configure these dashboard widgets.



When you click on the AML icon on the left vertical navigation bar, the main app sections are displayed. Click on an app section to display it in the main workspace to the right.



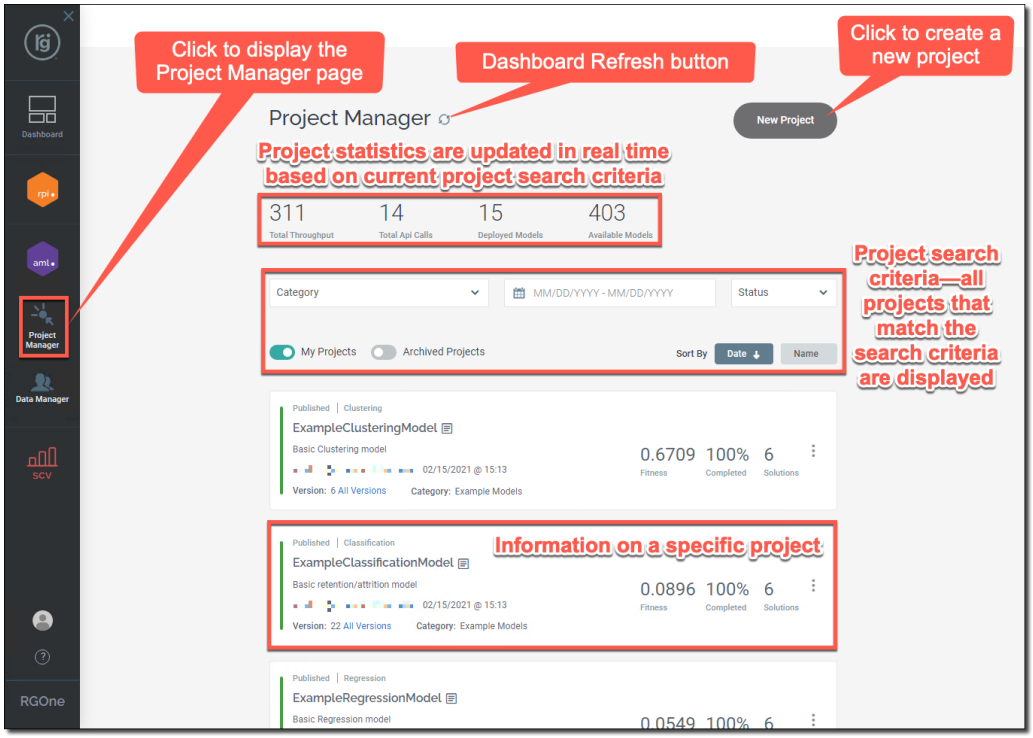


3.2 Project Manager

This page is used to manage AML projects. Specifically, this page allows you to do the following things with projects:

- create
- edit
- stop and resume training
- view project settings

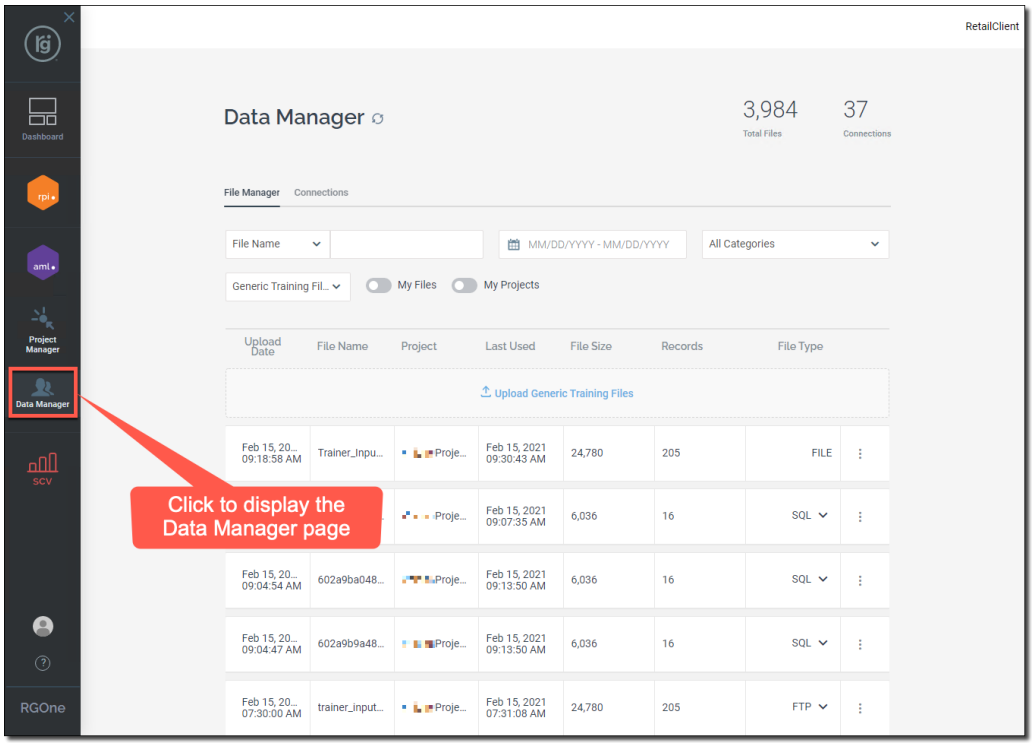
- archive
- clone
- view a project's fitness report
- view a list of the project's generated solutions



Clicking the **Refresh** icon queries the database for any state changes, using the current dashboard filters. It is especially useful when periodically checking the progress of projects in training. We recommend that you use the dashboard **Refresh** icon because this updates the dashboard much faster than refreshing the entire page using the browser refresh button.

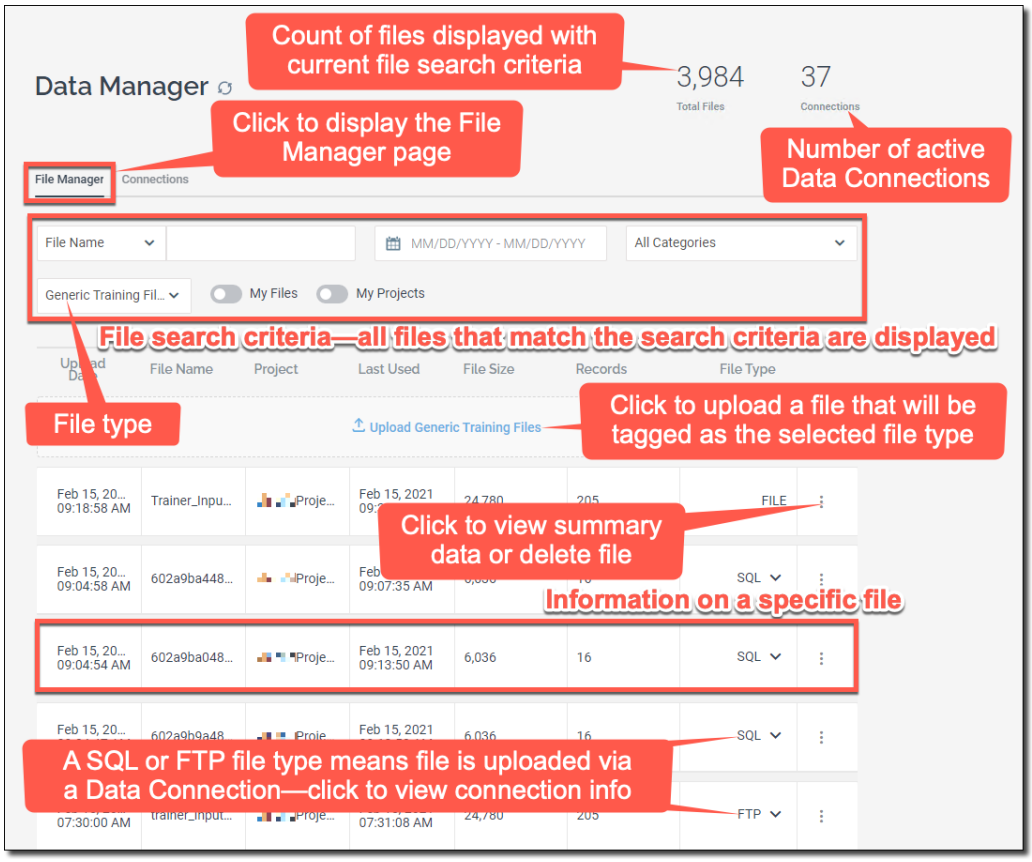
3.3 Data Manager

This page is used to manage the data files used for AML projects.



3.3.1 File Manager

The File Manager is used to upload files to AML, display their characteristics, and keep track of them once uploaded.



Notes

If you upload a file through the Project Manager, a modified version of the File Manager is displayed.

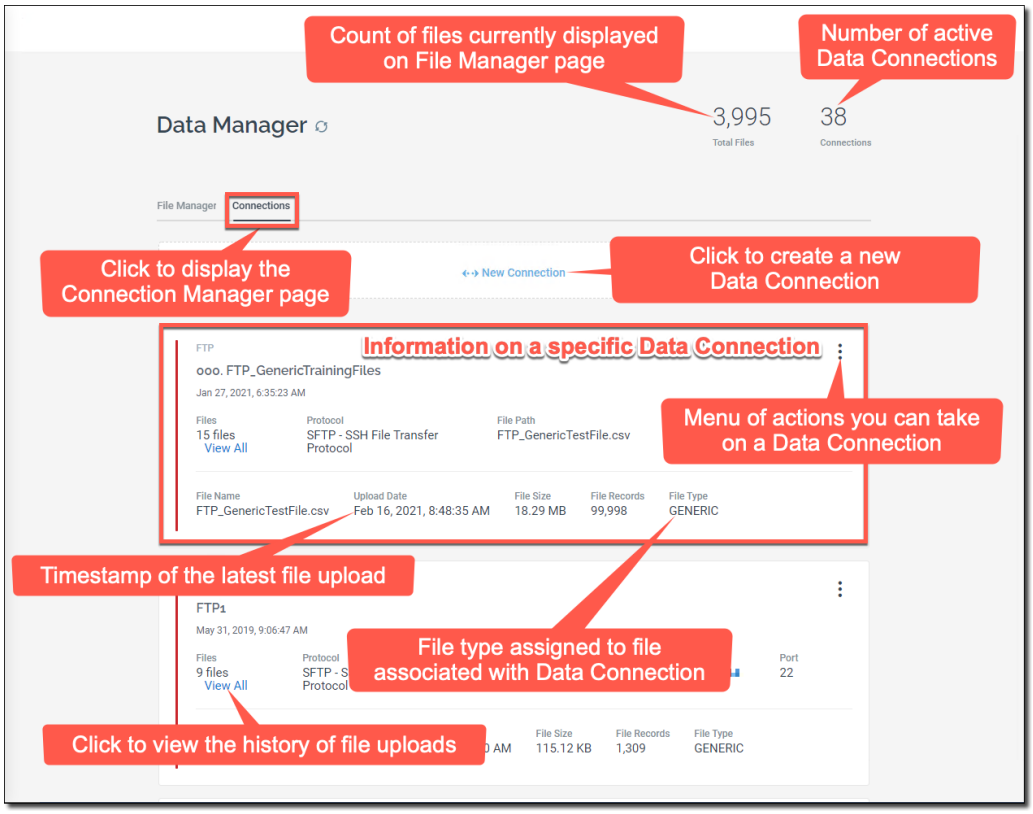
You can only delete files that you have uploaded.

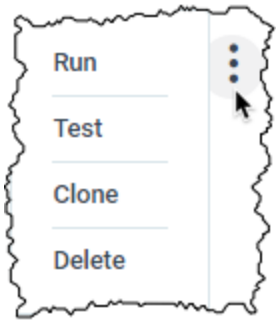
When you view a file's summary day, up to the first 200 records are displayed.

File size values are in bytes.

3.3.2 Connection Manager

The Connection Manager is used to create Data Connections, display their characteristics, and keep track of them.





The **Action** menu on a Data Connection allows you to do the following things:

Run—Direct the Data Connection to immediately upload the associated file.

Test—Test that the connection works.

Clone—Create a copy of the Data Connection. You can change any of the settings during the creation process. If you clone a Data Connection another user has created, you must re-enter the password value.

Delete—Delete the connection (you can do this only if you created the connection).

For more information on Data Connections and how to use them, see To publish and choose a data file via a Data Connection 35.

4 Create a new project

In a nutshell, a project is the atomic unit of AML. It defines all the constraints used when generating a set of solutions for a given model.

The following topics explain exactly what a project is and how to create one.

4.1 Models vs. solutions

Before we continue, let's distinguish between a *model* and a *solution*. In theory they refer to the same thing (a mathematical function that takes input data and generates output data), but the name changes depending upon where it is in its life cycle. In AML, the initial models a project generates are called *solutions*. When one of these solutions is published and can be used for prediction, we refer to it as a *model*.

4.2 What is a project?

A project:

- Defines all the constraints used when generating a set of solutions. Solutions generated by a project can be reviewed, discarded, or published. A user may want to experiment with different project constraints to see how these changes affect the solutions that are created.
- Provides the mechanism for tracking the different versions of constraints used when generating solutions.

4.3 Steps for building a project

In general, these are the steps for building a project in AML:

1. On the **Project Manager** page, click **New Project**.
2. Give the project a name and description, and select a category.
3. Choose whether to create a project from an existing template or a basic model type.
4. Select the model training files (which may require you to upload the training data to AML).
5. Select the model's input and output features. (A *feature* is a single column from the input data file.)

4.4 Choose a template or model

There are two ways to create a project in AML:

- Choose a template from a list of business problems you want to solve. Each of these templates is optimized to solve a specific problem, such as "predict customer sales" or "predict yearly/monthly spend".
- Choose a basic data model type.

In either case, AML chooses initial project settings to get you started quickly.

4.4.1 Project templates

You can configure an AML project by selecting from a list of business problems you want to solve. Each of these templates is optimized to solve a specific problem, such as "predict customer sales" or "predict yearly/monthly spend". When you select a business problem, AML automatically creates a project, chooses the best machine learning model, and chooses initial project settings to get you started quickly.

The following table describes each of the available project templates.

Template	Description
Anomaly Detection	Predict expected characteristics (sales quantity, data throughput times, network activity, and so on) and apply subsequent thresholds to the predicted results to detect significant deviations (for example, 2-sigma over/under prediction). Input features typically may consist of process control parameters (for example, temperature, pressure, time of day, web page hits, DB hits) and other monitored operating environmental conditions.
Predict Best Message Channel	Based on the historical behavior exhibited for each customer's interactivity with the company, predict the channel with the highest probability of response. Input features typically consist of demographic info, previous purchases, purchase frequency, purchase prices (avg, median, min, max), message channel, discount, purchase recency, and so on. The (discrete) message channel to predict should be contained in the set of known (previous) responses to messages on the set of channels used in the historical data, without inclusion of a "none-of-the-above" enumeration option.
Predict Best Message Content	Based on the historical behavior exhibited for each customer's interactivity with the company, predict the message content with the highest probability of response. Input features typically consist of demographic info, previous purchases, purchase frequency, purchase prices (avg, median, min, max), message channel, discount, purchase recency, and so on. The (discrete) message content to predict should be contained in the set of known (previous) responses to

Template	Description
	messages, without inclusion of a "none-of-the-above" enumeration option.
Predict Customer Retention/Attrition	Based on historical behaviors, predict the probability of a customer being retained (that is, likely to repeat previous behaviors). Input features typically consist of demographic info, previous purchases, purchase frequency, purchase prices (avg, median, min, max), message channel, discount, purchase recency, and so on. **Note**: Likelihood for churn (attrition) can be easily predicted by inversely ranking the attrition candidates (that is, probability of attrition = 1 - probability of retention).
Predict Customer Sales	Based on historical behavior (spending patterns, time differentials between purchases during a specified timeframe), predict the amount of money the customer will spend relative to an expected timeframe (for example, week, month, year). Input features are typically similar to those used in predicting attrition/retention.
Predict Likelihood to Rebook	Based on historical behaviors, predict the probability of a customer rebooking (for example, likely to repeat previous bookings to a hotel, residence, or other property). Input features typically consist of demographic info, previous purchases, purchase frequency, purchase prices (avg, median, min, max), message channel, discount, purchase recency, and so on.
Predict Time to Next Purchase	Based on historical behavior (differentials between purchases during a specified timeframe), predict time before the customer will purchase something (spend). Input features are typically similar to those used in predicting attrition/retention.
Predict Yearly/Monthly Spend	Based on historical behavior (amount spent during a specified timeframe), predict the customer spend for the next desired timeframe. Input features are typically similar to those used in predicting attrition/retention.

Template	Description
Segment Customer Behaviors	Assign groups/segments/clusters based on similarities within the data features. Clusters represent membership within specific groupings based on these similarities. Input features typically consist of demographic info, previous purchases, purchase frequency, purchase prices (avg, median, min, max), message channel, discount, purchase recency, and so on. Predicted clusters can be used (post-analysis) to drive marketing messaging and offers, as well as derive insight into inter-segment and intra-segment differences.
Segment Customer Demographics	Clusters represent membership within specific groupings based on these similarities. Input features typically consist of demographic info, previous purchases, purchase frequency, purchase prices (avg, median, min, max), message channel, discount, purchase recency, and so on. Predicted clusters can be used (post-analysis) to drive marketing messaging and offers, as well as derive insight into inter-segment and intra-segment differences.

4.4.2 Data models

In AML, these are the basic (atomic) data model types. You can create a project based directly on one of these types. If you create a project using a template, the template will ultimately be based on one of the following model types.

Model type: Classification

Classification models use existing data points and variables to classify future outcomes based on the existing training data. Both defined inputs and (correlated) outputs are used to train the model.

Classification models are typically used to predict discrete-value outputs (for example, whether a record belongs in class 1, class 2, or class 3).

You can read more about the Classification model here 58.

Model type: Clustering

Cluster/segment analysis (AKA *clustering/segmentation*) is the task of grouping a set of objects in such a way that objects in the same group (called a cluster or segment) are more similar (in some sense) to each other than to those in other groups (clusters/segments).

A common use of clustering/segmentation is dividing users, customers, or subscribers into clusters/segments of individuals based on similarities that may be relevant to your marketing. One can perform cluster/segment analysis on customer attributes to answer questions such as:

- What are the demographic characteristics of my best customers?
- How do customers behave while purchasing?
- What groupings of products do people buy from?
- How many clusters/segments best describe this data? (For example, three clusters/segments, five, six, ten?)

You can read more about the Clustering model here 59.

Model type: Regression

The Regression model uses a set of input features to predict output features. Input features can be continuous (floating point), discrete (integer), text (turned into enumerated integer values), and binary (0/1).

The output features to predict are usually continuous values (like sales or cost). While Regression models can also be used to predict discrete values (like yes/no, categories, and so on), Classification models are generally better for discrete-value cases.

Regression models can be used to predict things like:

- profit
- potential sales
- transaction volume
- credit score

You can read more about the Regression model here 62.

Model type: Product Recommender

The Product Recommender (PR) model compares customer purchase history or other customer preference attributes to a set of product attributes, and generates one or more product recommendations for each customer.

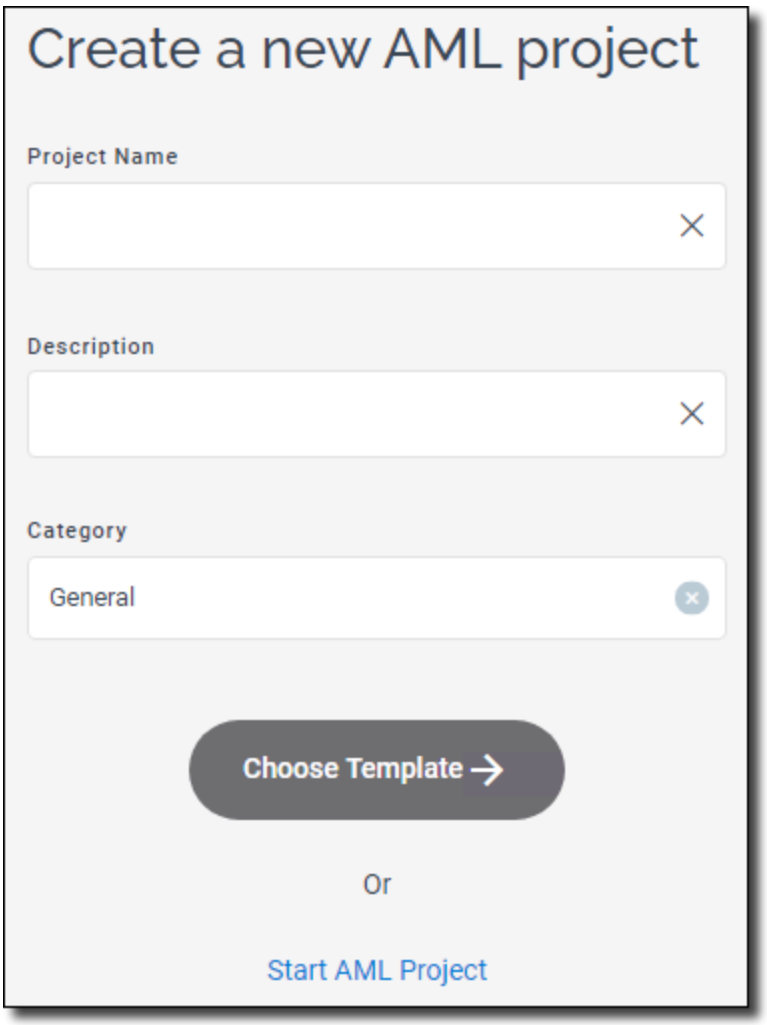
You can read more about the PR model here 64.

4.4.3 To choose a template or model

1. Navigate to the **Project Manager** page.

2. At the top right of the page, click **New Project**.

The **Create a new AML project** dialog is displayed.



3. Enter a **project name**.
4. Enter a short project **description** (optional).

5. Choose an existing **category** value or create a new one (optional).

   To create a new category (If your AML admin user has assigned your account a role with permission to create project categories):

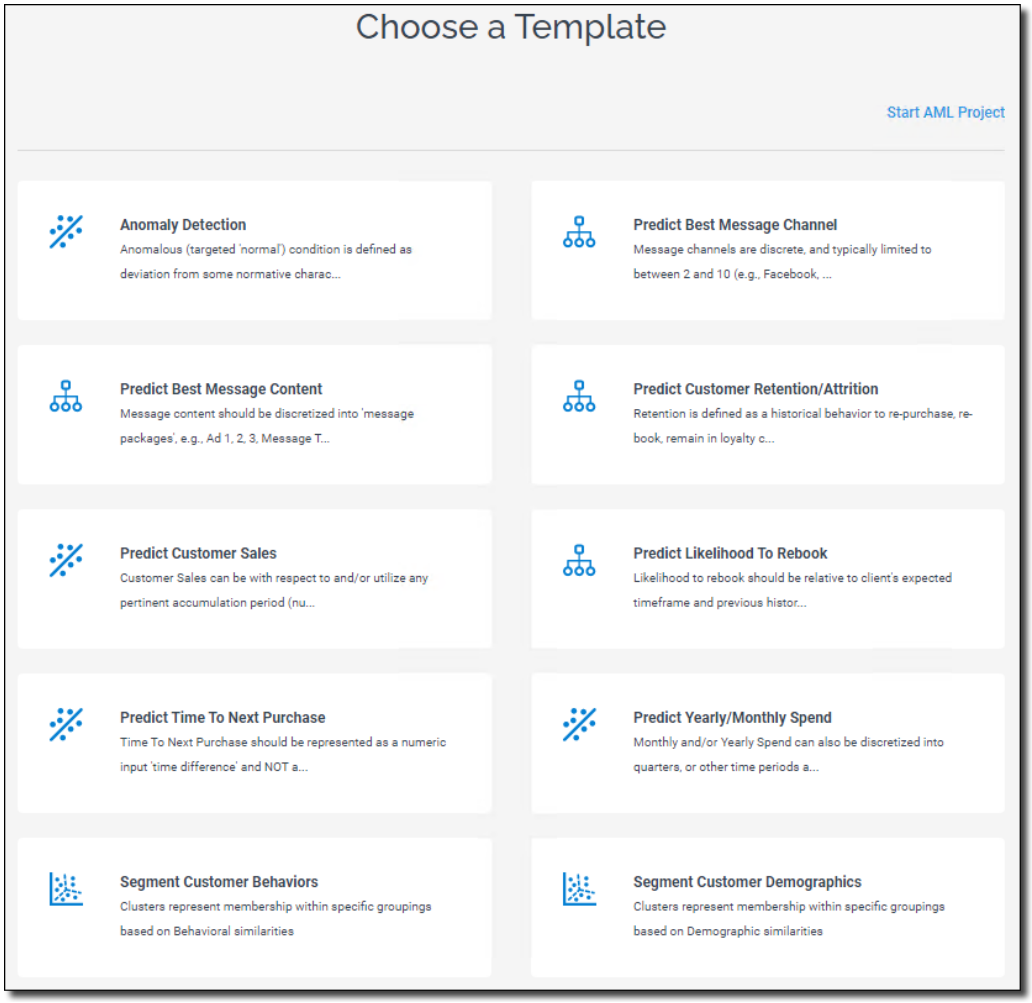
   a. Click inside the **category** box.

   b. Delete the default **category** value "General".

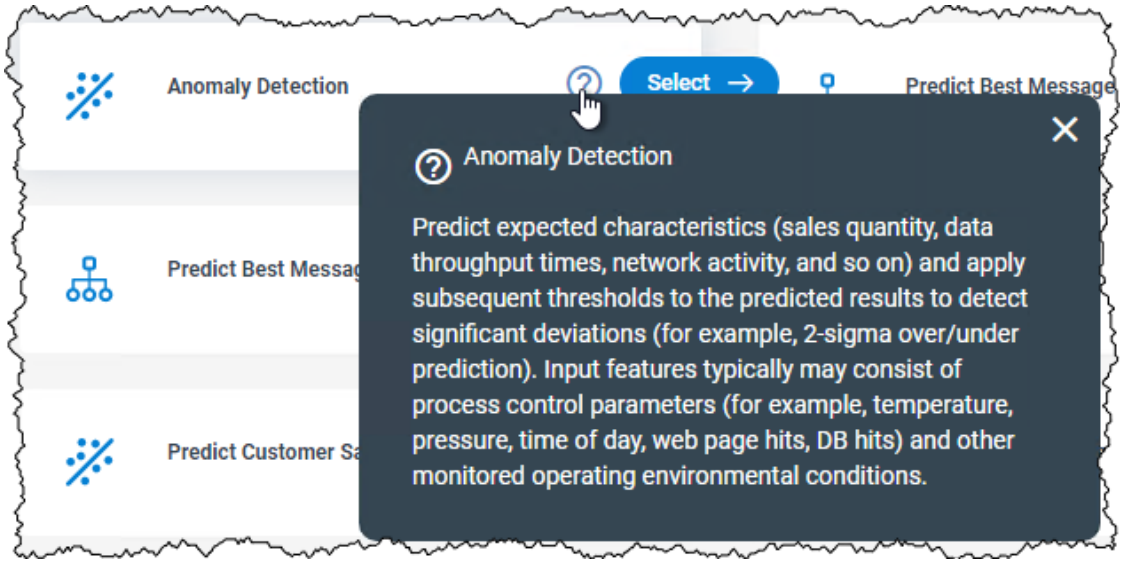
   c. Enter a new **category** value.

6. Click **Choose Template** or the **Start AML Project** link.

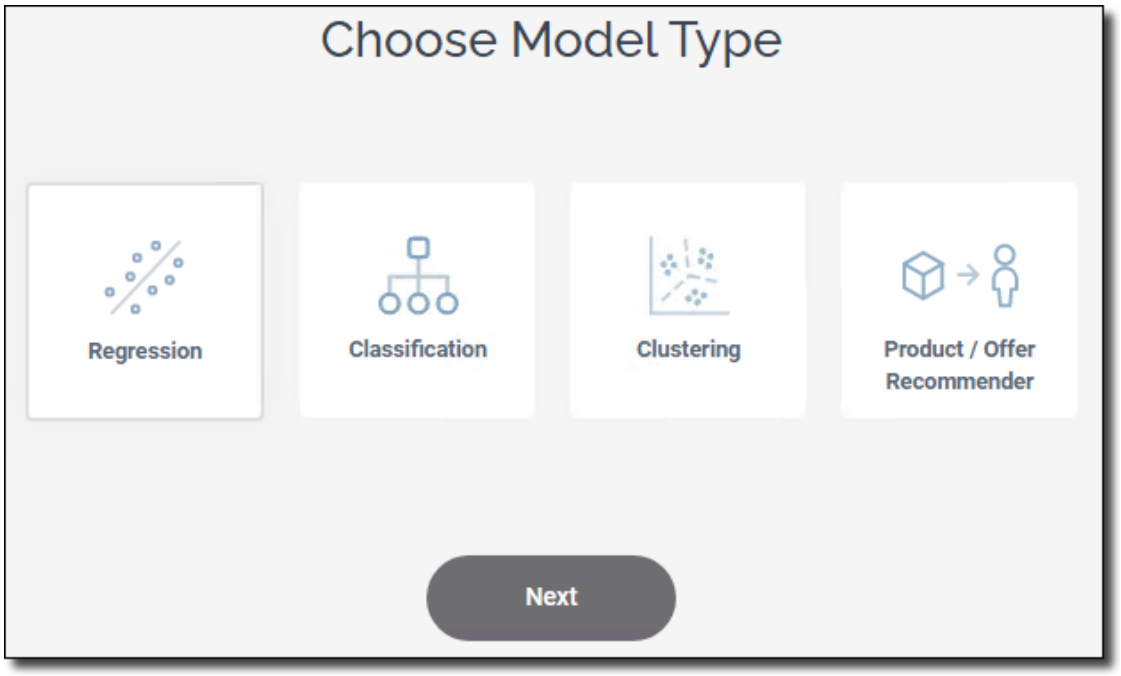
   - If you click **Choose Template**, the **Choose a Template** page is displayed. Hover your mouse pointer above a template card and click **Select**.



   A description of a template is displayed if you hover your mouse pointer above the template card and click the help (**?**) icon.



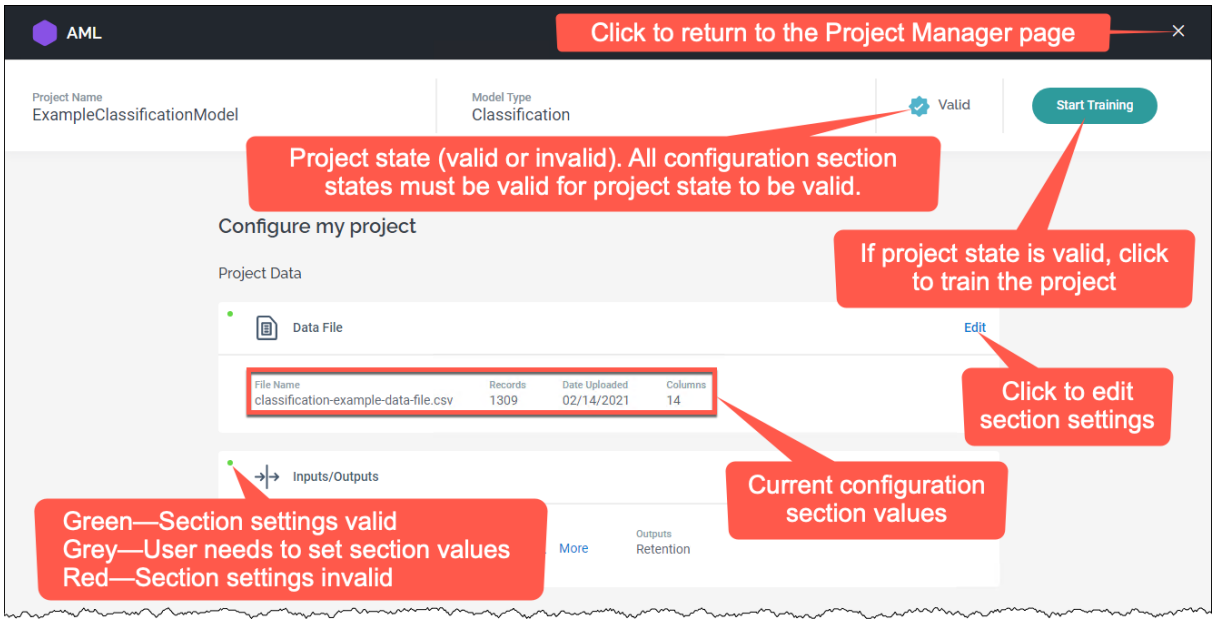
- If you click the **Start AML Project** link, the **Choose Model Type** dialog is displayed. Click on a model type and click **Next**.



7. The **Project Configuration** page is displayed.

4.5 Project configuration page

When creating a project, after you choose whether to base the project on a template or a model, the **Project Configuration** page is displayed. It displays (at a high level) a project's current settings, which are split into sections.



When creating a new project, or when not actively training a project, you can edit the settings in each section.

Note that:

- You can edit sections in any order.
- You are required to edit only the **Data File** and **Inputs/Outputs** sections.
- Editing all other sections is optional, and these sections have reasonable default settings for most common cases.

4.5.1 Project Configuration page sections

A **Project Configuration** page is split into two main sections:

- **Project Data**
- **Project Configurations**

Project Data

This section chooses what data is put into the project, and how those data fields (aka *features*) are used as inputs and outputs for training the models generated by the project.

Data File

This subsection lets you upload data files, or choose data files that you have previously uploaded.

Inputs/Outputs

These fields are the inputs and outputs for the solutions being trained. A project trains and produces multiple solutions, all of which use the same set of inputs and outputs defined in this project configuration section.

Project Configurations

This section is focused on project (as opposed to data) settings. For a detailed explanation of the settings in the following subsections, see Project Configuration page settings 108.

Training Options

These are high-level model settings that control such things as which model algorithms to use and specific options for each algorithm (such as the number of nodes and layers to use in the Neural Networks algorithm).

Data Options

This subsection controls:

- Training/testing set size and sampling technique
- If data is normalized for input and output features
- Training set vs. testing set distribution

Optimizer

The optimizer automatically tries to refine and tweak your modeling during each iteration of training.

Run Settings

These settings focus on the project training process. These settings include:

- Training time limit.
- Model performance score (answers the question "how good is good enough?").
- Whether to automatically publish the best solution.

Automation Run Settings

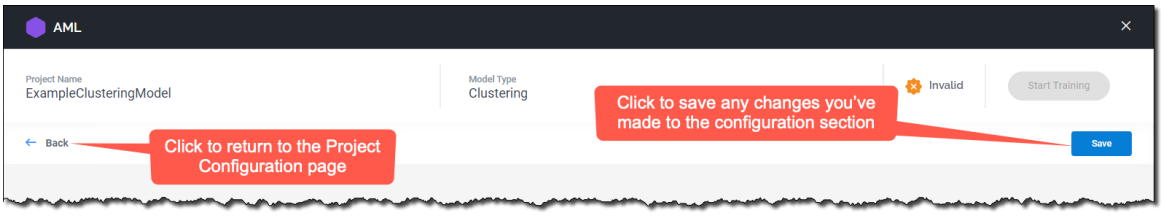
This subsection allows you to automate project training. For example, you could direct that a particular project be retrained every Tuesday at 6pm, but only if the input data file has been modified.

4.5.2 To edit a section

To edit a section on the **Project Configuration** page:

1. Click the section's **Edit** link. The section page is displayed.

   When you edit a **Project Configuration** page section, additional **Save** and **Back** buttons are displayed at the top of the page.



2. Choose/change the settings as you see fit.

   If you change one or more settings in a section, the changes are not permanent until you click **Save**.

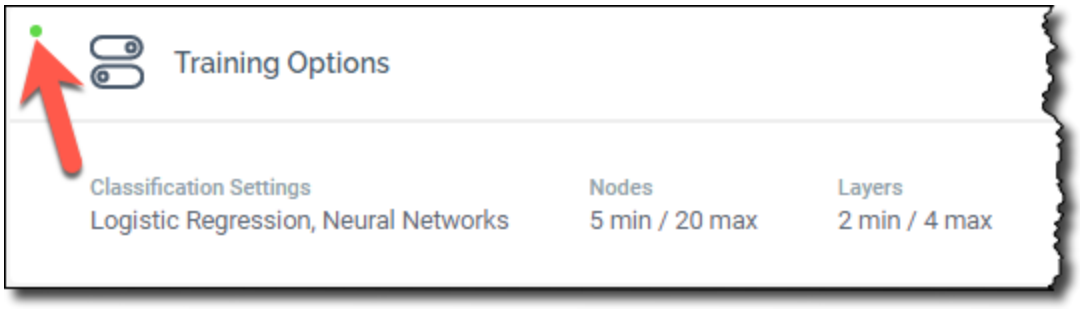
3. Click **Save**.

4. Click **Back**. The **Project Configuration** page is again displayed.

   If you don't save your changes and click **Back**, you will be given the option of saving your changes before returning to the **Project Configuration** page.

4.5.3 Section warnings

On the **Project Configuration** page, every section displays either a grey, red, or green light.



- Grey—A user needs to set the section settings.
- Green—All the section settings are valid.

- Red—One or more section settings is invalid.

If a section light is red, edit the section so that all the settings are valid (the section light turns green).

4.5.4 Project validation

If one or more section lights are red:

- The project status at the top of the page displays the message "Invalid".
- The project is not ready to train.
- Change the settings in each invalid section to make that section valid (the section light will turn green).

When all of the section lights are green, the project is ready to train.

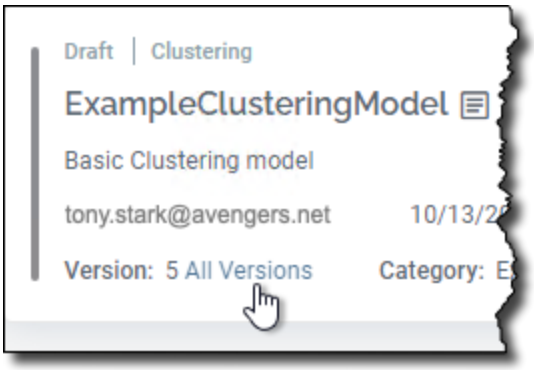
- The project status at the top of the page displays the message "Valid".
- You can start training the project by clicking the **Start Training** button.

4.6 Draft status

Whenever you edit a project (new or existing), the project status is changed to **Draft**.

A new project is assigned a version value of 1. If you edit an existing project and save the changes, the project version number is incremented by 1. Previous versions of projects are saved, and you can access them through:

- The **All Versions** link on the **Project Manager** page project card.



- The **Version** dropdown menu on a project **Solutions** page.

Note

If you edit a project that has completed training (change the project status from **Completed** to **Draft**), the metrics on the **Project Manager** page project card are no longer displayed. This is a known issue, and will be corrected in a future software version.

4.7 A brief aside about data file notation and project features

Data files using the `.csv` format (the kind of data files that AML uses) contain one or more records, with each record containing one or more fields. If you look at such a data file in a text editor, each row corresponds to an individual record, and the columns correspond to fields within the records.

For an AML project, we add a data file that provides:

- Input (independent) values. In ML/AI, these fields are known as *input features*. You can define one or more input features.
- Predicted output (dependent) values. In ML/AI, these fields are known as *output features*. For a given record, this is the output value(s) based upon the given input value(s). You can define zero or more output features (zero, if the project type does not generate output features, one in the case of a Classification project).

  During training, the values in the output fields are used for supervised learning. That is, the project predicts the output value(s) based upon the input value(s), and the file output value(s) are used to determine if the project's output prediction(s) are correct.

  After training, a user selects one of the project solutions and publishes it (turns it into a model). When a user calls the model, the user submits only the input field(s), and the model returns the predicted output(s) for the given set of input(s).

When thinking about a project's features, we recommend that you first decide which field(s) to use as the output feature(s). This is the most important decision you will make during the project configuration process because this decision ultimately decides what the model will be used for.

Once the output feature(s) are set, you will generally not change this mapping between different versions of the model. However, you may significantly change the mapping of file fields to input features between versions of a model as you see which inputs have the most influence on the output.

For more guidance on selecting output and input features, refer to Features 87.

4.8 Add data files

This section describes the process for uploading external data files into AML and choosing these uploaded files for use in a project.

4.8.1 Data set types used by AML

A data file used by AML must be tagged as one of the following types. You assign a file type tag when you upload the file.

Generic training

This file type can contain almost any kind of data useful for machine learning, and is generally used for a non-PR model. However, any file can be tagged as **Generic Training**.

Product/Offer data

This file type contains product attributes, which are keywords associated with each product ID. It is usually used for a PR model.

Only a file tagged as **Product/Offer data** can be used as the product file in a PR model.

Customer data

This file type contains customer history and/or customer preference data. It is usually used for a PR model.

Customer history data are product IDs associated with a customer ID. Generally, this is a list of transactions showing products the customer has purchased.

Customer preference data are keywords or product IDs associated with each customer ID. Generally, this is a list of preferences or products associated with a customer, often assumptions based on the customer's behavior or self-identified traits.

Only a file tagged as **Customer data** can be used as the customer file in a PR model.

4.8.2 File upload methods

There are two ways to upload files to AML:

- Via the File Manager
- Via the Connection Manager (creating a Data Connection)

Uploading files via the File Manager

The File Manager is located on the **File Manager** tab of the **Data Manager** page.

You can manually upload a file once using the the File Manager. If you want to upload a modified version of the file, you must repeat the process and upload a new file.

You can think of uploading a file through the File Manager as a static process. If you don't think the file contents will change over time, this is a convenient method to use.

Uploading files via the Connection Manager (creating a Data Connection)

The Connection Manager is located on the **Connections** tab of the **Data Manager** page.

Instead of uploading a file once through the File Manager, you can define a Data Connection that can upload it automatically and replace the existing file.

A Data Connection stores the file location, the credentials, and the connection information necessary to access the file. Data Connections can be of the following types:

- FTP
- SQL

You can manually initiate a file upload using the Data Connection, or you can automate the process so that a file is uploaded at a set interval.

To enable Data Connection functionality, see Deploy AML and link it to a DM system.

Note

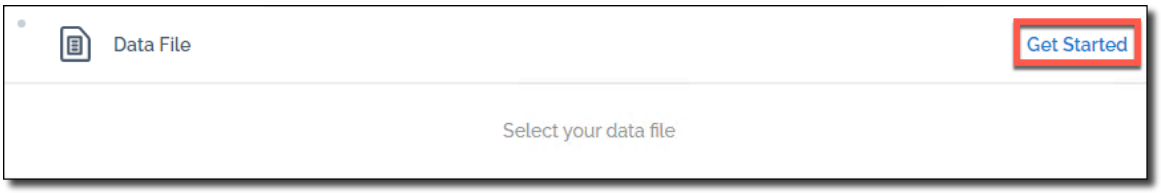
A Data Connection allows you to upload a file that has previously been placed in a specific location. For example, if you want to create a Data Connection to a file on an FTP site, that file must already be copied to a specific folder on the FTP server.

4.8.3 To upload and choose a data file via the File Manager

Note

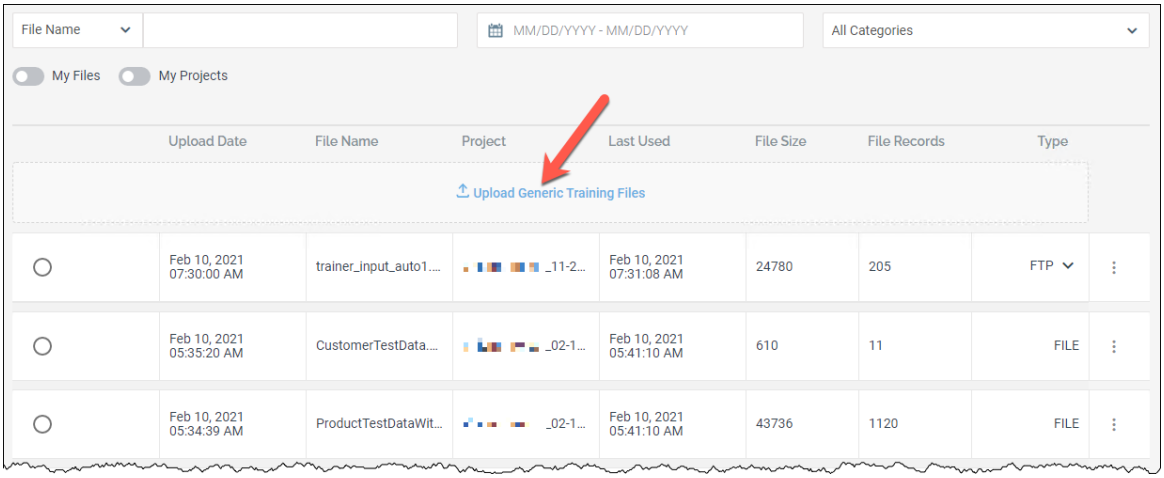
Some project types may use more than one data file. See Data set types used by AML 32 for more information.

1. On the **Project Configuration** page, In the **Data File** section, click the **Get Started** link. The File Manager is displayed.



2. Click the **Upload Files** link.

   The file type you can upload is determined by the type of project you are creating. See Data set types used by AML 32 for more information.

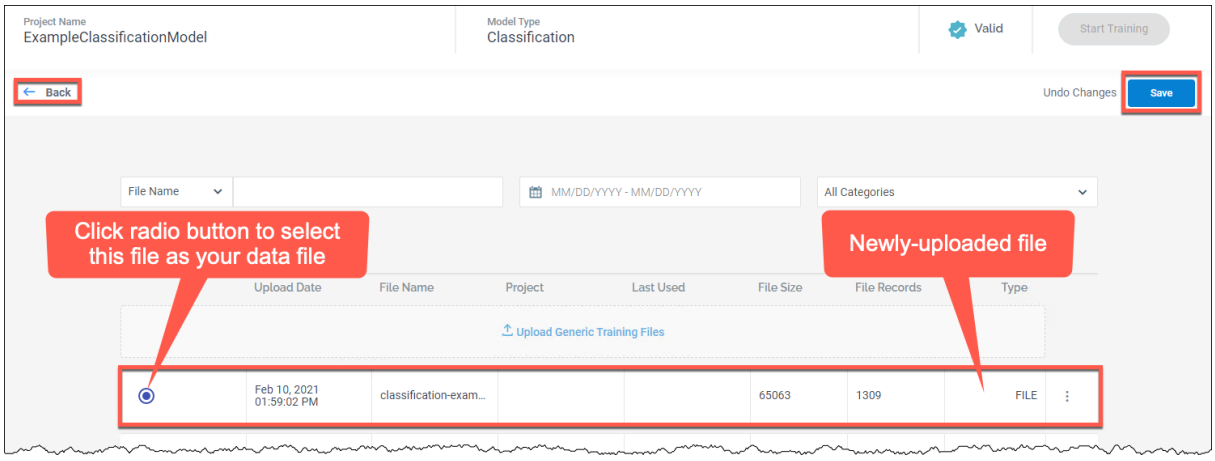


3. Browse for the file on the file selection dialog.

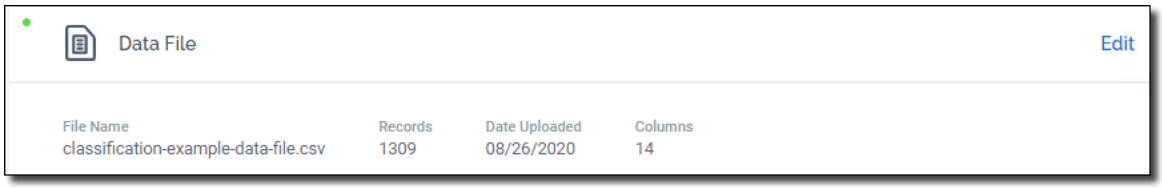
   Once a data file is uploaded, AML:

   - Automatically determines the data schema.
   - Profiles the data to gather statistics about the internal content.
   - Finds missing data elements.
   - Looks for any problems that might affect the modeling process.

4. In the File Manager, the newly-uploaded file is displayed in the file list. Click the file radio button to select this file as your input data file.



5. At the top right of the page, click **Save** to save your file choice.
6. At the top left of the page, click **Back**.
7. On the **Project Configuration** page, the **Data File** section now displays the file you just uploaded and selected. If the section indicator light is green, the selected data file(s) are valid.



4.8.4 To upload and choose a data file via a Data Connection

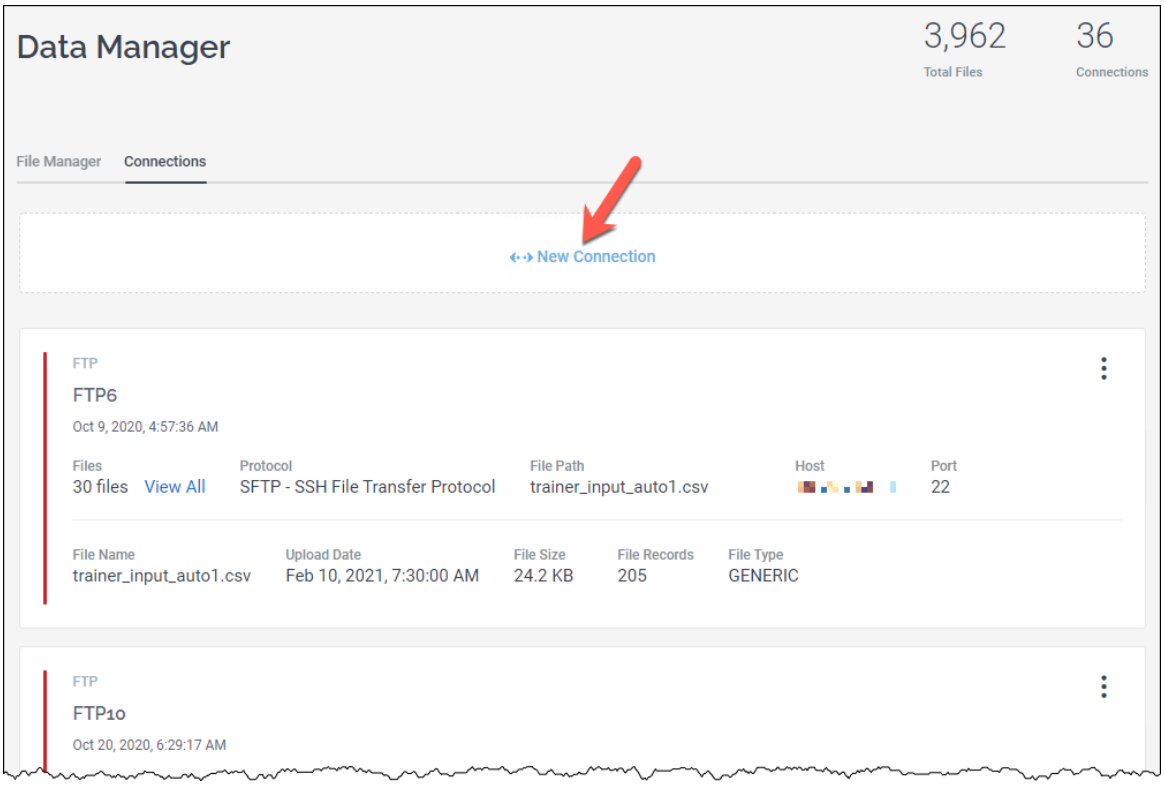
Note

You must create a Data Connection before attaching it to a project.

4.8.4.1 To create a Data Connection

To enable Data Connection functionality, see Deploy AML and link it to a DM system.

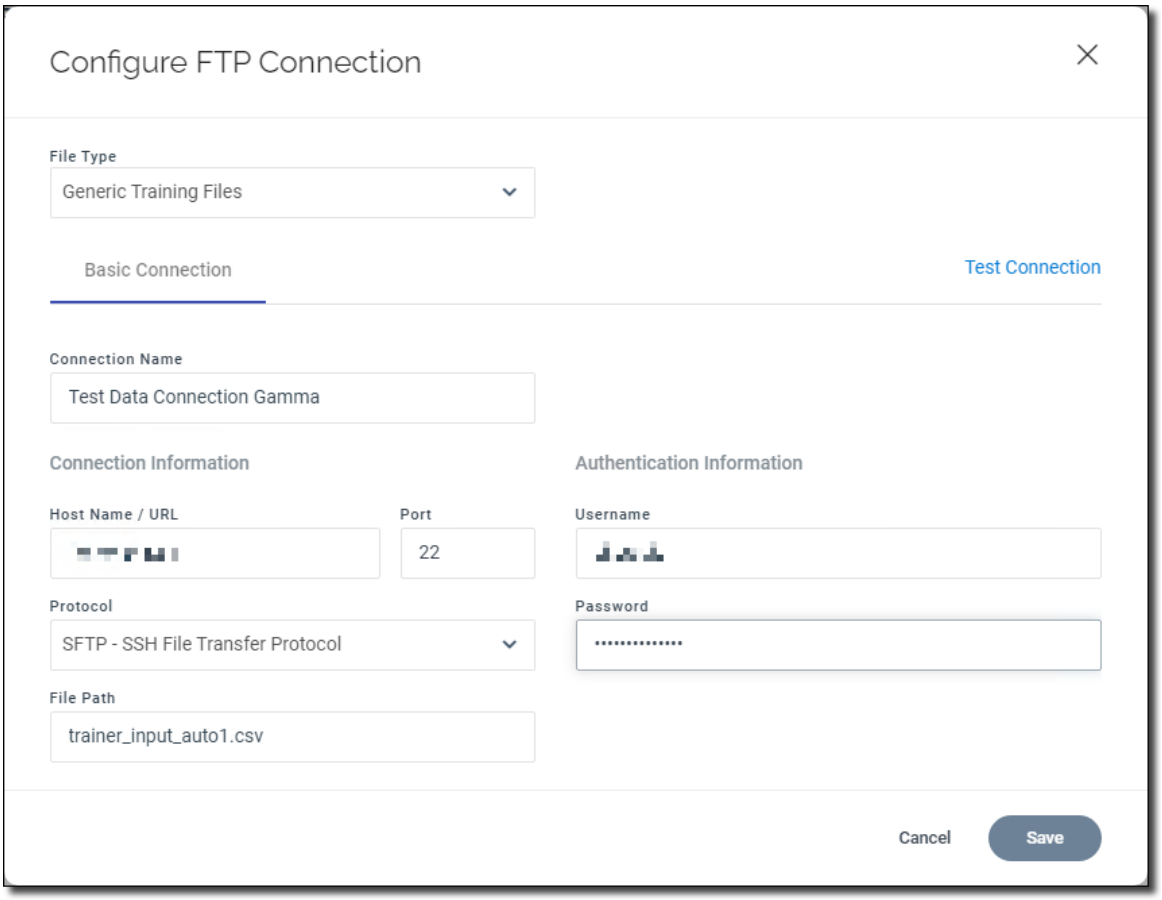
1. Navigate to the **Data Manager** page.
2. Click the **Connections** tab.
3. Click the **New Connection** link.



4. On the **Choose New Connection** dialog, click a connection method (**FTP** or **SQL**) and click **Next**.

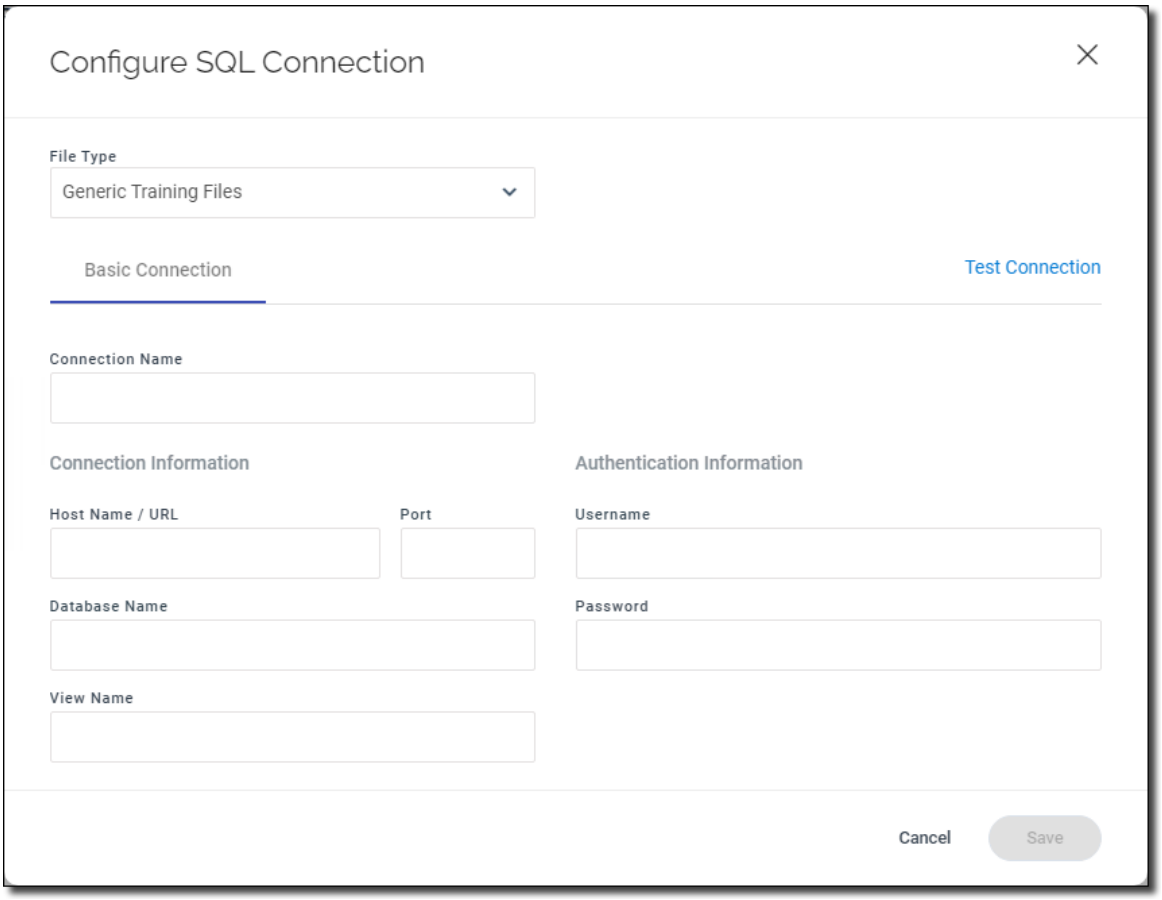
5. If creating an FTP Data Connection:

   - Select a **File Type** (every data file attached to a project must be associated with a file type). See Data set types used by AML 32 for more information.
   - Enter a **Connection Name**.
   - Enter the file's **Connection Information** and **Authentication Information**.



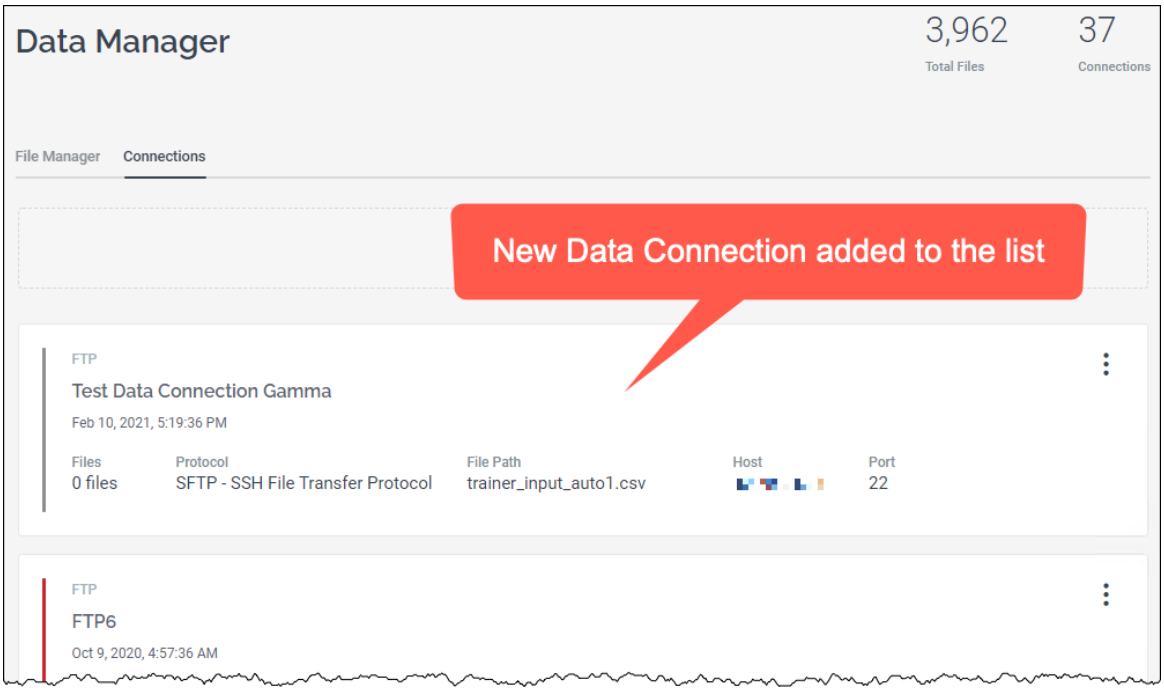
If creating a SQL Data Connection:

- Instead of selecting a **Protocol** and **File Path**, enter a **Database Name** and a **View Name**.



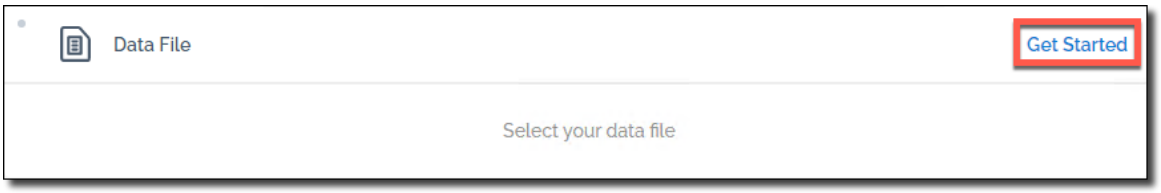
6. Before moving off of the dialog you can make sure the file information is valid by clicking **Test Connection**. If the connection information is valid, a green check mark is displayed:

7. The newly-created Data Connection is added to the list.



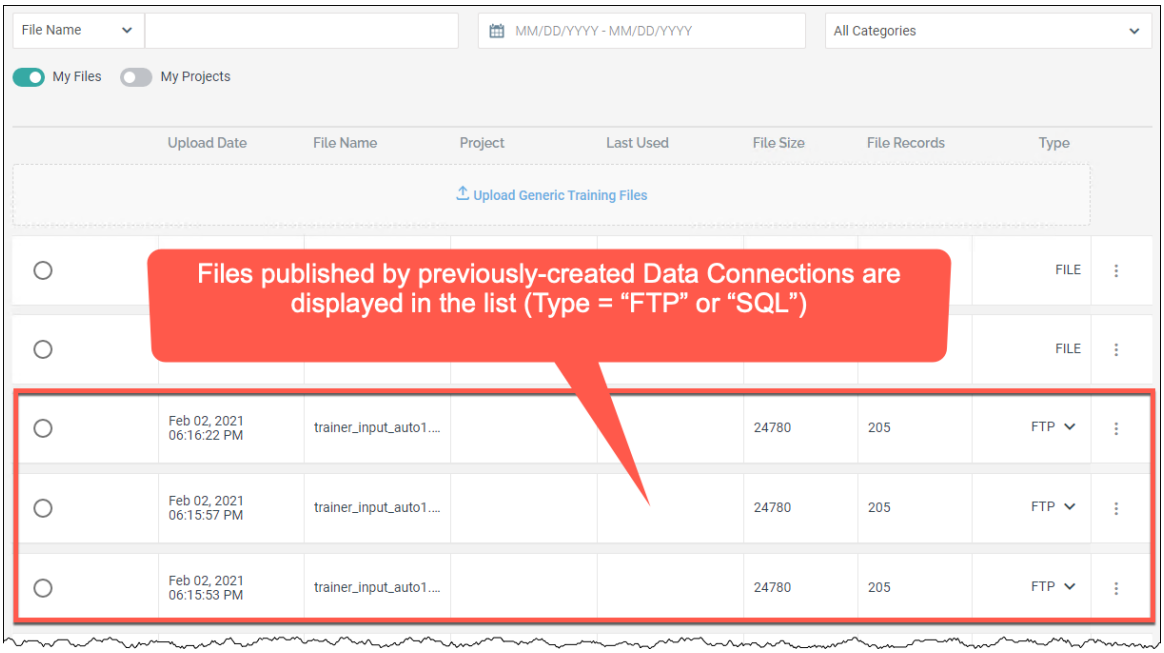
4.8.4.2 To attach a file uploaded by a Data Connection to a project

1. On the **Project Configuration** page, In the **Data File** section, click the **Get Started** link. The File Manager is displayed.

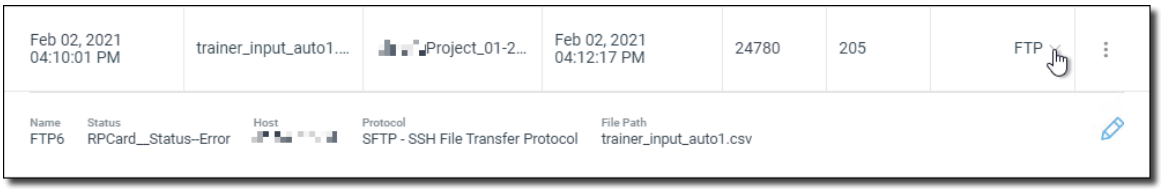


2. The files published by previously-created Data Connections are displayed in the file list (they have a **Type** value of **FTP** or **SQL**), and you can click a file's radio button to select it as your input data file.

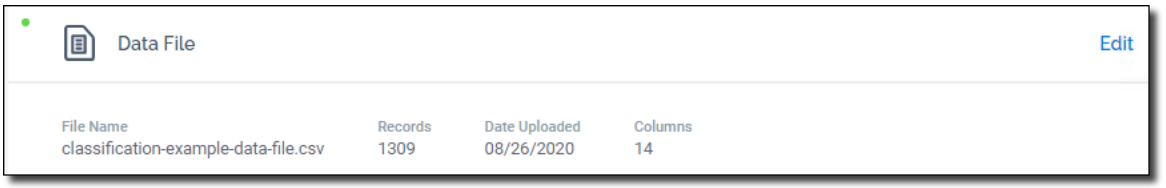
   Note that the files displayed are restricted to the type that can be attached to the project type. See Data set types used by AML 32 for more information.



You can display information about a file's associated Data Connection by clicking the **Type** field (which displays either the value **FTP** or **SQL**):



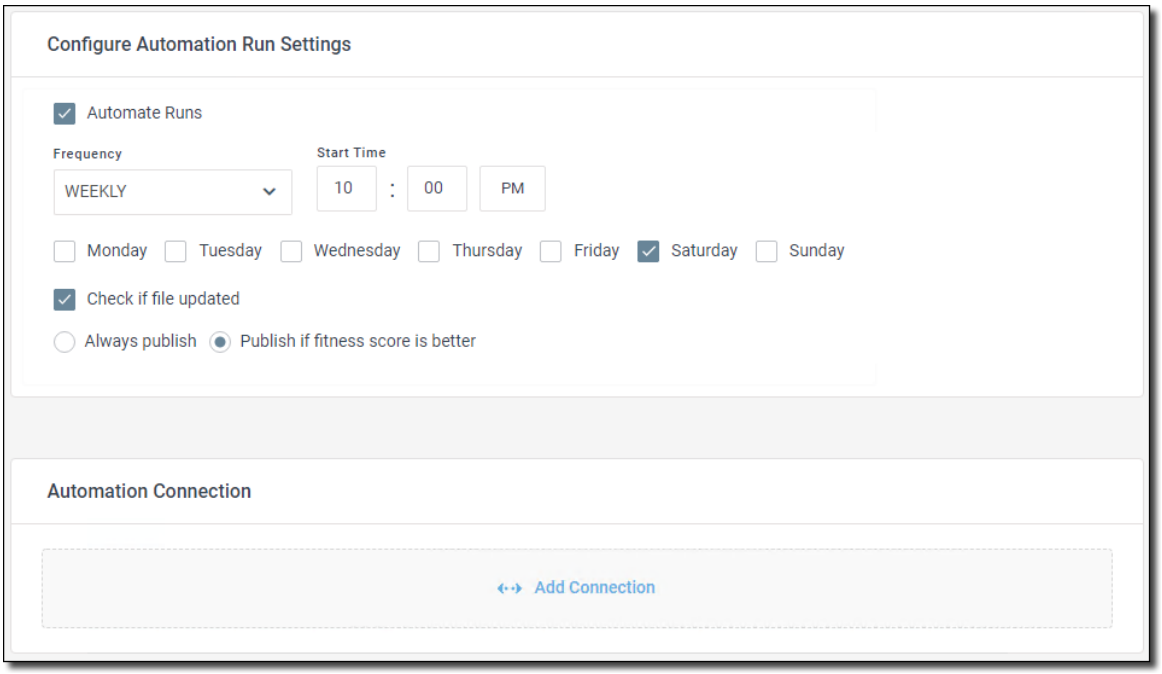
3. At the top right of the page, click **Save** to save your file choice.
4. At the top left of the page, click **Back**.
5. On the **Project Configuration** page, the **Data File** section now displays the Data Connection file you just selected. If the section indicator light is green, the selected data file(s) are valid.



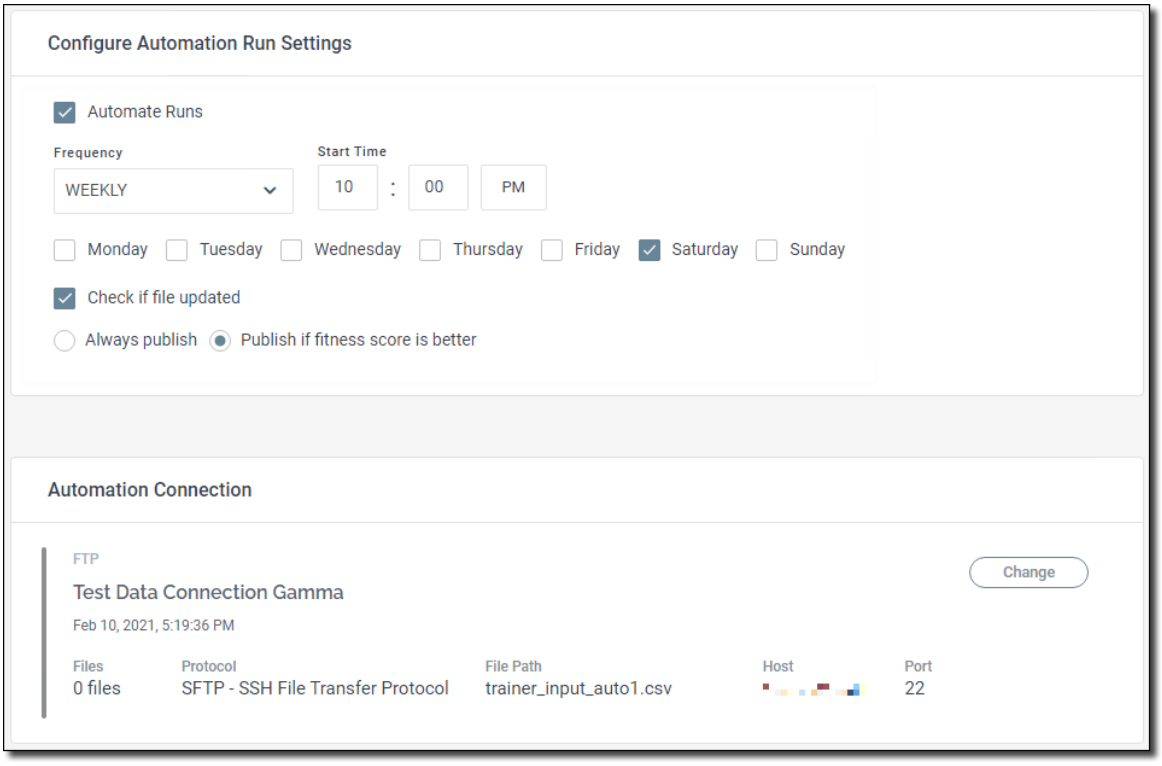
4.8.4.3 To automate file uploads via a Data Connection

After creating a Data Connection you can manually initiate a file upload. You can also automate the process so that the file is uploaded at certain intervals.

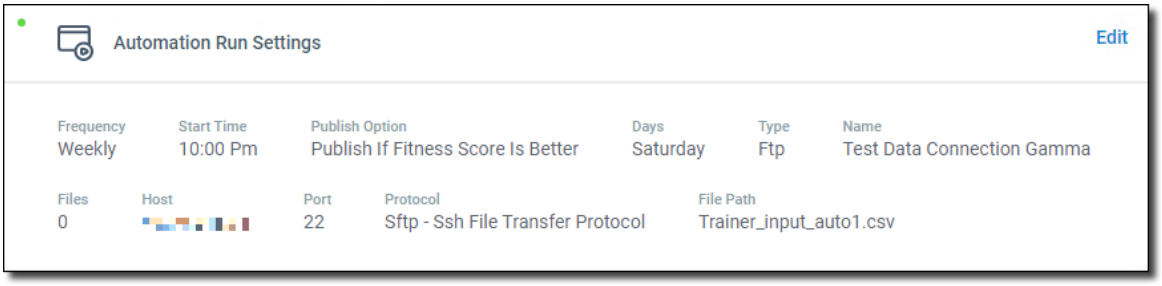
1. On the **Project Configuration** page in the **Automation Run Settings** section, click the **Edit** link.
2. Click the **Automate Runs** box and set the automation criteria.



3. Click **Add Connection**.
4. In the Data Manager, create a new Data Connection and then select it from the list, or select a previously-created Data Connection.
5. Information about the selected Data Connection is displayed on the **Automation Run Settings** page.



6. At the top right of the page, click **Save** to save your file choice.
7. At the top left of the page, click **Back**.
8. On the **Project Configuration** page, the **Automation Run Settings** section now displays the Data Connection file you just selected, along with the associated automation criteria. If the section indicator light is green, the selected data file(s) are valid.



4.9 Choose features

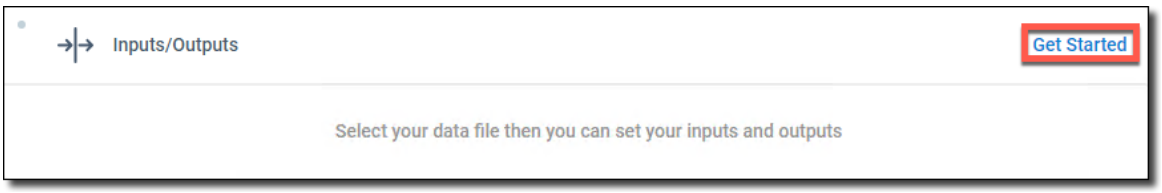
For guidance on selecting output and input features, refer to A brief aside about data file notation and project features 31 and Features 87.

4.9.1 To choose output feature(s)

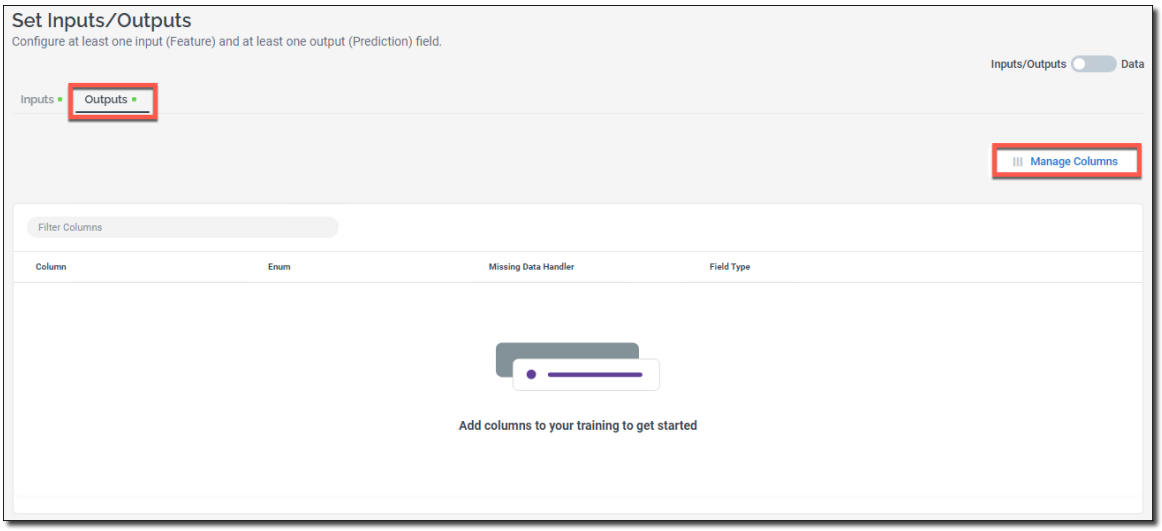
Note:

Regression models are supervised, which means they require the user to specify an output "truth" feature.

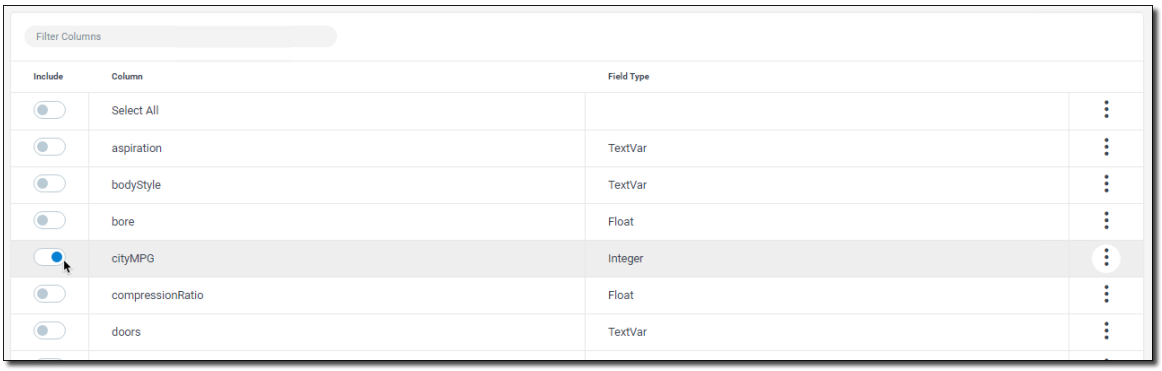
1. On the Project Configuration page's Inputs/Outputs section, click the Get Started link. The Set Inputs/Outputs page is displayed.



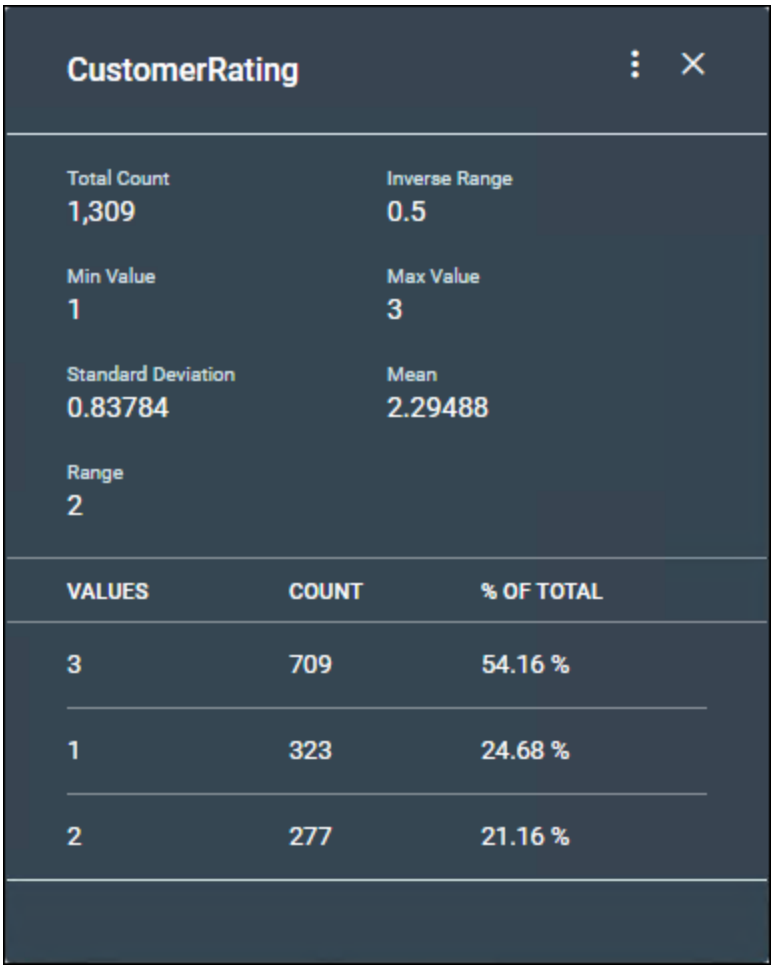
2. Click the **Outputs** tab.



3. Click **Manage Columns**. A table of the file's columns is displayed. Click the radio button or checkbox next to the column(s) you want to designate as output features.



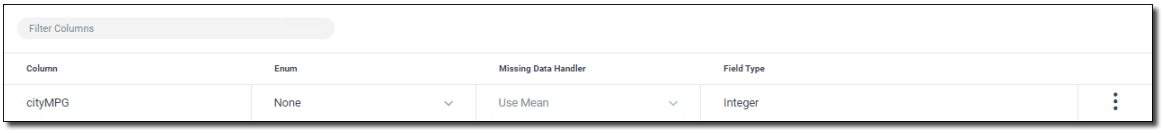
While selecting column(s), you can hover over a column menu on the right side of the table and select **Column Summary**, which displays a dialog of statistics about that column.



Note

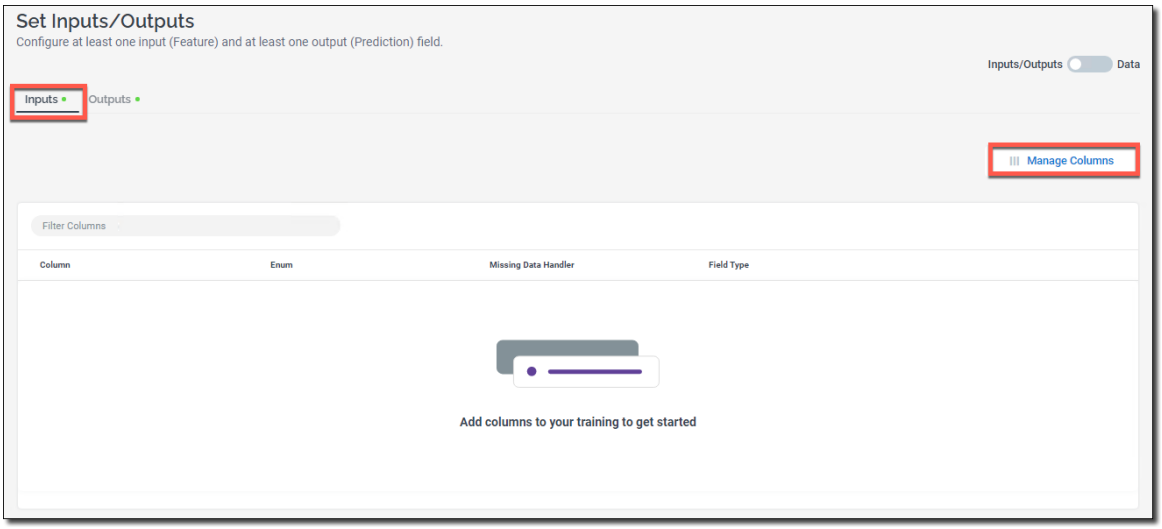
If a model can have only one output field, the UI enforces this by allowing selection via a radio button (which allows only one column of many to be selected). For model types that can have zero or more outputs, the UI allows the fields to be selected via toggles (none, one, or many columns can be selected).

4. Click **Done**. The newly-selected output column is displayed, along with additional information.

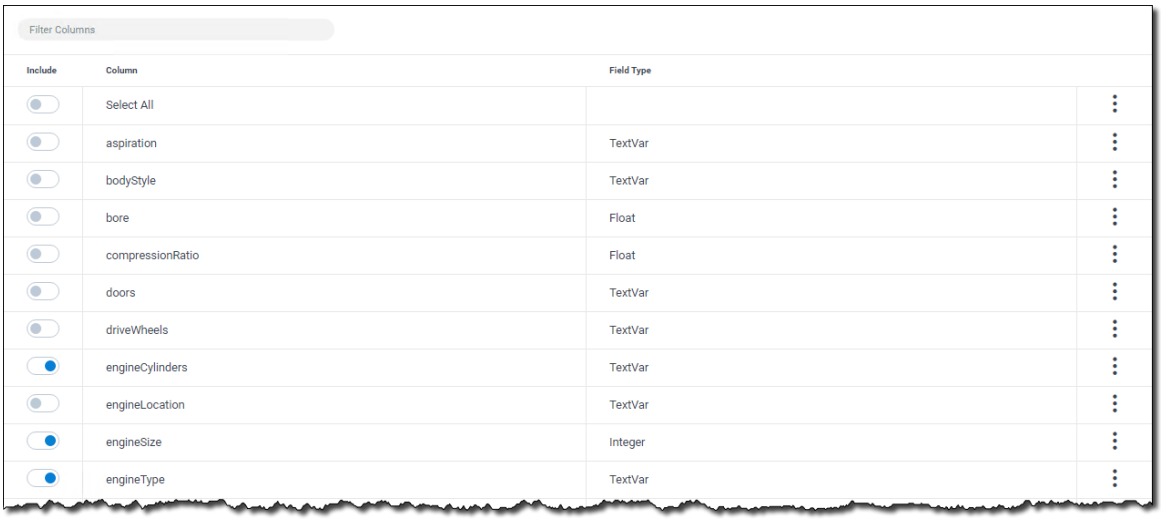


4.9.2 To choose input feature(s) for most model types

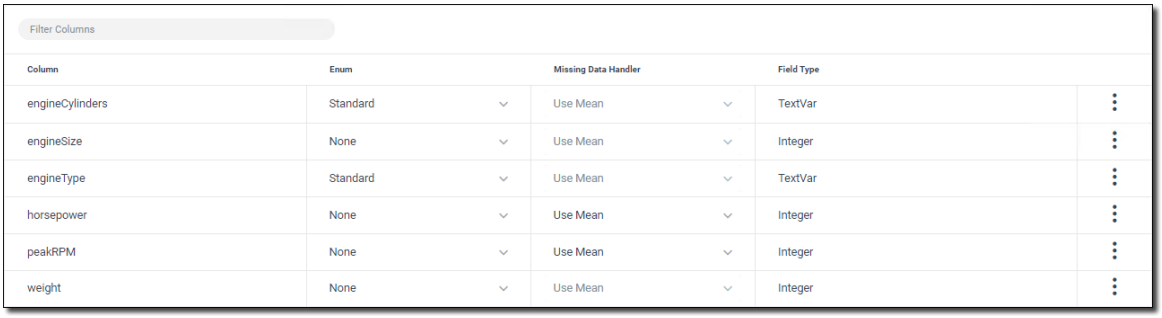
1. On the **Set Inputs/Outputs** page, click the **Inputs** tab.



2. Click **Manage Columns**. A table of the file's columns is displayed. Click the radio button or checkbox next to the column(s) you want to designate as input features.

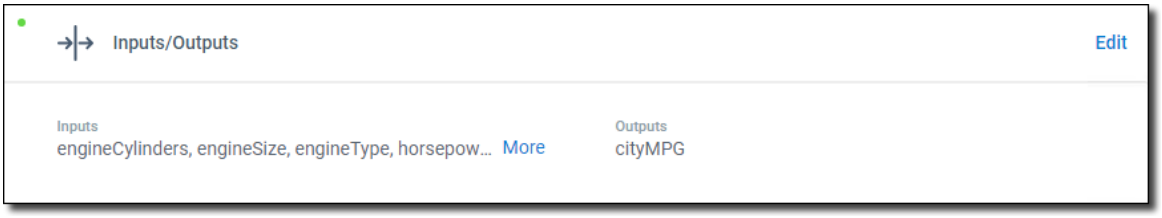


3. Click **Done**. The newly-selected input features are displayed, along with additional information.



4. Click **Save**, then click **Back**.

5. On the **Project Configuration** page, the **Inputs/Outputs** section displays the selected input and output features.

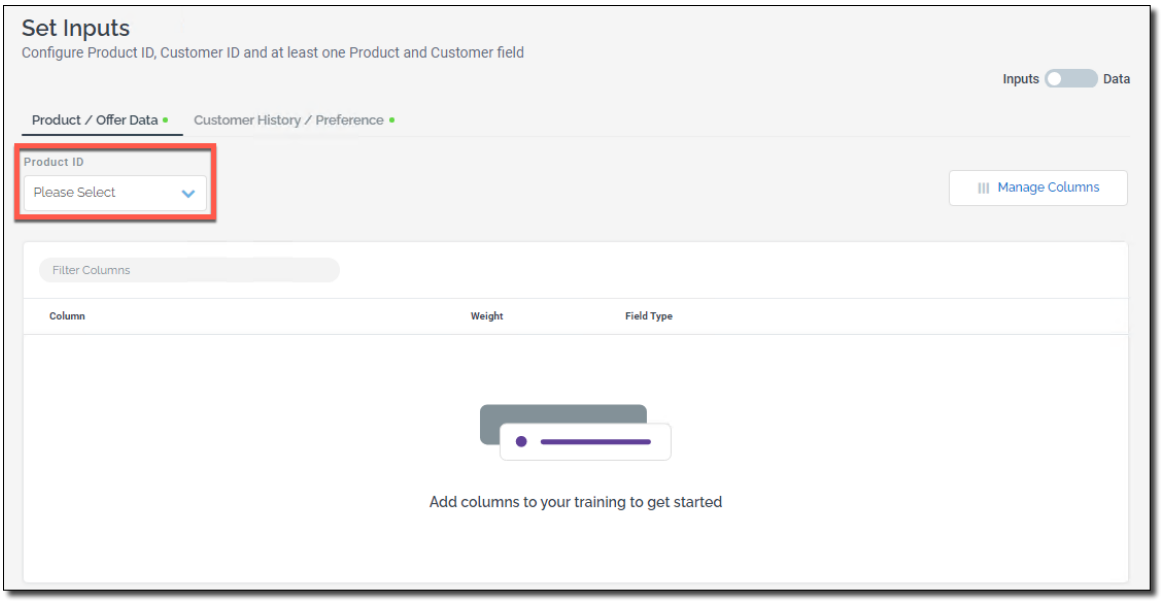


The section indicator light is now green, which means the section settings are valid.

4.9.3 To choose input feature(s) for PR models

The process for choosing input feature(s) for PR models is a bit different than for other model types. Specifically, for PR models, you need to set the Product ID and Customer ID fields, and at least one product and one customer field.

1. On the **Set Inputs** page, the **Product/Offer Data** tab selected by default.

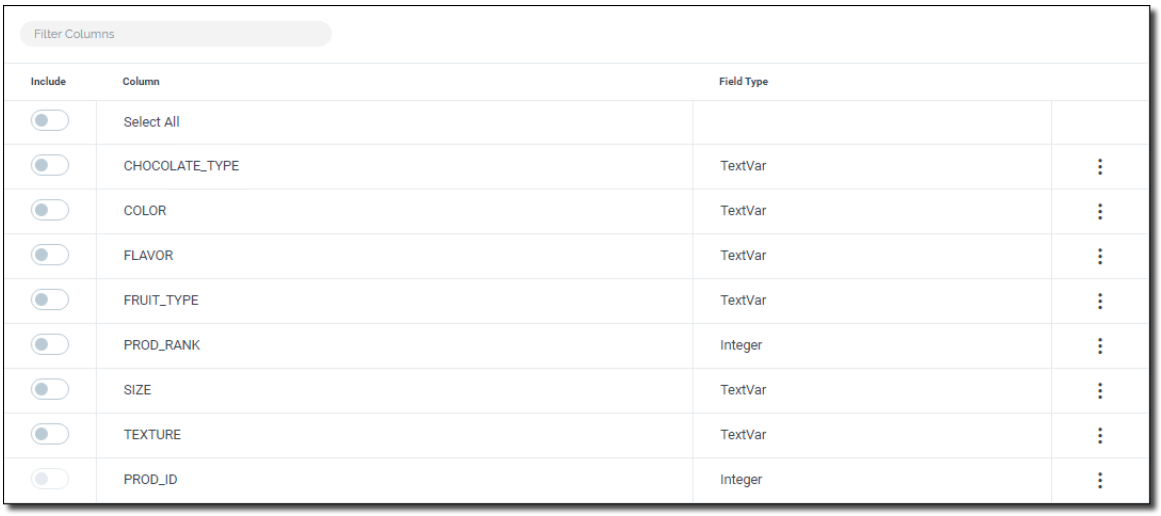


Select the column in which the product ID is stored by clicking the **Product ID** dropdown menu and selecting the appropriate column.

2. One of the columns may contain product rank values (the product rank is optional). These are product ranking values used when there is no customer information available for the model. That is, if the model has no information about this customer, products/offers are suggested in order of this ranking, rather than by random choice.

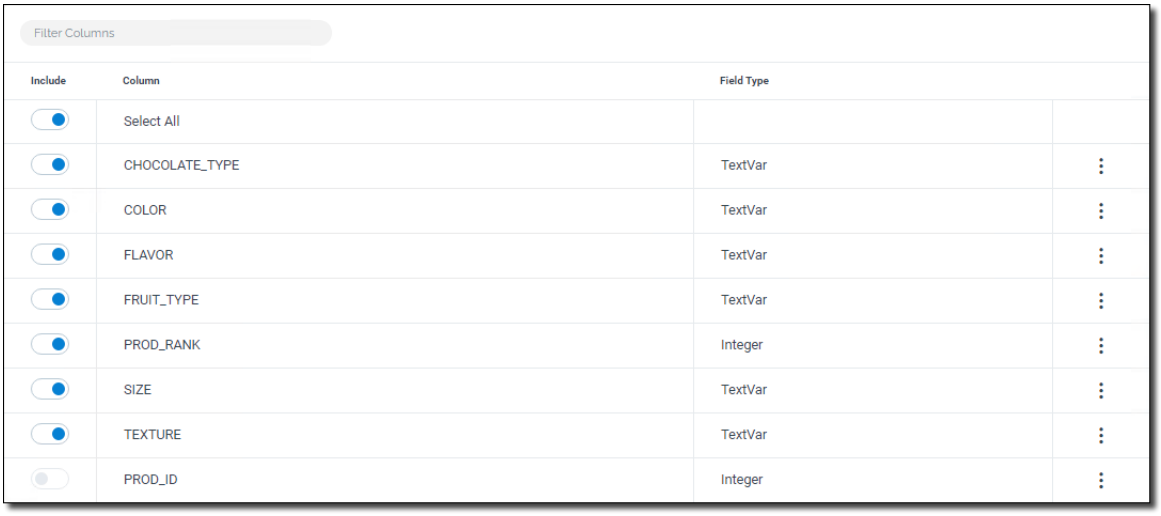
   All the other columns contain preference data, so we want to include the product rank column and all the preference data columns.

   Click **Manage Columns**. A table showing all the file columns is displayed:



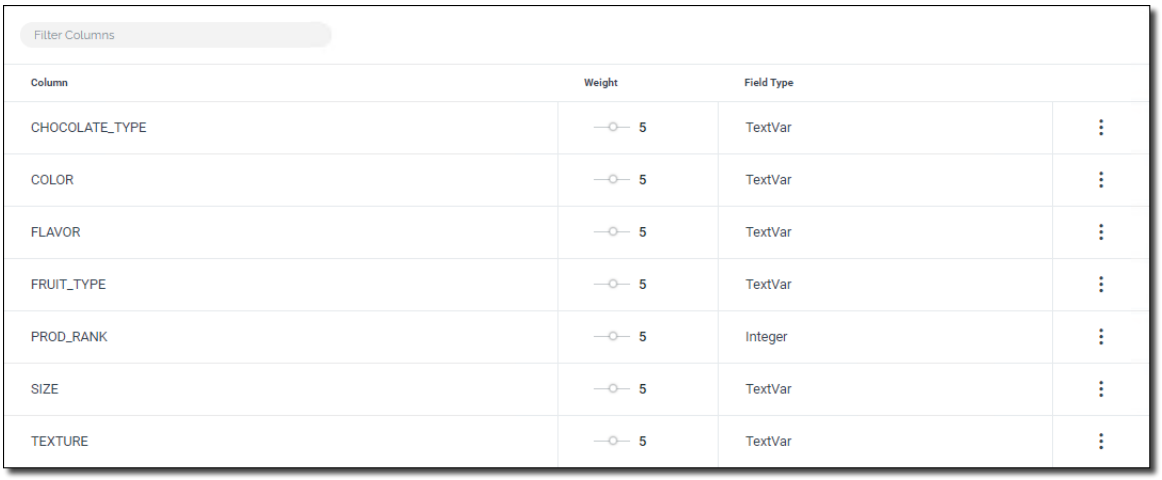
Notice that in this case, the **PROD_ID** column is greyed out (non-selectable). Since we have already designated that column as our product ID field, we cannot also add it as regular product information.

Select each column we want to include as product information:



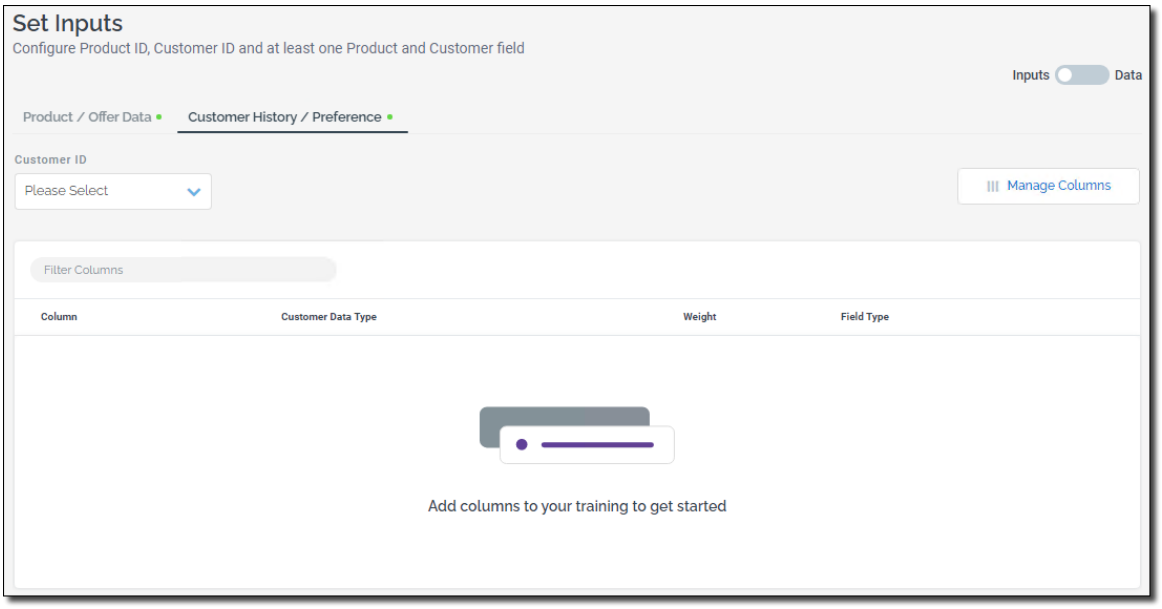
Click **Done**.

3. The **Product/Offer Data** table changes a bit:

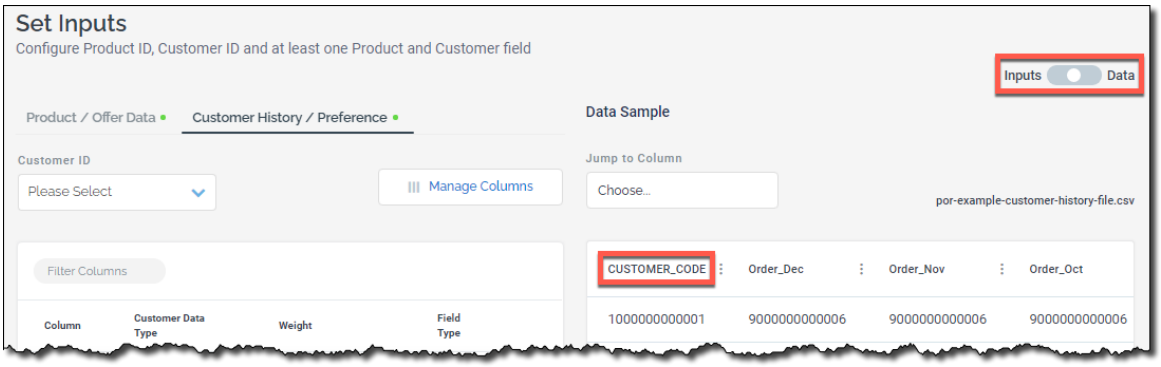


Each column has a weight value assigned to it. The weight value (which can range from 1 to 10) determines the weight of a column when calculating product recommendations. Assigning higher weights to keywords and attributes prioritizes certain criteria (such as latest purchase, most often purchased product, and highest ranking survey response). You can change a column weight by clicking the column's weight value and moving the popup slider.

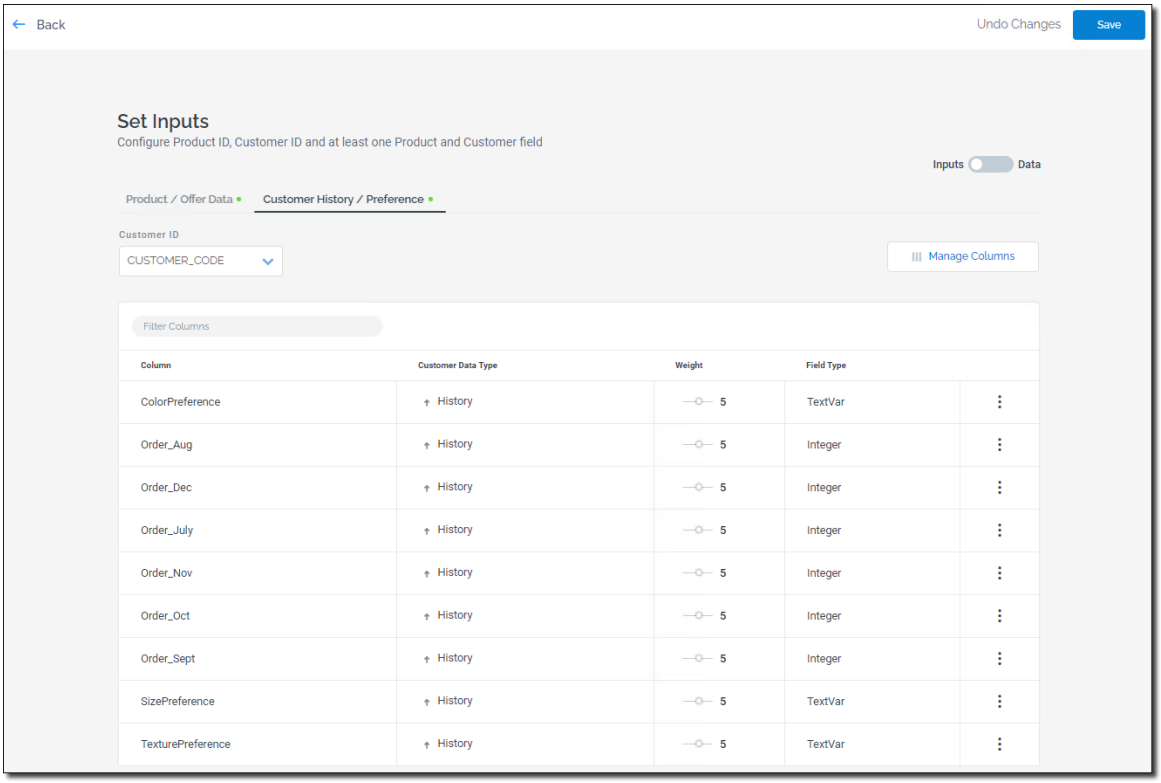
4. Click the **Customer History/Preference** tab.



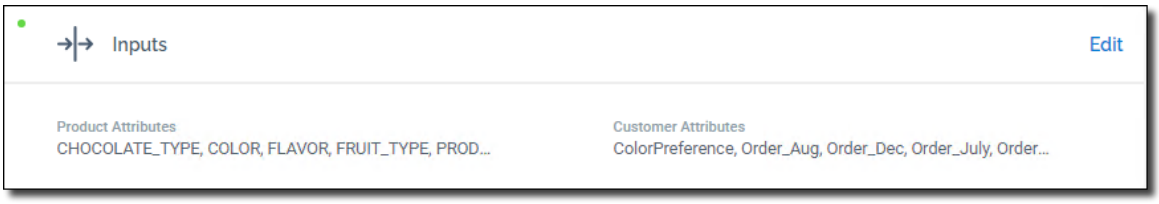
5. We need to figure out which column stores the customer ID value, so view the customer history/preference file's contents by clicking the center of the **Inputs/Data** switch (so that the switch is halfway between **Inputs** and **Data**).



6. In this case, we see that the customer ID value is stored in the `CUSTOMER_CODE` column, so you would click the **Customer ID** menu and select **CUSTOMER_CODE**.
7. Click **Manage Columns**. All the columns (except `CUSTOMER_CODE`) are customer purchase history or customer preference, so we want to include all the columns as model input (except for `CUSTOMER_CODE`, which is unselectable) and click **Done**.
8. All of our file inputs have been added.



9. Click **Save**, then click **Back**.
10. On the **Project Configuration** page, the **Inputs** section displays the selected product and customer attributes.



The section indicator light is now green, which means the section settings are valid.

Note

When you create a new Product Recommender project, you are required to change the settings only in the **Project Configuration** page **Data File**, **Customer Data File**, and **Inputs** sections. The settings in the other sections are optional. For an explanation of the settings in the optional sections, see Project Configuration page settings 108.

4.10 A reminder about modifying project sections

As we mentioned before, after you edit a project's **Data File** and **Inputs/Outputs** sections, AML populates the rest of the project sections with reasonable default settings for most common cases. Editing these other sections is optional, though you can if you want to.

See Project settings and UI 108 for descriptions of the settings in these other sections.

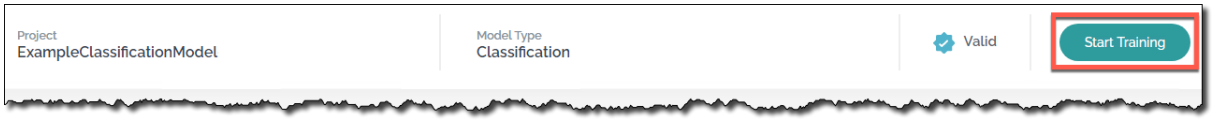
5 Train a project

The following topics describe the project training process.

5.1 To train a project

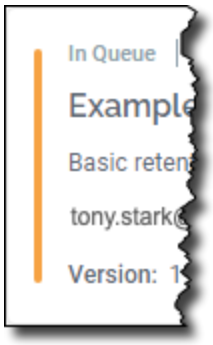
At this point all of the project sections have valid settings, and the project status indicator at the top of the **Project Configuration** page displays the message "Valid". This means the project is ready to train.

At the top of the **Project Configuration** page, click **Start Training**.



When you start training a project, the **Project Manager** page is displayed.

While your project sits in the training queue, the project status bar is orange and the displayed status is **In Queue**.

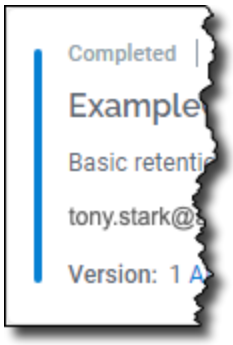


Note

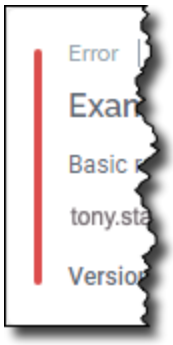
Click the **Project Manager** page **Refresh** button to query the database for any state changes (using the current dashboard filters). It is especially useful when periodically checking the progress of projects in training. We recommend that you use the dashboard **Refresh** button because this updates the dashboard much faster than refreshing the entire page using the browser refresh button.

When your project moves to the head of the training queue, the project status bar turns aqua and the displayed status is **Training**.

When your project has finished training, the project status bar turns blue and displays the status **Completed**.



If there is an error encountered during training, the project status bar turns red and displays the status **Error**.



If you stop project training before completion, the project status bar turns red and displays the status **Canceled**.

After training a project, you can always return to the **Project Configuration** page and view or change the project's settings.

5.2 Train a cloned project

You can clone a trained project and resume training the clone from the original project's completion point. That is to say:

- The original project must have successfully completed training at least once.
- If you clone this project and resume training the clone, the clone uses as the parents of the first generation the *n* top-scoring solutions from the original trained project (where *n* = number of parents in the project).

5.3 Things you can do with a trained project

Note

Redpoint's AML software allows you to create trained models, but you own the models you train.

Publish a project solution

A trained project produces a set of solutions, each with an assigned score. You can publish one of the solutions (the best-scoring one is a good choice) to make it available to the prediction engine. A published solution is called a *model*.

Another way to think of this is publishing a solution makes the solution “live”—puts it in production. Published solutions (models) can be called through the AML API[132] in order to predict expected outputs based on a set of inputs. AML also tracks model metrics (number of API calls and throughput).

Retrain the project

If you get more or different data, you can use that data to retrain an existing project to achieve better correlation. You can even set up a project to auto-retrain itself, without needing the supervision of a data scientist.

Call the model through the AML API

After you’ve deployed a model, you can call it directly through its prediction API[132].

Use the model in other Redpoint products

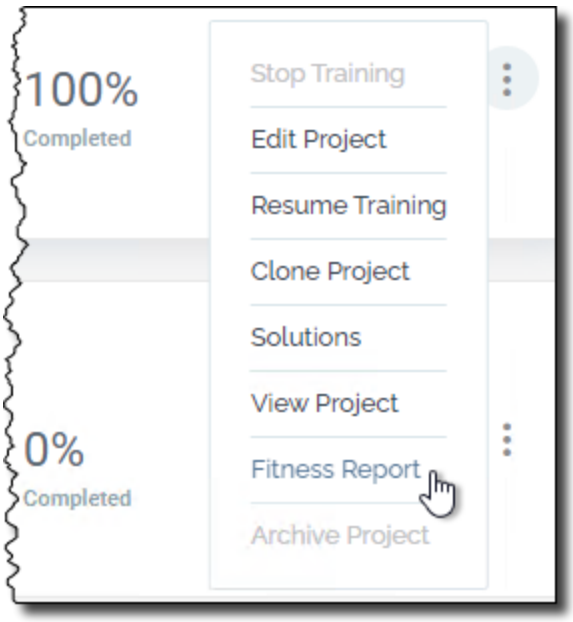
You can use a deployed model in Redpoint RPI. See the RPI documentation for more information.

6

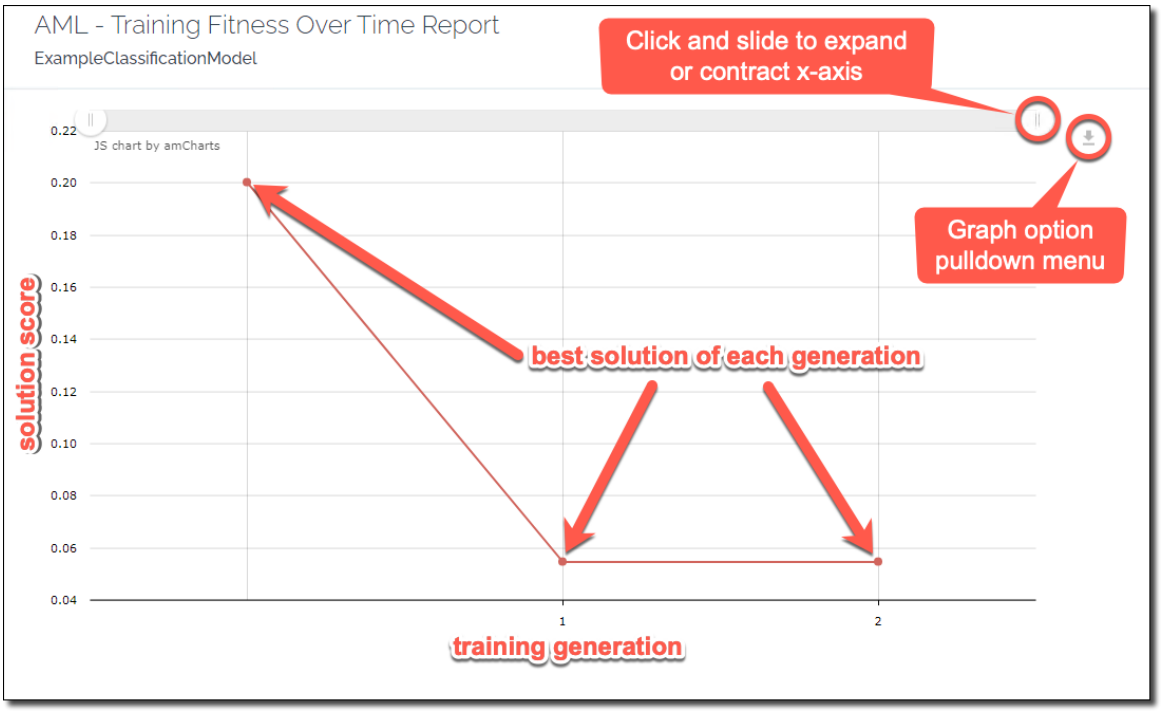
Analyze a project's training results

After training a project, you can view the project's fitness report and review details about the training process.

1. On the **Project Manager** page, in your project section, hover over the project pulldown menu and select **Fitness Report**.



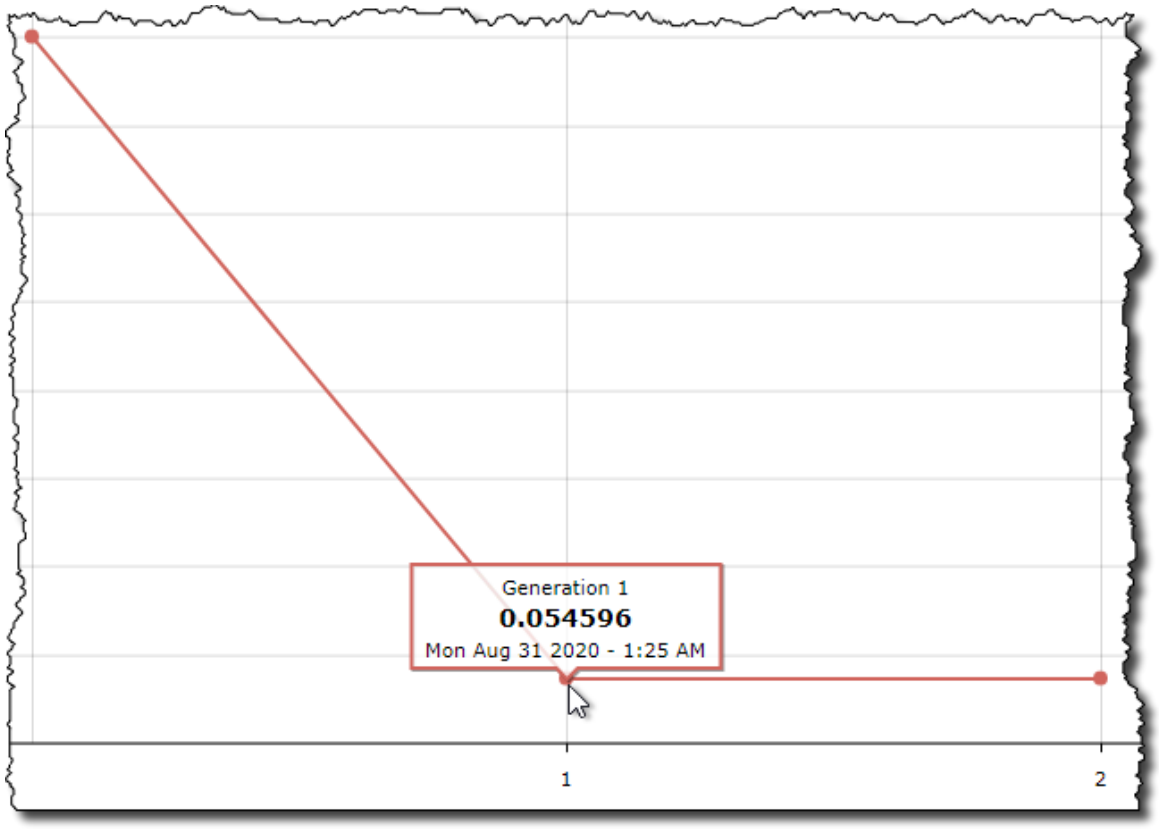
2. The project **Training Fitness over Time** report is displayed. This graph plots the fitness score of the highest-scoring solution of each generation vs. training generation (generation 1, 2, and so on). (Note that in AML, the lower a score value, the better it is).



See Graph options 102 for a description of the options available from the **Graph Option** pulldown menu.

You can hover over one of the graph points to display more information about the solution:

- Generation value.
- Fitness score of the best-scoring solution of that generation.
- Generation timestamp (timestamp of when the solution score was calculated).



The graph tells us:

- For how many generations the project was trained.
- The score of the highest-scoring solution for each generation.
- How the solution scores vary over time. If the scores reach a plateau, for example, that score was the best that could be achieved, and additional training time did not discover a solution with a better score.

3. At the bottom of the graph dialog, click **Close**.

7

Publish a solution

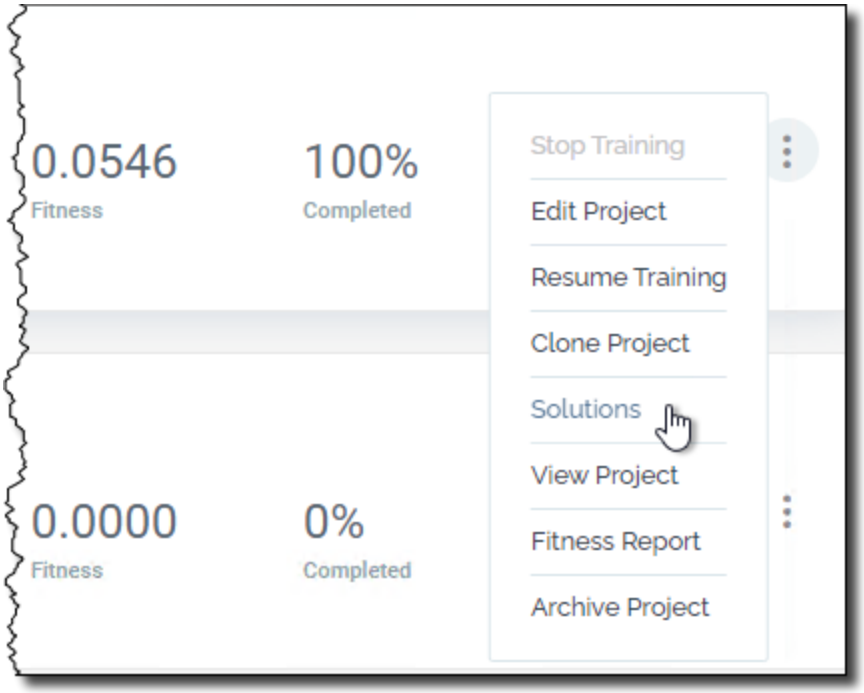
A trained project produces a set of solutions, each with an assigned score. You can publish one of the solutions (the best-scoring one is a good choice) to make it available to the prediction engine. A published solution is called a *model*.

Another way to think of this is publishing a solution makes the solution "live"—puts it in production. Published solutions (models) can be called through the AML API 132 in order to predict expected outputs based on a set of inputs. AML also tracks model metrics (number of API calls and throughput).

7.1

To publish a solution

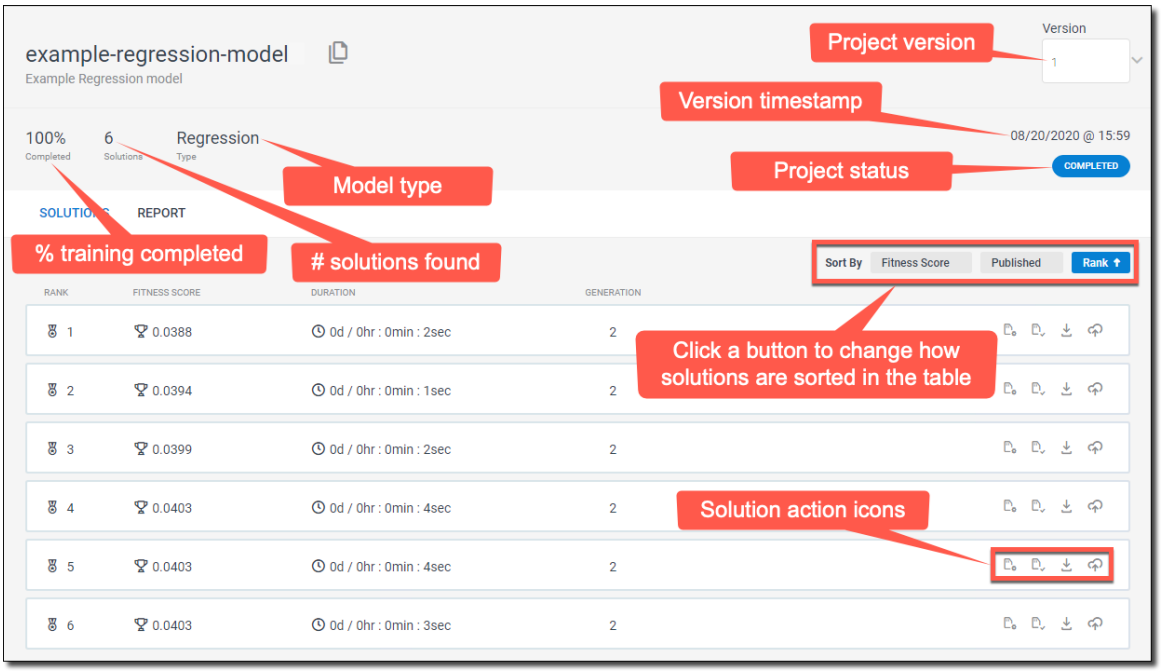
1. On the **Project Manager** page, in your project section, hover over the project pulldown menu and select **Solutions**.



2. The **Solutions** page displays a table of the top-ranked solutions for a given project. You can use the information on this page to help you decide which solution is best for your needs.

   Note:

   Unlike other model types, a Product Recommender model creates only one solution.

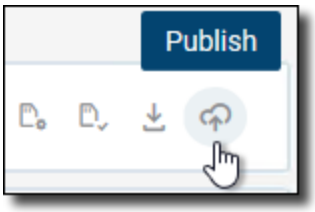


In this context, solutions with a higher rank have lower error scores (as calculated per whatever metric/measure is used). That is, models with better fitness (lower error scores) have a higher ranking (1, 2, 3, ...) than those models with lower fitness (higher error scores). Some modeling tools define fitness in terms of another measure (for example, R-Squared) in which higher values are usually “better”.

You can sort the solutions by **Fitness Score**, whether or not the solution is **Published**, and calculated **Rank**.

For a detailed explanation of this page, see Solutions page 104.

3. In the row of the solution you want to publish, click the **Publish** icon:



4. After a solution is published, the icon turns green.

After you’ve published a solution, you can call it directly through its prediction API. See AML API 132 for more information.

You can also use a published solution in Redpoint RPI. See the RPI documentation for more information.

8 Classification model example

In AML, one or more data models are wrapped in a *project*. Let's build a simple project, consisting of a single data model.

In this example, we will build a Classification model that trains with the `classification-example-data-file.csv` data set. Classification models use existing data points and variables to classify future outcomes based on the existing training data. Both defined inputs and (correlated) outputs are used to train the model.

Classification models are typically used to predict discrete-value outputs (for example, whether a record belongs in class 1, class 2, or class 3).

8.1 To build an example Classification model

Download the example project data file to your machine

1. Download this .zip file to your local machine:

   https://rpgcdnfiles.blob.core.windows.net/aml/aml-model-example-data-files.zip

2. Open the .zip file and save `classification-example-data-file.csv` to your local machine.

Create a new project

Create a new Classification model project 19 with the following values:

- Project name: "ExampleClassificationModel".
- Project description: "Basic retention/attrition model" (optional).
- Use the default category value "General".

Upload your data

Upload the file 32 `classification-example-data-file.csv` to AML and add it as your project input data file.

Select the output feature 42

The purpose of this project is to predict (based on things we already know about a customer) whether the customer will be retained or not (0 = no, 1 = yes). So the output feature is:

- `Retention`

Select the input features 44

For this project, the input features are:

- CustomerCaptureMethod
- CustomerContactCode
- CustomerRating
- DaysSinceLastPurchase
- HouseholdChildren
- ImmediateRelatives
- LatestPurchaseItemID
- LatestPurchasePrice
- LoyaltyMember
- RegionCode

Train the project

Train the project 50 using the default values for the other project sections.

Analyze the project results

You can view the project's Fitness report 53 and **Solutions page** 56.

9

Clustering model example

Cluster/segment analysis (aka *clustering/segmentation*) is the task of grouping a set of objects in such a way that objects in the same group (called a cluster or segment) are more similar (in some sense) to each other than to those in other groups (clusters/segments).

In this example, we will use as input a file of vehicle metrics and insurance loss/risk data. We will then create a Clustering project using a subset of this data as input features, and train the model to see how the vehicles are clustered/segmented. The project output assigns a cluster/segment number to each vehicle. The generated solutions can be used to do things such as derive insurance pricing, create offers to present to existing and potential customers, and so on.

9.1

To build an example Clustering model

Download the example project data file to your machine

1. Download this .zip file to your local machine:

   https://rpgcdnfiles.blob.core.windows.net/aml/aml-model-example-data-files.zip

2. Open the .zip file and save `clustering-example-data-file.csv` to your local machine.

Create a new project

Create a new Clustering model project [19] with the following values:

- Project name: “ExampleClusteringModel”.
- Project description: “Basic Clustering model” (optional).
- Use the default category value "General".

Upload your data

Upload the file [32] `clustering-example-data-file.csv` to AML and add it as your project input data file.

Select the input features [44]

Note:

Some cluster measure functions are semi-supervised, which means they require the user to specify an output “truth” feature. See Semi-supervised Clustering measures functions [129] for a list of these functions.

The cluster measure function that we are using for this example (Davies-Bouldin) is unsupervised, so it does not require an output “truth” feature. See Unsupervised Clustering measure functions [129] for a list of all the cluster measure functions that are unsupervised.

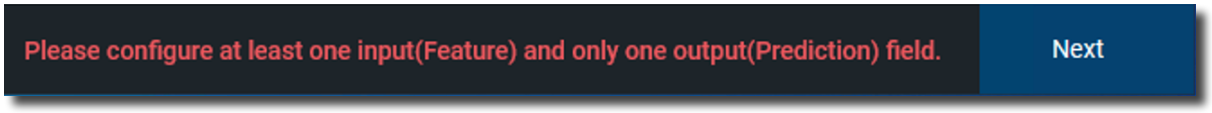
For this project, the input features are:

- `doors`
- `driveWheels`
- `engineCylinders`
- `highwayMPG`
- `horsepower`
- `symbolingRisk`

- `weight`
- `wheelBase`

Note:

If in the future you do choose a semi-supervised cluster measure function, make sure to add an output “truth” feature. If you neglect to do so, the following error will be displayed:



Train the project

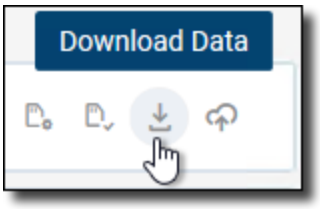
Train the project 50 using the default values for the other project sections.

Analyze the project results

In addition to viewing the project's Fitness report 53 and **Solutions page** 56, you can view the calculated data clusters.

The calculated data cluster information is contained in the generated solutions. Let’s take a look at the data cluster information for the highest-scoring solution.

1. Navigate to the project **Solutions** page. In the solution 1 row, click the **Download Data** icon:



2. Open or save the output data file.
3. Open the file in a text editor:

clusterModel field

This field contains a number of sub-fields of interest:

- `algorithmType`—The algorithm that was selected as “best”.
- `numClusters`—The number of generated clusters.

clusterCentroids field

This field contains a list of the generated clusters. Each cluster begins with the `dataSampleIndex` and `uniqueID` fields, which contain the cluster ID.

dataSampleIndex (cluster) field

This field contains a number of sub-fields that describe an individual cluster:

- `centroidDistances`—Distances from the centroid point to the boundary in each dimension.
- `centroidPts`—Describes the N-dimensional center points for the cluster. For example, assuming three dimensions (X,Y, and Z), a center point (centroid) might be {3.95, 1, 424.7} where the centroid center is located at x = 3.95, y = 1, z = 424.7. You can substitute descriptive field names for X, Y, and Z for clarity. Note that since some features are not numeric (for example, text, enumerated, T/F) the “center” points in those dimensions represent mapped values (for example, 0 = "red", 1 = "blue", 2 = "yellow"). For user clarity, these should be remapped back to their original features.
- `dataIndices`—The record numbers (indices) assigned to the cluster.

10

Regression model example

This sample project creates a Regression model that predicts the city MPG value for a given vehicle, based on input features such as number of engine cylinders, engine size, and engine horsepower.

10.1 To build an example Regression model

Download the example project data file to your machine

1. Download this .zip file to your local machine:

   https://rpgcdnfiles.blob.core.windows.net/aml/aml-model-example-data-files.zip

2. Open the .zip file and save `regression-example-data-file.csv` to your local machine.

Create a new project

Create a new Regression model project [19] with the following values:

- Project name: “ExampleRegressionModel”.
- Project description: “Basic Regression model” (optional).
- Use the default category value "General".

Upload your data

Upload the file [32] `regression-example-data-file.csv` to AML and add it as your project input data file.

Select the output feature [42]

The purpose of this project is to predict (using input features such as a vehicle’s number of engine cylinders, engine size, and engine horsepower) a vehicle’s city MPG fuel consumption.

For this project, the output feature is:

- `cityMPG`

Select the input features [44]

For this project, the input features are:

- `engineCylinders`
- `engineSize`
- `engineType`
- `horsepower`
- `peakRPM`
- `weight`

Train the project

Train the project 50 using the default values for the other project sections.

Analyze the project results

You can view the project's Fitness report 53 and **Solutions page** 56.

11

Product Recommender model example

A Product Recommender (PR) model compares customer purchase history or other customer preference attributes to a set of product attributes, and generates one or more product recommendations for each customer.

Specifically, this project outputs a specified number of customer ID/product ID pairs for each unique customer ID, along with rank and match scores for each pair.

11.1

To build an example Product Recommender model

Download the example project data files to your machine

1. Download this .zip file to your local machine:

   https://rpgcdnfiles.blob.core.windows.net/aml/aml-model-example-data-files.zip

2. Open the .zip file and save the following files to your local machine:

   - `por-example-product-data-file.csv` (product data file)

     The product data file contains product attributes, which are keywords associated with each product ID. This file must contain a unique ID field and one or more attribute fields.

   - `por-example-customer-history-file.csv` (customer data file)

     The customer data file contains customer history data, which are product IDs associated with a customer ID. Customer history and customer preference data can both be embedded in this file. This file must contain a unique ID field and one or more attribute fields.

Create a new project

Create a new Product Recommender model project 19 with the following values:

- Project name: “ExamplePRModel”.
- Project description: “Basic Product Recommender model” (optional).

- Use the default category value "General".

Upload your data

1. In the **Product Data File** section, upload the file 32 `por-example-product-data-file.csv` to AML and add it as your project input data file.
2. In the **Customer Data File** section, upload the file `por-example-customer-history-file.csv` to AML and add it as your project customer data file.

Select the input features 46

For this project:

- The product ID column is `PROD_ID`.
- The product rank column is `PROD_RANK`.
- Product information columns:
  - `CHOCOLATE_TYPE`
  - `COLOR`
  - `FLAVOR`
  - `FRUIT_TYPE`
  - `PROD_RANK`
  - `SIZE`
  - `TEXTURE`
- The customer ID column is `CUSTOMER_CODE`.
- Customer purchase history or customer preference columns:
  - `ColorPreference`
  - `Order_Aug`
  - `Order_Dec`
  - `Order_July`
  - `Order_Nov`
  - `Order_Oct`
  - `Order_Sept`
  - `SizePreference`
  - `TexturePreference`

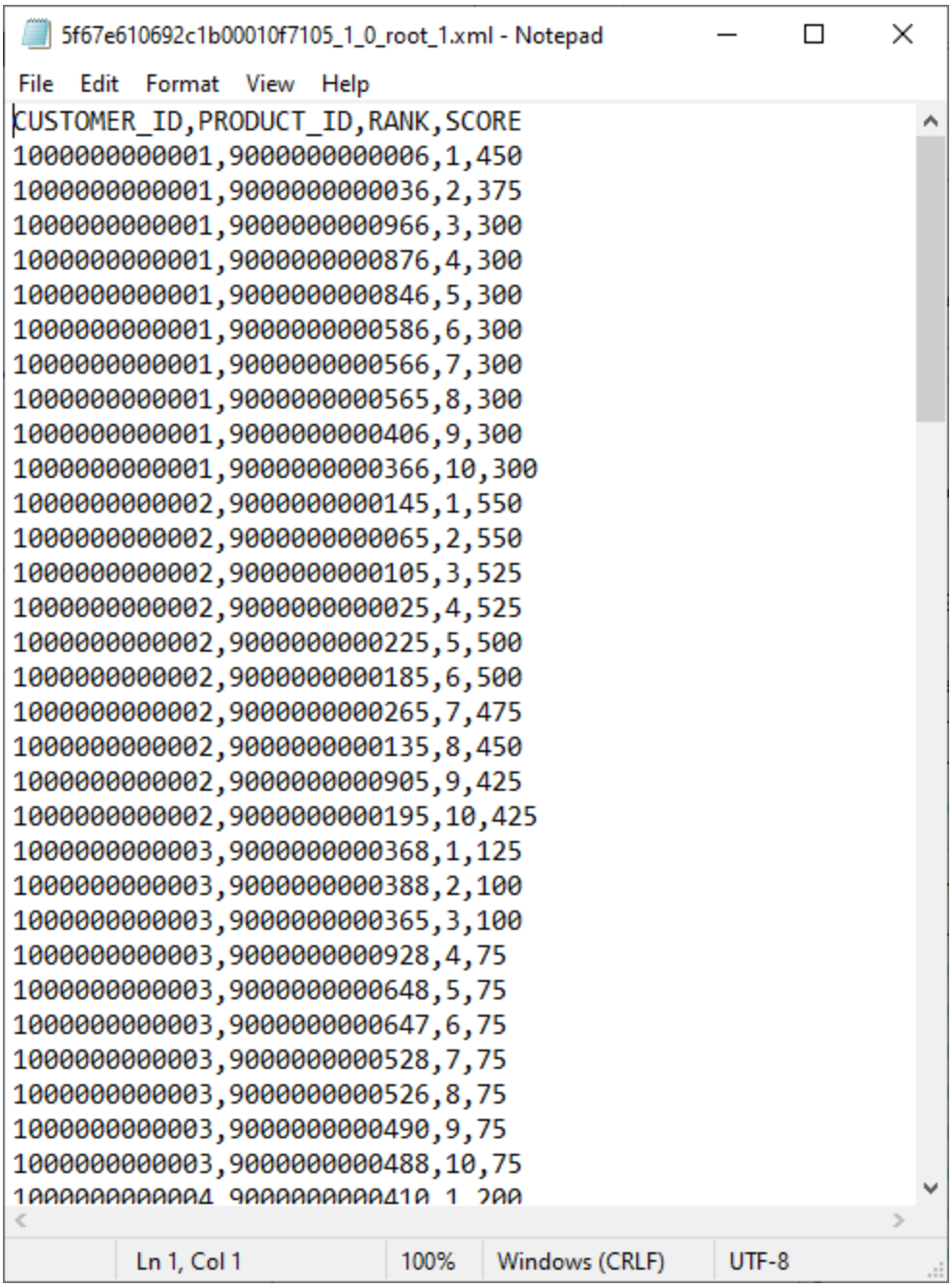
Train the project

Train the project 50 using the default values for the other project sections.

Analyze the project results

In addition to viewing the project's Fitness report 53 and **Solutions page** 56, you can view the customer recommendations.

1. On the **Project Manager** page, hover over your project section's pulldown menu and select **Solutions**. Note that unlike other model types, a Product Recommender model creates only one solution.
2. Since this solution is a concrete list of product recommendations per customer, let's look at the actual list. On the right side of the solution row, click the **Download Data** icon.
3. Open or save the output data file.
4. Open the file in a text editor:



Each customer has ten product recommendations (because that was the default value, though we could have chosen another value). Each product recommendation is identified in the `PRODUCT_ID` field and ranked (you guessed it!) by the `RANK` field. Each product recommendation is accompanied by its calculated `SCORE`.

For customer 1000000000001, product 9000000000006 is the first recommendation (with a score of 450), product 9000000000036 is the second recommendation (with a score of 375), and product 9000000000966 is the third recommendation (with a score of 300).

Customer 100000000009 has score values of 0 for its recommendations because we had no preference or history data for that customer. Because we used default ranking data in the model, the product recommendations are given by the ranking data, and there is no associated score.

12 Reference

The topics in this section are targeted for more advanced AML users.

12.1 Installation and configuration

The following sections explain how to install and configure AML on your own infrastructure.

12.1.1 Deploy AML on a single Linux VM

AML works best when deployed onto a multi-node cluster (either a Swarm or Kubernetes cluster). In this installation, we are installing on a single-node cluster.

12.1.1.1 Minimum system requirements

With all of AML installed on one node, the minimum system requirements are:

OS

- We recommend Ubuntu Linux 16.04 or 18.04
  - Compatible with any OS supported by Docker here:

    https://success.mirantis.com/article/compatibility-matrix
- 16 GB memory
- 4 CPUs

Ports

If this Linux VM is hosted externally, make sure the following ports are accessible:

Port	Description
22	For SSH access to install app
8983	https port for web browser access to main app
8993	https port for web browser access to admin app

Port	Description
8930	Socket IO service used to push real-time notifications back to web browser

API developers will likely want access to the Swagger UI. Open these ports for Swagger UI access:

Port	Description
8913	Machine Learning endpoints
8903	Framework endpoints (user management, authentication, and so on)

12.1.1.2 Installation prerequisites

Obtain a Docker account

AML is packaged as a set of Docker container images that are archived in a Redpoint Global Docker repository:

https://cloud.docker.com/u/redpointglobal/repository/list

You will need an account on `cloud.docker.com` to access these images.

Request a license from Redpoint support

From Redpoint support, request a Mercury-AML license. If this is for a time-limited POC, be sure to specify in the request for how long the key should remain valid.

Support will send you an activation key that looks something like:

```
AKNK070404G1H1J18M1Z331P6KGYFCUTN2
```

Save this key for later entry into the License Management page of the Admin app.

Create an account on cloud.docker.com, get access to AML

Once you have created a `cloud.docker.com` account, contact your Redpoint representative to grant your account access to AML.

Verify that Docker has been installed

In your OS shell, run this command:

```
docker --version
```

Most Linux VMs require that you execute Docker commands as sudo, so you may need to enter the command as such:

```
sudo docker --version
```

The displayed Docker version should be 18.06.0-ce (or higher).

Install or update Docker (if necessary)

1. If Docker (or the correct version of Docker) is not installed, execute the installer at:

   https://docs.docker.com/install/linux/docker-ce/ubuntu/

2. Make sure you can log in to the Docker repo by running this in your OS shell:

   ```
   sudo docker login -u docker_username -p docker_password
   ```

   If you get an error that looks like:

   ```
   Error saving credentials: error storing credentials
   ```

   You need to install additional packages in your Linux VM. Do this by running the following command in your OS shell:

   ```
   sudo apt install gnupg2 pass
   ```

Initialize Docker swarm mode

Initialize this single-node cluster by running this command in your OS shell:

```
sudo docker swarm init
```

If you have a system with multiple active IP addresses, you may get an error:

```
Error response from daemon: could not choose an IP address to advertise since this
system has multiple addresses on different interfaces (10.0.2.15 on eth0 and
192.168.99.100 on eth1) - specify one with --advertise-addr
```

In this case, rerun Docker swarm and choose one of the addresses:

```
sudo docker swarm init --advertise-addr 10.0.2.15
```

If you must specify an address, you will reuse this address (instead of using "localhost") when connecting to the AML addresses in the section Install AML locally 71.

If linking AML with DM for Data Connections (optional)

If you want to link AML with DM in order to enable DM-based Data Connections, you need to install the following software:

Software	Version (minimum)
Redpoint Data Management	9.1.3.2422

12.1.1.3 Install AML locally

Download AML deployment package

This package file includes:

- The primary service deployment script (`deploy_service_to_node.sh`) that generates the final Docker compose file and deploys to the target system(s).
- The set of templates (`<service>.template`) for the various Docker service configurations. These templates are combined during deployment to create the final Docker compose file (that is needed to deploy the containers).

To download an installer to your local Linux system, use one of the following Linux tools from your OS shell (where *x.y* is the version information):

```
curl https://cdn.redpointglobal.com./aml/redpoint_dcc-v6.x.y.tgz -o redpoint_dcc-v6.x.y.tgz
```

or

```
wget https://cdn.redpointglobal.com./aml/redpoint_dcc-v6.x.y.tgz -O redpoint_dcc-v6.x.y.tgz
```

Contact Data Management support or pre-sales support for download information.

Note

The `deploy_service_to_node.sh` script displays each Docker command as it executes them, which allows you to see how Docker is utilized (and may aid you in building new scripts). The deploy script has several functions that assist with managing AML services.

Unpack the zip/tarball file

On Linux, you can unpack the .tgz file using this command in your OS shell:

```
tar -zxf redpoint_dcc-vX.tgz
```

Unpacking the zip/tarball creates the directory:

```
redpoint_dcc
```

Create a default administrator account

The deployment script creates a default administrator account named "admin", with the password ".RedPoint123".

Use the following deployment options to change the name or password for the default administrator account:

`--admin-user` *username*

`--admin-pass` *password*

If you prefer that no administrator account be created during deployment, use the following deployment option:

```
--create-admin-user no
```

SMTP configuration

If you want to enable user password resets via email, you need to specify the SMTP host with the following option in the `deploy_service_to_node.sh` command:

```
--smtp-host smtp.host.com
```

Run the deployment script

Notes

You must use quotation marks around `db_connection_string` and around any passwords because special characters in either string cause an error when processing the Bash command line.

If using MongoDB, it must have authorization enabled and the connection string must contain user/password for an admin user with admin access for managing/creating tables (for example, `--mongodb mongodb://admin:redpoint@mongohost:27017"`).

1. Change the directory to `redpoint_dcc`:

   ```
   cd /path/to/redpoint_dcc
   ```

2. Deploy AML to the local host:

   ```
   sudo ./deploy_service_to_node.sh --cmd install --type dcc --type trainer --type
   predictor --registry-user docker_username --registry-pass "docker_password" --
   constraint manager
   ```

   The *docker_username* string is the username chosen when you create an account on cloud.docker.com 69.

   Note

   If this is a multi-node swarm cluster and not a single-node VM, you can omit `--constraint manager`. AML will deploy across all nodes in the cluster, taking advantage of the entire swarm.

   The `--constraint manager` flag deploys the Mercury services only onto the swarm manager node (which is the only node you have in a single node deployment).

   If you are using an external MongoDB instance or a CosmosDB instance (recommended for production or POC deployments), add the MongoDB/CosmosDB connection string options to the deployment command:

   If MongoDB instance:

   ```
   --mongodb "db_connection_string"
   ```

   If CosmosDB instance:

```
--mongodb "db_connection_string" --dbtype COSMOS_PARTITIONED
```

Note

If you do not provide an external database connection, Mercury automatically creates a local MongoDB instance to use. This option is great for short-term tests or POC, but the local MongoDB instance does not have high availability or backups. If anything happens to the VM where this is deployed, data from the local database will be lost. We highly recommend using an external database connection.

If you want to link AML with DM in order to enable DM-based Data Connections, add the following options to the deployment command:

```
--rpdm-enabled {yes|no} --rpdm-host host/ip --rpdm-user username --rpdm-pass
password \
--rpdm-temp-dir directory --rpdm-ml-uri uri
```

where:

`--rpdm-enabled`	`{yes|no}`	Enable DM for data connections (default=no)
`--rpdm-host`	*`host/ip`*	Hostname or IP for DM server
`--rpdm-user`	*`username`*	Username for DM server
`--rpdm-pass`	*`password`*	Password for DM server
`--rpdm-temp-dir`	*`directory`*	Temporary directory on DM server
`--rpdm-ml-uri`	*`uri`*	URI for ML-ServicesAPI accessible by DM server

3. Press the Enter key to continue the deployment (or add `--force` to the shell command to skip the prompt).

Depending on your network speed, it may take ten minutes or more to download all Docker containers.

Optional commands

Display all commands allowed by the deploy script:

```
sudo ./deploy_service_to_node.sh --help
```

Display the service status:

```
sudo ./deploy_service_to_node.sh --cmd status
```

Delete the service:

```
sudo ./deploy_service_to_node.sh --cmd delete
```

Check status

View the status of the local docker deployment with this command:

```
sudo docker service ls
```

This command displays all services that have been started. When everything is running, the `replicas` column should show 1 out of 1 (1/1) replicas running.

Before proceeding to the next step, verify that all services are executing.

(Optional) Add additional app admin or system admin users

When an instance of AML is deployed, it has only two associated user accounts:

- An app admin user with the username "admin".
- A system admin user with the username "system".

Additional app admin and system admin users can be created only through the AML API. The following topics provide these instructions:

- Add a system admin account 133 (that is, a user who has the ability to log in to the AML Admin page).
- Add an app admin user account 132 (that is, a user who has the ability to configure roles and add other users to the AML app).

Log in to AML's admin UI to enable the AML app for the global client

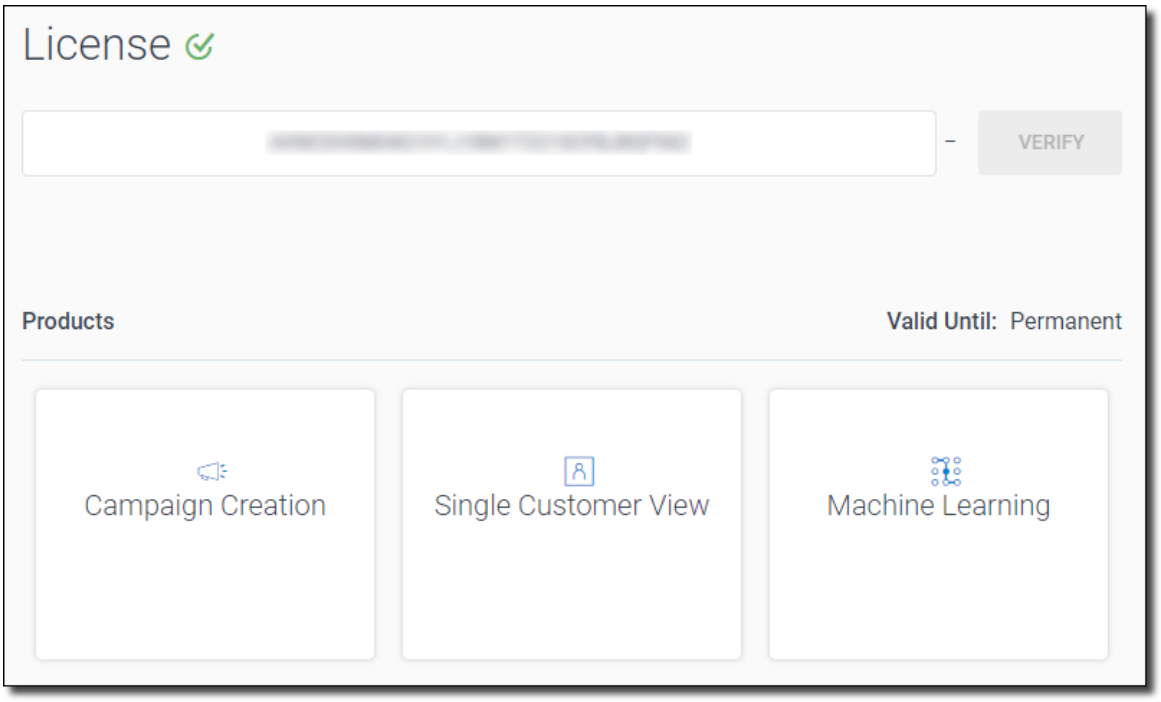
You will use the admin account created during the deployment to log in to the Web Admin UI. The default admin account is named "admin" and has the password ".RedPoint123".

1. If you specified the `--admin-user` and `--admin-pass` arguments to override the admin account defaults when calling the deployment script, use those account values to log in:

   `https://linux-hostname:8993/#/Login`

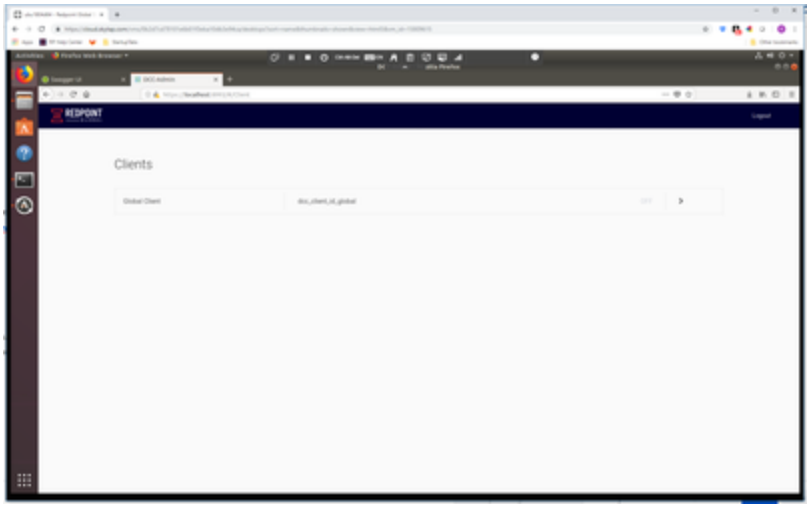
   `http://linux-hostname:8990/#/Login`

2. When you log in for the first time, the License entry page is displayed.
3. Enter the license key given to you by Redpoint Support.
4. Click **Verify**. You should see AML appear in the products list:



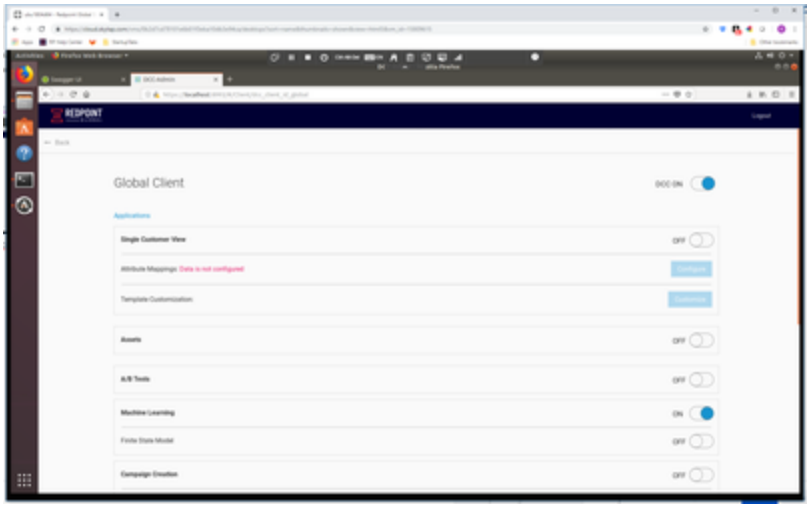
Enable the AML app for the global client

1. After verifying the license, click the Redpoint icon on the upper left-hand part of the screen to return to the Clients page.



2. Click the arrow for the Global client.
3. On the next page, enable:
   - **DCC**
   - **Enable Machine Learning**
   - **Disable Single Customer View**

Your configuration should look like this:



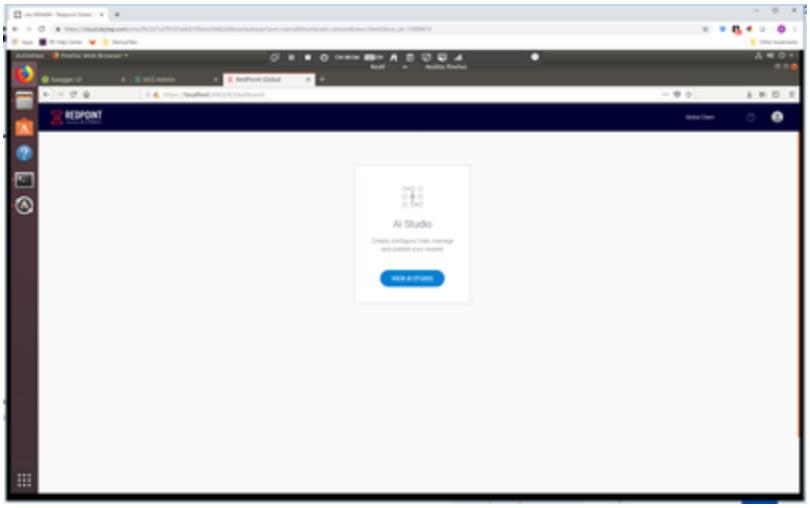
Log in to the AML app

1. Log in to the AML app using the default "admin" account (or any other app admin account you have created):

   ```
   https://linux-hostname:8983/#/Login
   ```

   ```
   http://linux-hostname:8980/#/Login
   ```

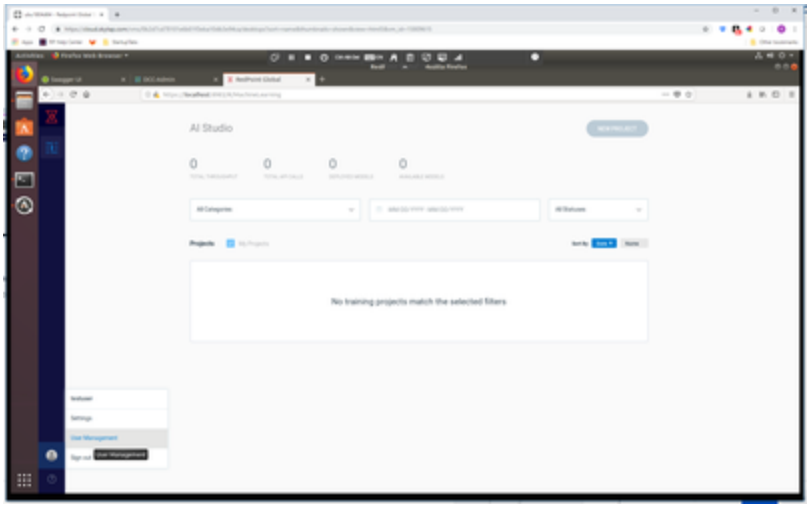
2. Click the **AML** card.



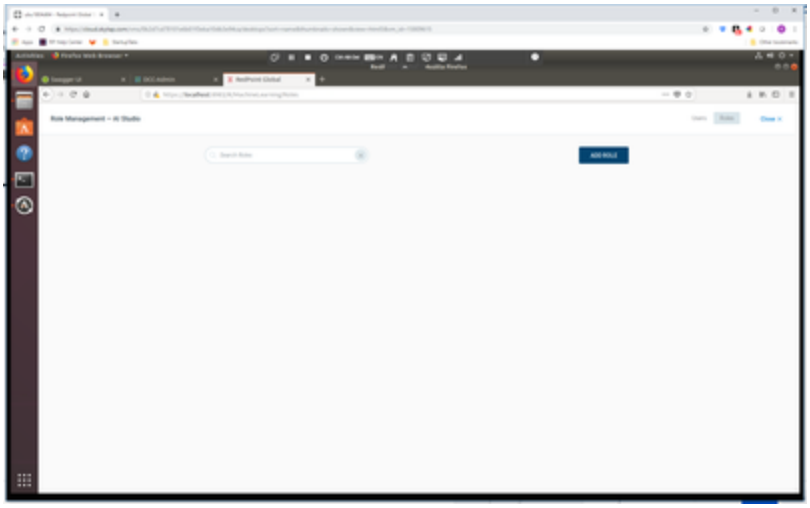
Create a role (with all permissions enabled)

As the first app administrator, you must define at least one role for the app and assign permissions to the role.

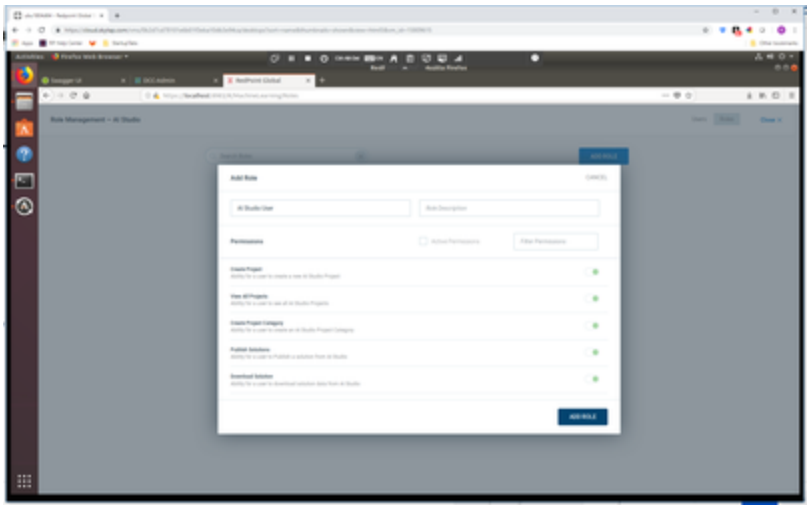
1. Select **User Management** in the lower left-hand corner corner of the page.



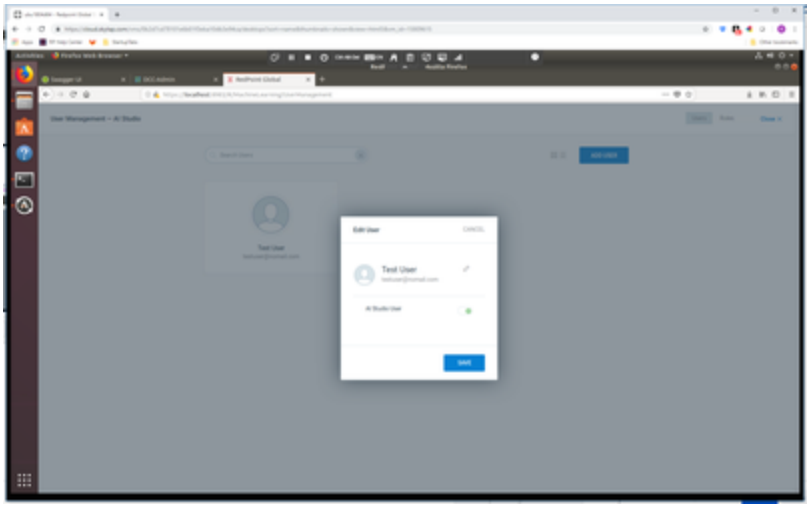
2. Click **Roles - Add Role**.



3. Give the role a name and enable all permissions.



4. Go to **Users** and edit the **admin** account.
5. Enable the new role for the admin user.



6. To activate the new roles, log out and then log back in with the "admin" account. You now have full access to AML's functionality.

Getting ready for production use

After verifying that the AML app is operational, we recommend that you add a unique user account 82 for every user that will be using the AML app. We do not recommend that users share accounts.

Notes

- If there will be several users accessing the system, configure a SMTP host (this supports user-initiated password resets).
- We recommend changing the password for the initial "system" account.

12.2 User management

This section explains:

- User types and roles used in AML
- How to create AML user roles and user accounts

12.2.1 User types

The categories of users in AML are:

User

A general AML user with normal permissions. When installing your instance of AML, Redpoint does not create any users of this type.

App admin

This user type has the ability to do user administration tasks such as creating new users, assigning roles to users, and so on. An app admin can also do anything a general user can do. At app installation, Redpoint creates one app admin with the username "admin".

System admin

This user type is given permissions to install and configure AML and other Redpoint web apps. At app installation, Redpoint creates one system admin with the username "system".

12.2.2 User roles

In AML, a user is assigned permission to do things through *roles*. Each role is assigned one or more atomic permissions, such as the ability to create a project or publish a solution.

For example, you could create a role named “Clone any Project” with the assigned permissions **Create Project** and **View all Projects**. Then you could assign that role to one or more users.

12.2.3 Creating roles and users

This section describes the web pages displayed when logged in as an app admin.

The AML installation process does not create general users or create preconfigured user roles. If you are logging in to AML for the first time as an app admin, you will need to:

- Create user roles
- Create new user accounts (and in the process, assign roles to each new user account)

12.2.3.1 Add a user role

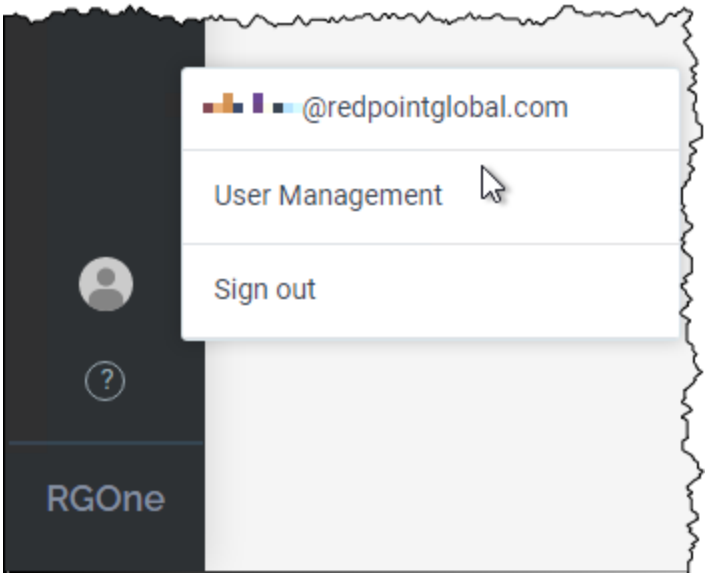
This section gives the procedure for adding a user role.

To add a user role

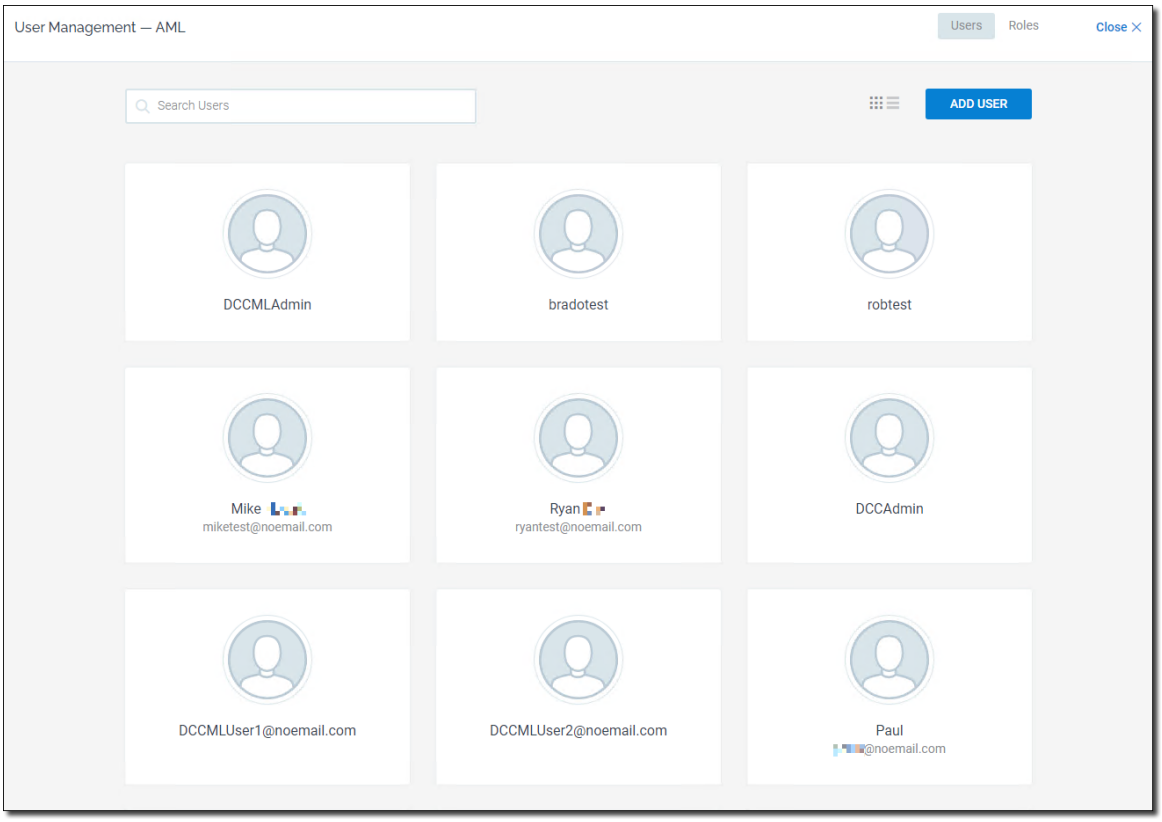
1. Sign in to AML as an app admin.
2. On the left-side navigation menu, click the user icon:

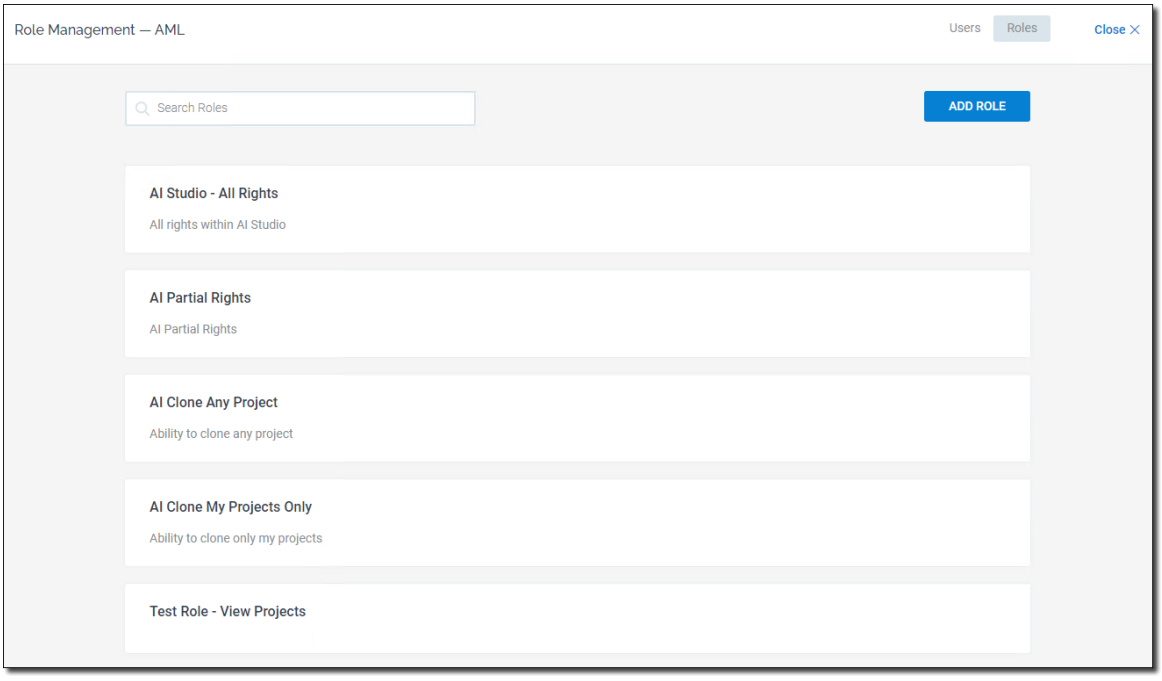
   On the popup menu, select **User Management**.

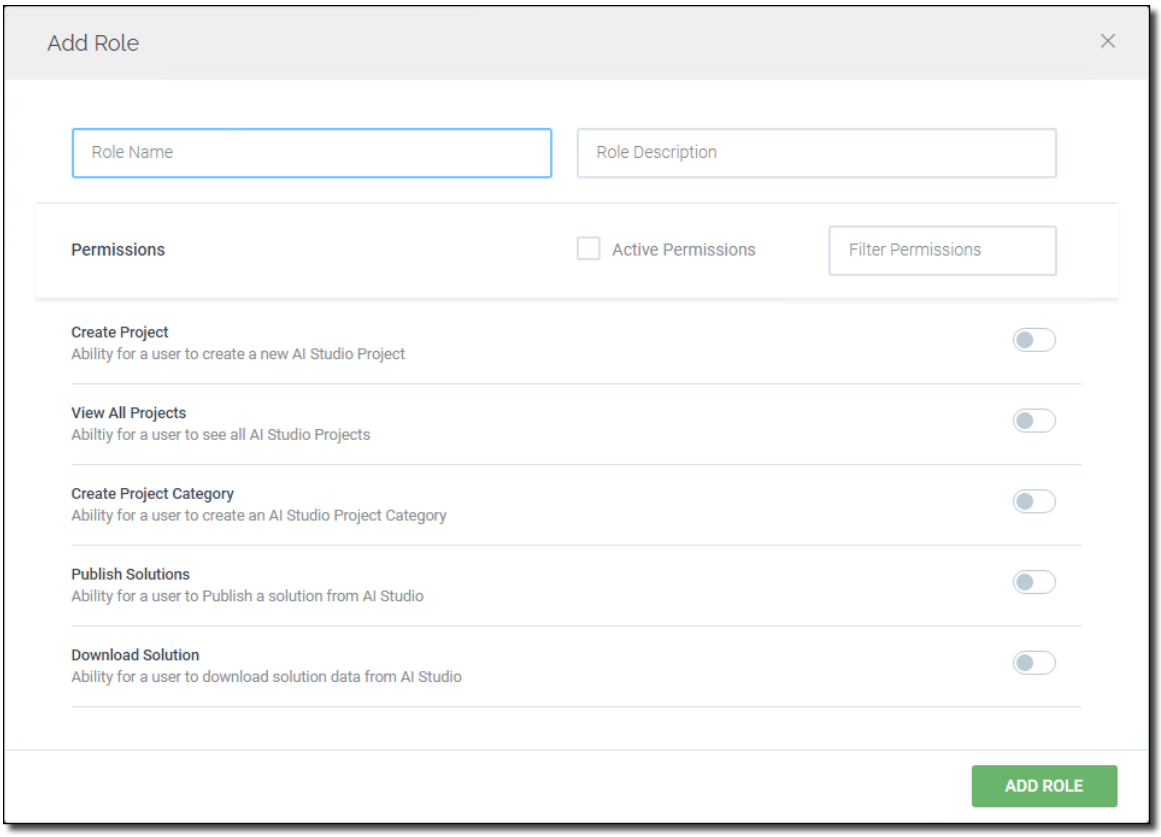
   The **User Management** page is displayed.



3. On the top of the page, click the **Roles** tab. The **Role Management** page is displayed, and all the existing roles are listed.



4. Click **Add Role**. The **Add Role** dialog is displayed.



5. Enter a **role name** and **description**.
6. Select one or more of the possible permissions from the list.
   a. Select **Active Permissions** to display only the permissions you've already selected.
   b. You can also filter the permissions displayed on this list by entering text into the **Filter Permissions** box. For example, if you enter "publish", only the permissions with the text "publish" in the name or description are displayed.
7. Click **Add Role**.

12.2.3.2 Add a user account

When an instance of AML is deployed, it has only two associated user accounts:

- An app admin user with the username "admin".
- A system admin user with the username "system".

You will probably want to give other people in your organization access to AML, so you will need to create accounts for them. When creating a user account, you will also assign permissions to the user through assigning one or more predefined roles.

Note

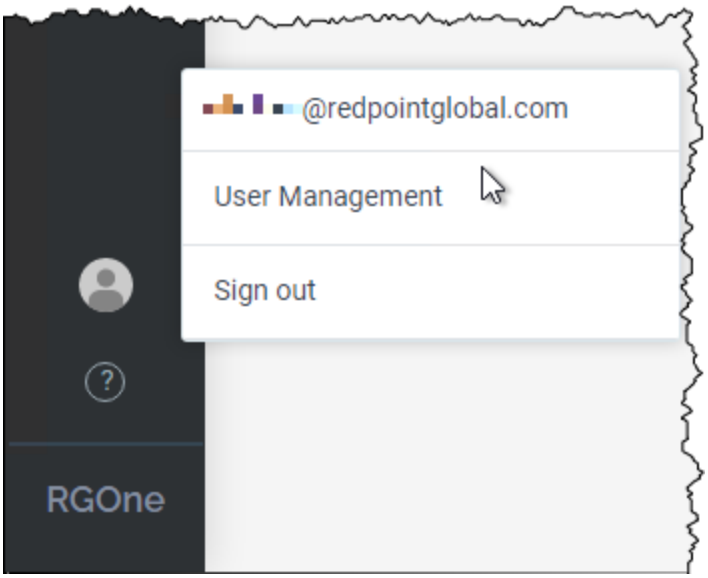
Additional app admin and system admin users can be created only through the AML API. See Add an app admin user account 132 and Add a system admin account 133 for instructions.

To add a user account

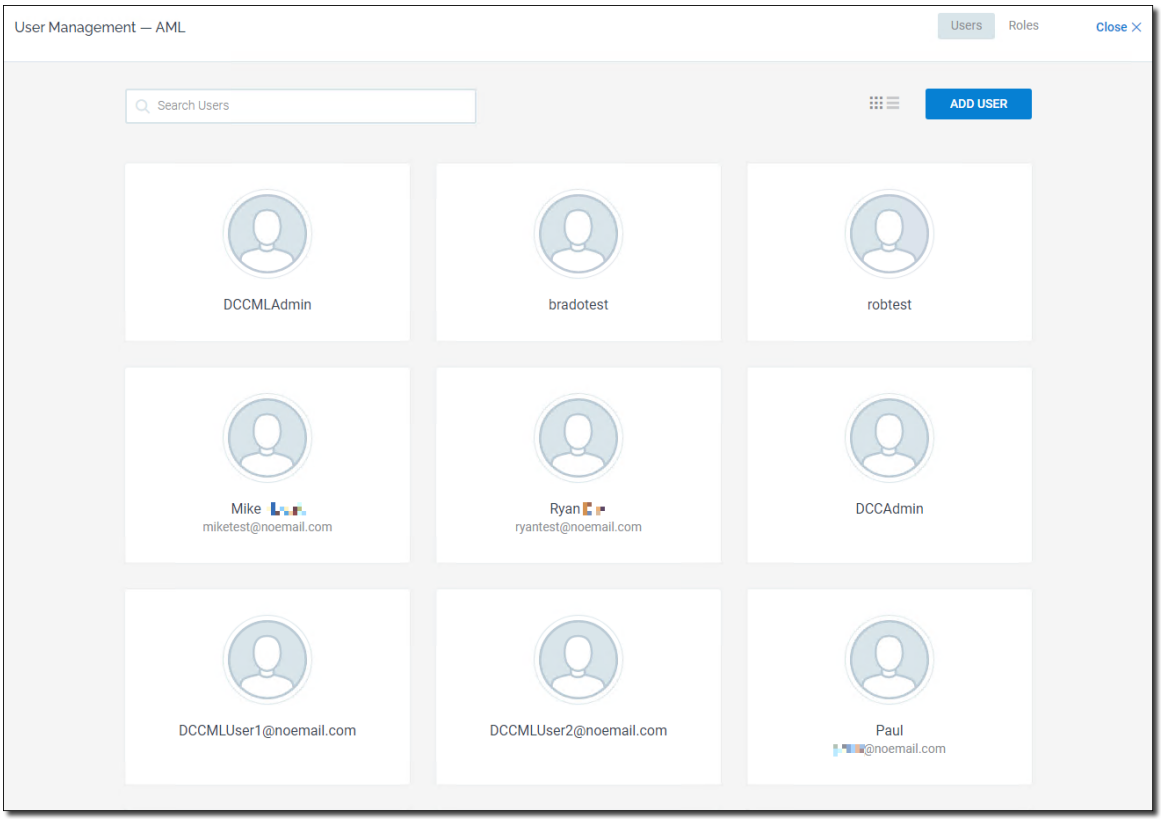
1. Sign in to AML as an app admin.
2. On the left-side navigation menu, click the user icon:

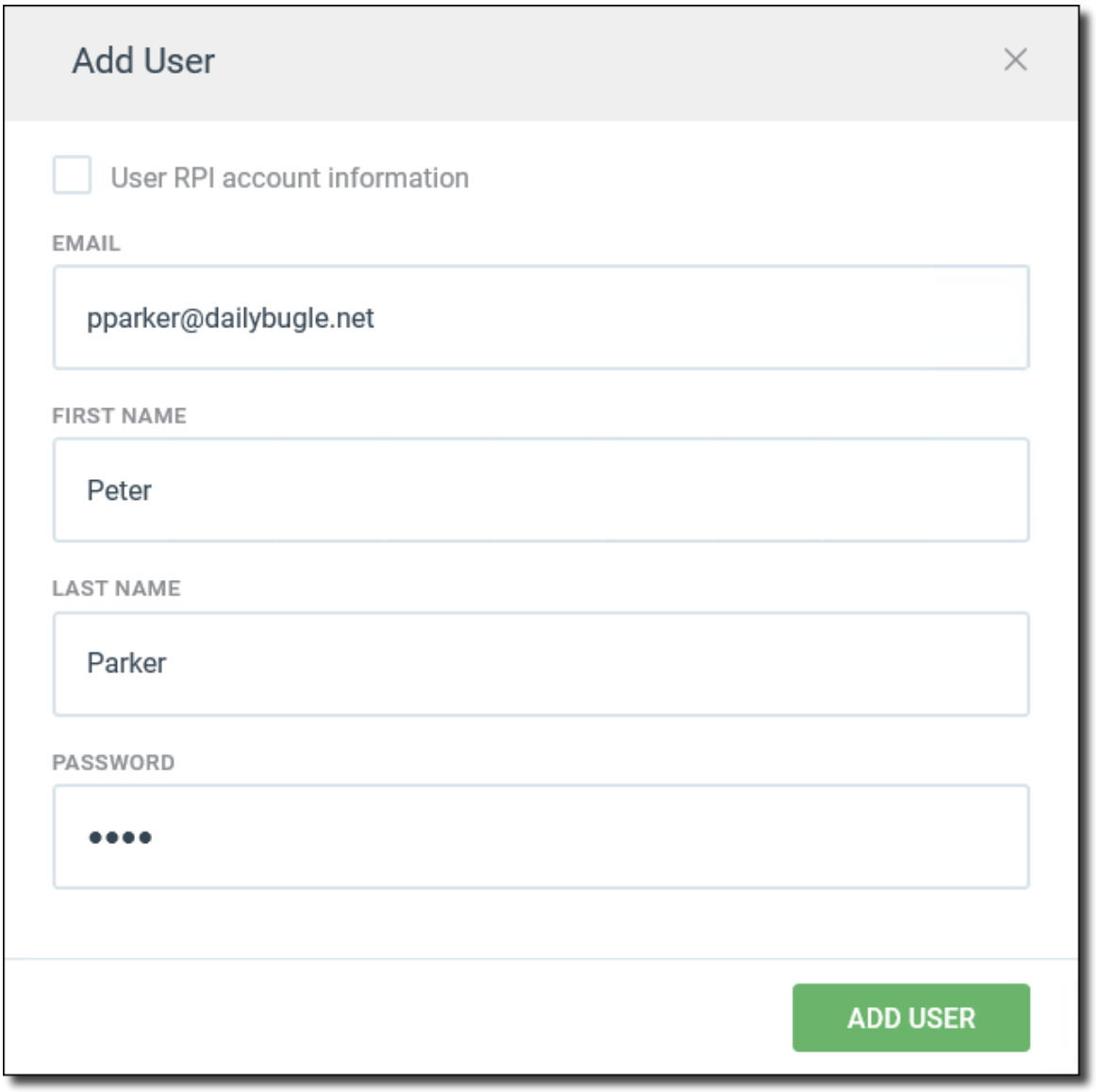
   On the popup menu, select **User Management**.

   The **User Management** page is displayed. If you have previously created user accounts, the users are listed on this page.

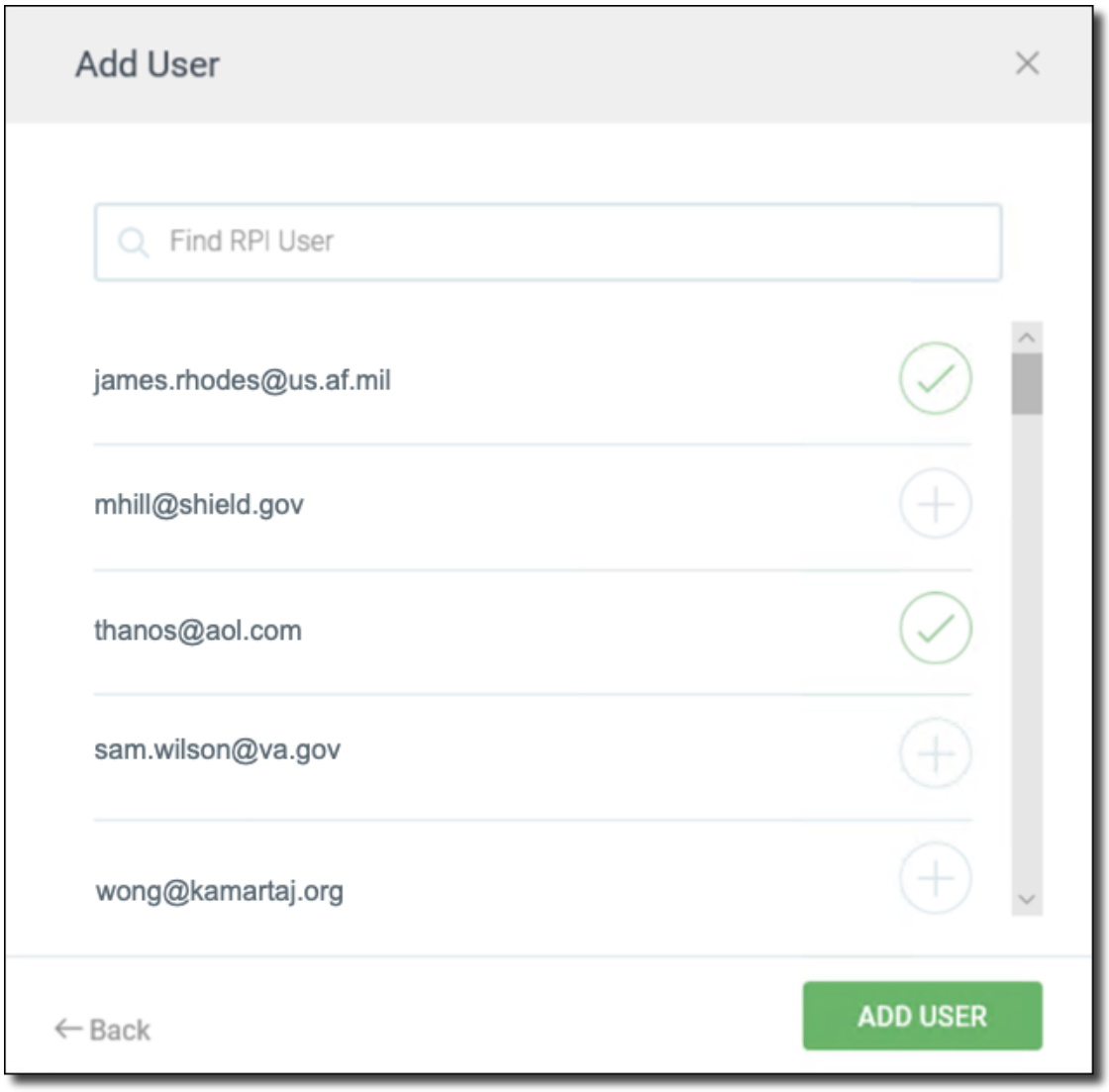


3. Click **Add User**. The **Add User** dialog is displayed.



4. Select the **User RPI account information** box to create AML users for one or more existing Redpoint RPI user accounts.

   a. The dialog displays a list of RPI usernames. You can search for RPI users by entering text in the **Find RPI User** search box.



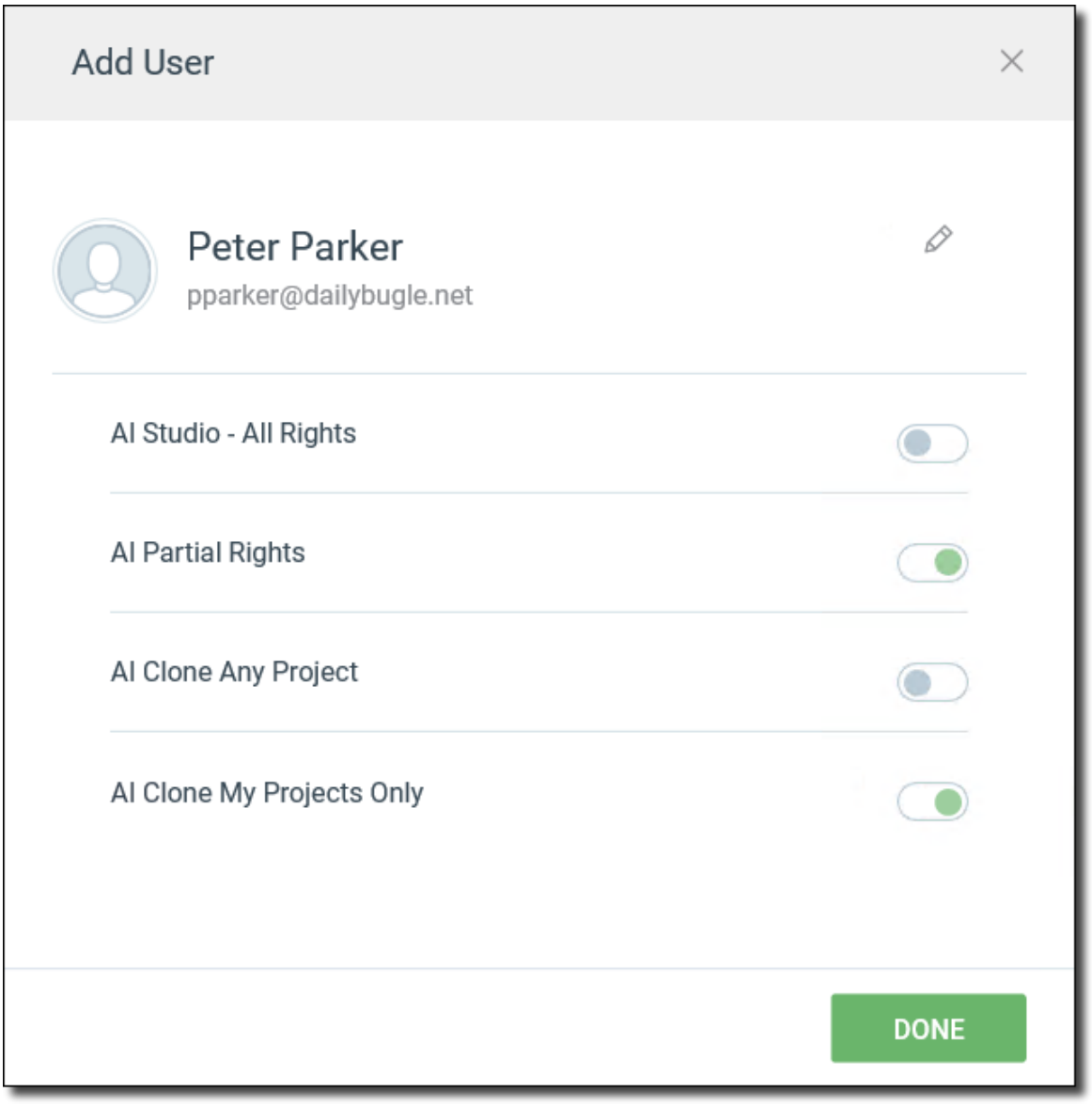
b. Select one or more RPI users on the list.

c. Click **Add User**. AML accounts are created for the selected users. The new AML accounts have the same usernames and passwords as the corresponding RPI accounts.

5. If you are not creating a user with RPI account information, enter an **email** address, **first** and **last** name, and a user password.

6. Click **Add User**.

7. Select one or more previously-defined roles to assign to the user.



8. Click **Done**.

12.3 Features

The following topics are deeper-diving discussions on features and how they are used.

12.3.1 General thoughts on input features

Data files using the `.csv` format (the kind of data files that AML uses) contain one or more records, with each record containing one or more fields. If you look at such a data file in a text editor, each row corresponds to an individual record, and the columns correspond to fields within the records.

For an AML project, we add a data file that provides:

- Input (independent) values. In ML/AI, these fields are known as *input features*. You can define one or more input features.
- Predicted output (dependent) values. In ML/AI, these fields are known as *output features*. For a given record, this is the output value(s) based upon the given input value(s). You can define zero or more output features (zero, if the project type does not generate output features, one in the case of a Classification project).

  During training, the values in the output fields are used for supervised learning. That is, the project predicts the output value(s) based upon the input value(s), and the file output value(s) are used to determine if the project's output prediction(s) are correct.

  After training, a user selects one of the project solutions and publishes it (turns it into a model). When a user calls the model, the user submits only the input field(s), and the model returns the predicted output(s) for the given set of input(s).

When thinking about a project's features, we recommend that you first decide which field(s) to use as the output feature(s). This is the most important decision you will make during the project configuration process because this decision ultimately decides what the model will be used for.

Once the output feature(s) are set, you will generally not change this mapping between different versions of the model. However, you may significantly change the mapping of file fields to input features between versions of a model as you see which inputs have the most influence on the output.

A predictive model is simply a function that maps input features to some desired output feature(s). That’s it. Regardless of whether a model uses machine learning techniques, is constructed manually, or is merely a guess, it is only a data map.

In an ideal world, input features are always causally related to the outputs. For example, in a perfect world the action of “a customer buys bread, peanut butter, and jelly” always causes the creation (output result) of a peanut butter and jelly sandwich. However, there is a large gap between causation and correlation. In the real world, the best that can be typically done is to find input features that are correlated with predicted outputs. In the real world, the action of “a customer buys bread, peanut butter, and jelly” is strongly correlated with the output of a peanut butter sandwich, but some customers may make peanut butter toast, and eat jelly with a spoon (or some variation thereof). Various studies have indicated the presence of a full moon to be correlated with an increase in crime. This does not necessarily mean that a full moon causes more crime to happen. If they actually are correlated, moon phases could, however, be potentially useful as a predictive input feature when constructing a model to predict crime statistics.

In general, the best input features are descriptors of some underlying process or relationship. Descriptors include:

- colors
- weights
- sizes
- amounts

- memberships
- relationships
- product categories
- product types
- ...and so on

These values can be text, numeric, or boolean, but generally not date/time stamps unless they are converted to another numeric form (for example, number of days since a starting time).

Names, especially personal names (for example, Bruce Banner, Wanda Maximoff, or Peter Quill) are an example of personally identifiable information (PII). PII is generally understood as a collection of sensitive material which, taken together, would be sufficient to locate, contact, or otherwise identify a single person. PII data is subject to extensive regulation in the United States (which varies by state), the European Union (under the umbrella of the General Data Protection Regulation, or GDPR), and many other countries. Due to its sensitivity, PII should never be included in a model. Filter out all PII fields from a data set before you upload it to AML. Beyond that consideration, names are generally considered identifiers instead of descriptors. As such they have very limited use as input features for predictive models. To illustrate, it wouldn't make sense to decide what marketing offer an individual gets based on their name. (All people named "Peggy" get a 25% discount, but all people named "Hank" only get a 15% discount).

Since at the lowest level computers process numbers, the relative relationships between the values of these numbers is very important to the modeling process. As with personal names, addresses are also identifiers and generally not well-suited for use as input features. However, state names (for example, MA, NE, or AZ), region codes (for example, NW, SE, or SW), and 5-digit ZIP codes could be used as input features if they potentially contain relationship information to the desired output prediction.

12.3.2 Number of input features vs. number of data samples

A basic rule of thumb is to restrict input features to the lowest number that can reasonably describe the underlying process being modeled. This reduces the noise that can degrade prediction accuracy.

In most real-world marking applications, there is little need to utilize hundreds of input features to model a process. In most cases, up to a few dozen input features (if chosen sensibly) are sufficient to obtain good results.

As with all modeling tasks, the amount of data used to train the models must contain sufficient information to describe the underlying system. Thus, the number of data samples (data rows) needs to cover the space of possible input-output mappings.

Another general rule of thumb is to allocate at least 30 (randomly selected) data samples per each input feature to get reasonable coverage of the mapping. Since the number of required data samples rises exponentially with the number of features, this provides another reason to constrain the number of input features (instead of just pouring all the input features into the model).

A large number of data records also takes longer to train models, which can be important if you need to create and publish a model quickly. In practice, up to 500,000 training samples may be necessary to create accurate models, but often a few thousand may suffice if the number of input features is kept reasonably small.

12.4 Model training and testing

Model creation is a two-part process, consisting of training and testing. Once you've trained your model on a data set, you need to answer the question "how good or bad is this model?" This question is answered by scoring a model against a set of data.

A model score in itself only tells you how well a model scored using a particular data set. Assessing fitness with the training data only shows how well the data scores with "known" data (answer: the model will score highly, since it is being scored against the data it was trained on). Therefore, the statistically valid way to assess fitness is by testing a model using previously unseen (*naive*) data. A good model will score highly using naive data. Naive data is sometimes referred to as *holdout* data (data that you've *held back* from training).

Testing a model with data the model has never seen before answers the question "did the model do what we thought it should do?". For example, if you created a model using stock market data from a certain decade, you could test the model using data from another decade.

12.4.1 Training vs. testing data

Training data is the data you use initially to train the model.

After you train the model, you score the model using *testing data* to assess model fitness. One way to look at this is out of all the data you have, you will use some of it to train with, and you will hold some of it back (the *holdout data*) to test with.

In AML, the default a model's initial training session includes a testing component. After a model's initial training session (that includes some testing), you can do additional testing for further statistical assessment of the model.

There are a number of ways you can split up the data you have into training and testing components:

- Use a single file that contains both your training and testing data.
    - Withhold testing data from one part of the file:
        - Use some part of the file for training.
        - Use another part of the file for testing.
        - For example, you could use 40% of the data file records for training, and use the other 60% for testing.
        - This is the default training and testing process in AML (an initial training session with some testing at the end).
    - Withhold testing data from two parts of the file:

- Use some part of the file for initial training and testing (the AML default process).
- Use another part of the file for further statistical performance assessment of the model.
- For example, if you have 10 million rows of data in a file, you could use 5 million rows to create the model. Some of these 5 million rows would be used for training, and some for testing for initial fitness assessment. The remaining 5 million rows in the file that have never been used for training or fitness assessment could be used for further statistical performance assessment of the model.
- This process is done with the **Sample Prediction** 106 option.

- Use two data files. One data file contains your training data, and the other contains your holdout data.
  - This process is done with the **Sample Using Holdout Data** 105 option.

12.4.2 Choosing holdout data

Normally, the best way to choose holdout data is:

1. Out of the records set aside for training (but never used for training), randomly select some of these records as potential holdout data.
2. Out of this group of random potential candidates, randomly select some of these records to actually be used as holdout data.

This "random of random" data selection means that a model scored twice using this method could produce different scores, because the two groups of data used for testing are randomly selected, and hence not the same data. (Scoring a model with different data can produce different scores.)

However, if you use this "random of random" scheme for holdout data, no matter how many times you train a model (that is, how many versions of the model you generate), you can directly compare the scores generated from these models (you are comparing apples to apples).

12.4.3 How many records to use in training and testing

If you're using a single data file for training and testing, what percentages you use for each action depends on how many records you have. If you have a large data set, you many only need a small percentage of a file for training and testing (because using more records would be overkill). If you have a small data file, you would probably use a larger percentage of the file for each action, because you need a minimum number of records for decent training and testing.

12.4.4 How parents and children are chosen for each generation

In evolution simulations, parents (AKA parent solutions or parent models) are the starting solutions for a generation. Each parent produces one or more offspring (AKA offspring solutions or offspring models) that are variations of the parent.

When creating a project (model), the user sets the values for the number of parents (x) and number of children (y). These values are used for every generation, including the first.

For x parents and y offspring:

- There will be x*(y+1) solutions (children) for that generation to rank.
- Out of those children, the top x become the parents for the next generation.

In every generation, for each solution:

- The model is trained.
- A fitness score is calculated for each solution in that generation.
- A fitness score represents “goodness of fit”, and is not a percentage. It measures how much error the model has when compared to known “truth”.
- Fitness scores for Regression and Classification models (supervised learning) are normalized to the range [0.0, 1.0]. Cluster fitness models are not bounded to [0.0, 1.0]—scores can exceed 1.0 for certain measure types.
- The lower the value, the better the score. A fitness score close to zero indicates the model has a very low amount of error. A score of 0 indicates the model is perfectly accurate (across the set of known input values that were provided). This is normally not likely (except for certain deterministic models like decision trees).
- Additionally, fitness scores use different measures for Classification (cross-entropy) and Regression (mean-squared error). Multiple error measures are typically calculated (for example, area under the curve, ROC curves, Z scores, and P scores) that may be included in a final fitness score sometime in the future.

Example

Every parent x has y number of children. A single generated solution is represented by s.

Generation 1

If we start with x=3 and y=5:

x	x	x
yyyyy	yyyyy	yyyyy
ssssss	ssssss	ssssss

In this example, for every parent there are five children, which means 1 parent + 5 children = 6 solutions to consider per parent. Since there are 3 parents, the total number of solutions for this generation is 3*(5+1) = 3*6 = 18.

Generation 2

Out of 18 possible solutions, the top x (3) solutions become parents for the next generation.

12.4.5 How AML calculates a solution fitness score

In general, a good model performs well with data it has never seen before (AKA *naive data* or *testing data*). Assessing fitness with the training data really only shows how well the data does with "known" data (which it should, since it was trained with it). Therefore, the statistically valid way to assess fitness is by testing using previously unseen (naive) data.

A solution fitness score is calculated with this equation:

```
overallFitness = weight * F(trainingData) + (1-weight)* F(testingData)
```

`F(trainingData)` = Partial score for training data. This score is calculated using the portion of data used for training (for example, 30%, 50%, or 75% of the data set). The portion percentage is set by the **Sample** value.

`F(testingData)` = Partial score for testing data. This score is calculated using the portion of data used for testing (for example, 30%, 50%, or 75% of the data set). The portion percentage is set by the **Sample** value.

`weight` = value from 0 to 1. This is the **Fitness Portion** value.

12.4.5.1 Sample

Sample determines what percentage of a data set is used for training, and what percentage is used for testing, according to the formulas:

% data set used for training = `sample`

% data set used for testing = `(1 - sample)`

A default value of 0.5 means 50% of a data set will be used for training and 50% will be used for testing.

A value of 0.3 means that 30% of a data set will be used for training and 70% used for testing.

12.4.5.2 Fitness Portion

As explained above, this is a value from 0 to 1. It sets the percentages of the partial scores used for calculating the overall fitness score, and is represented in the fitness score equation by the variable `weight`.

If `weight` (fitness portion) is set to 0.0, `overallFitness` is simply equal to `F(testingData)`. (That is, `overallFitness` is calculated using only the testing data.)

If `weight` is set to 1.0, `overallFitness` is simply equal to `F(trainingData)`. (That is, `overallFitness` is calculated using only the training data.)

If `weight` is set to 0.4 (40%), then the overall fitness score is calculated using 40% of the training fitness score and 60% of the testing fitness score.

12.4.5.3 Example

Let's assume:

- That we're using a data set of 1,000 records.
- We set **Sample** to 0.4.
- We set **Fitness Portion** to 0.7.

Using the fitness score equation:

```
overallFitness = weight * F(trainingData) + (1-weight)* F(testingData)
```

% data set used for training = 40% of all records, so (0.4 * 1000) = 400 records

% data set used for testing = 60% of all records, so (0.6 * 1000) = 600 records

Then, `overallFitness = 0.7 * F(trainingData) + 0.3 * F(testingData)`

Which is to say, `overallFitness` is 70% of the fitness score using training data (400 records out of the data set) + 30% of the fitness score using testing data (600 records out of the data set).

12.4.6 How AML calculates a model rank value

In every generation, after solution fitness scores are calculated, each solution is ranked by fitness score using either an:

- Elitist method (AKA survival of the fittest)—Rank scores from worst to best.
  - The top x solutions become the parents of the next generation.
  - This method is structured and repeatable (the ranking is always the ranking). However, this method lacks variation, which may lead to a plateau in which you lock yourself away from a better solution.
  - Basically, this ranking is across all generations but the hope is that the best scoring models are in the latest generation (though this is not always true).
- Tournament method—Think of this as a medieval tournament of knights (each knight competes to be retained based on competition with N other knights).
  - A knight is ranked by combat (fitness score) against N randomly-selected knights within the population of that generation.
  - A knight fights N other knights and gets so many wins.
  - The two knights are picked at random (except you can't fight yourself).
  - If there is a tie between two knights, the winner is chosen at random.
  - A knight fights other knights, gets so many wins, until it reaches the limit of the number of knights it's permitted to fight (N). At the end, the knights are ranked by number of wins. (For now, N is a constant, but in the future it can be modified in the UI).
  - Tournament method is not *survival of the fittest*; think of it as *culling of the least fit*.
  - Tournament method allows for more variance than the Elitist method, which provides opportunity for finding a better solution.

- In Tournament method, the top-ranking solution slot is always guaranteed, but after that solution ranking could be different.

The ranking method is chosen when setting up the model project. For this Classification model example, we are using an elitist ranking method.

At the end of a generation (cycle), solutions are sorted based on their ranking, and only the top X solutions are retained as starting points (parents) for the next generation.

The evolutionary process continues until one of the two user-selectable criteria are met:

- Total wall-clock time has expired to create a finished best model, or
- The desired model fitness score goal is attained.

12.4.7 Training algorithms

The following topics explain the algorithms AML uses to train a model.

Each algorithm has its strengths and weaknesses. Selecting more than one model algorithm for the project gives the optimizer the freedom to find the "best" algorithm (and parameterizations thereof) for the data. (That is, selecting more than one model algorithm gives the optimizer more degrees of freedom to find something that works well.) Selecting a single model algorithm potentially constrains the search to only sub-optimal model algorithms.

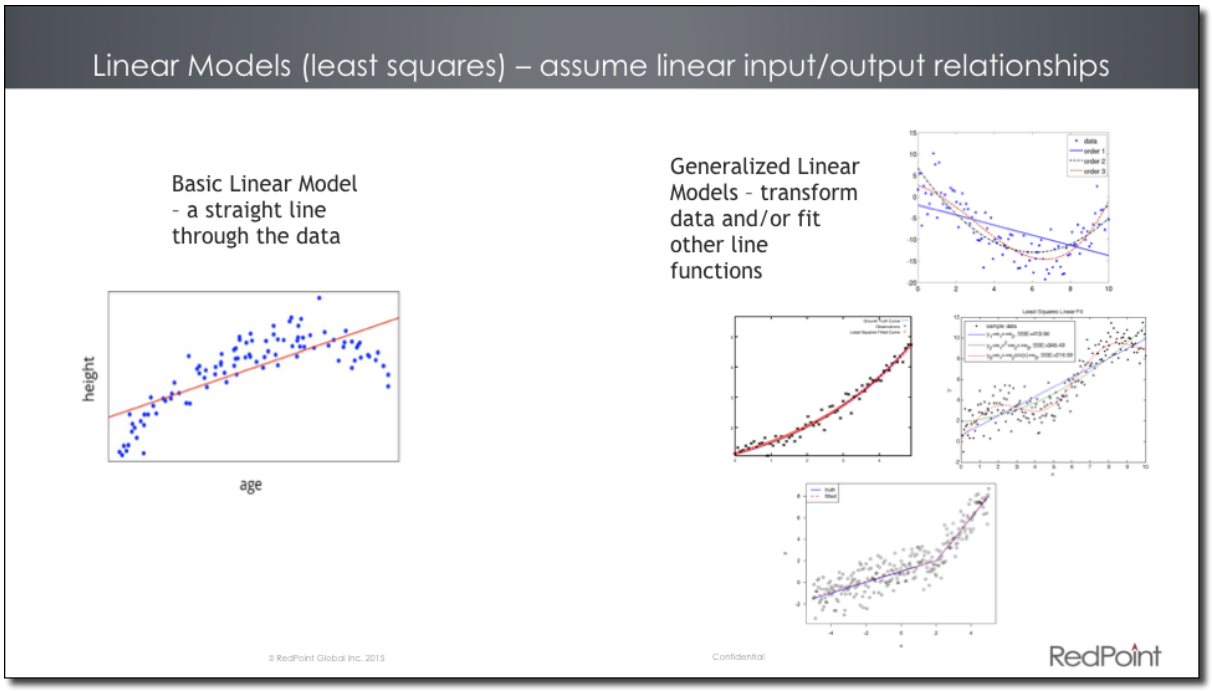
The Linear Least Squares and Partial Least Squares algorithms allow only one output feature at a time to be predicted. However, the Neural Networks algorithm can generate predictions for more than one output feature at a time.

12.4.7.1 Linear Least Squares

The *Linear Least Squares* algorithm examines the linear relationship between two continuous variables (one input feature and one result output) and finds the best fitting straight line through a set of points.

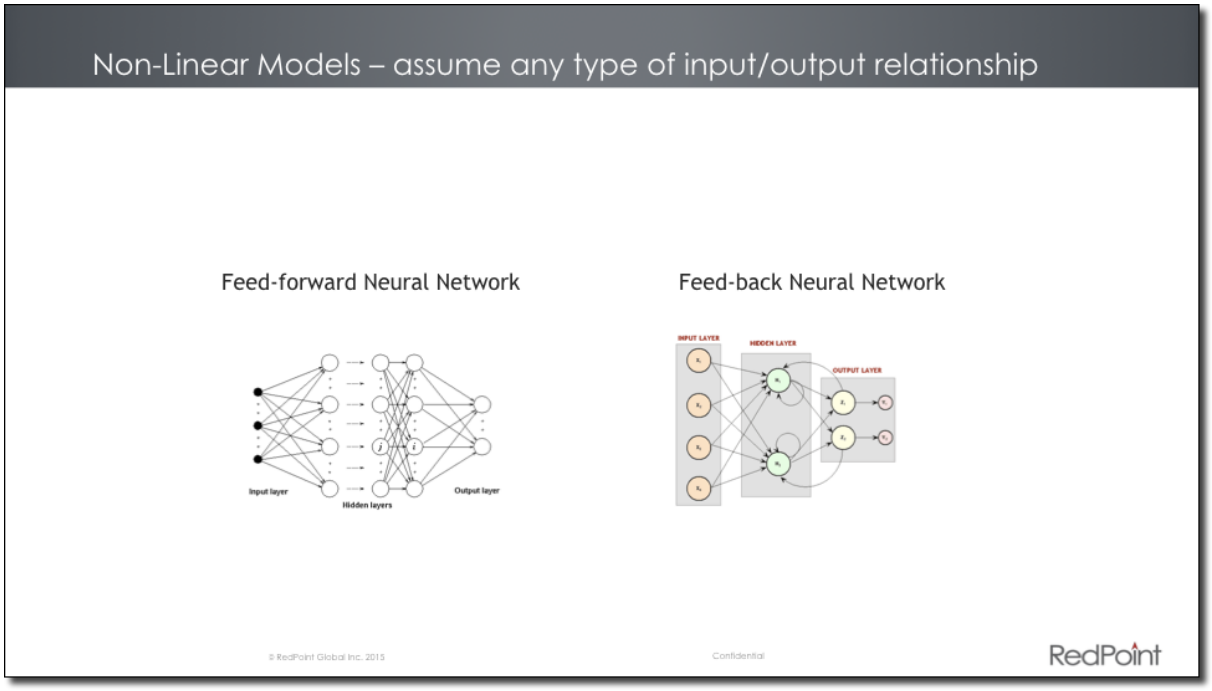
12.4.7.2 Partial Least Squares

The *Partial Least Squares* algorithm reduces the predictors to a smaller set of uncorrelated components and performs least squares regression on these components, instead of on the original data.



12.4.7.3 Neural Networks

The *Neural Networks* algorithm passes input features forward through a sequence of layers before turning them into outputs. In each layer, inputs are weighted in various combinations, summed, and passed on to the next layer.



Note that selecting **Neural Networks** displays additional algorithm parameters that control and/or constrain the modeling process.

Nodes and layers

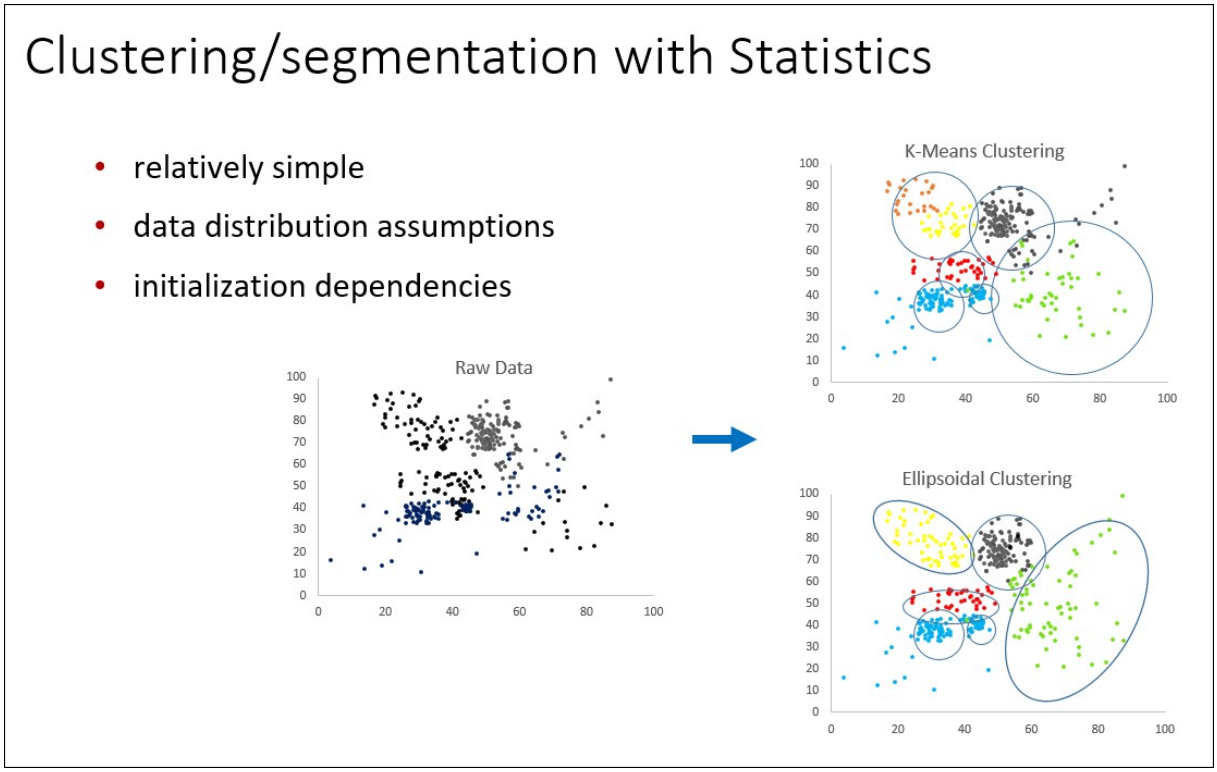
A *node* is a model of a single neuron in the brain (which sums up all the inputs and triggers the output if a given threshold is reached). A *layer* is a group of nodes at the same "level".

Per proven theory, two layers is enough to map any regression prediction task. However, theory doesn't specify how to do it (just that it can be done). The number of nodes in a layer, node functions, and weights need to be discovered through training.

A good rule of thumb is that the number of nodes in a layer should usually be proportional to the number of input features. In most cases, 2-3 layers should be sufficient for most prediction problems. The default **layer** and **node** values should work for most situations. Note that using larger numbers of nodes and layers facilitates learning at the expense of training time and increases the risk of overtraining (that is, learning the unwanted "noise").

12.4.7.4 Clustering algorithms

Clustering/segmentation (*model*) algorithms utilize distance measures to assign records to clusters/segments. The distances are measured, then records are (typically iteratively) assigned to clusters. The "goodness" of a clustering/segmentation solution is assessed using the **cluster measure** option.



For a list of publications that describe the cluster model algorithms in detail, see Clustering model algorithms reference publications 128.

Cluster measure is the function used to evaluate cluster quality.

For a list of publications that describe the cluster measure functions in detail, see Clustering measure functions reference publications 129.

Note

Some cluster measure functions are semi-supervised, and require the user to specify an output "truth" feature. See Semi-supervised Clustering measure functions 129 for a list of these functions.

Cluster range sets the minimum and maximum number of clusters to generate. See Thoughts on cluster size 126 for a discussion of this topic.

12.5 Project actions

The following topics describe the actions you can take on a project through its pulldown menu on the **Project Manager** page.

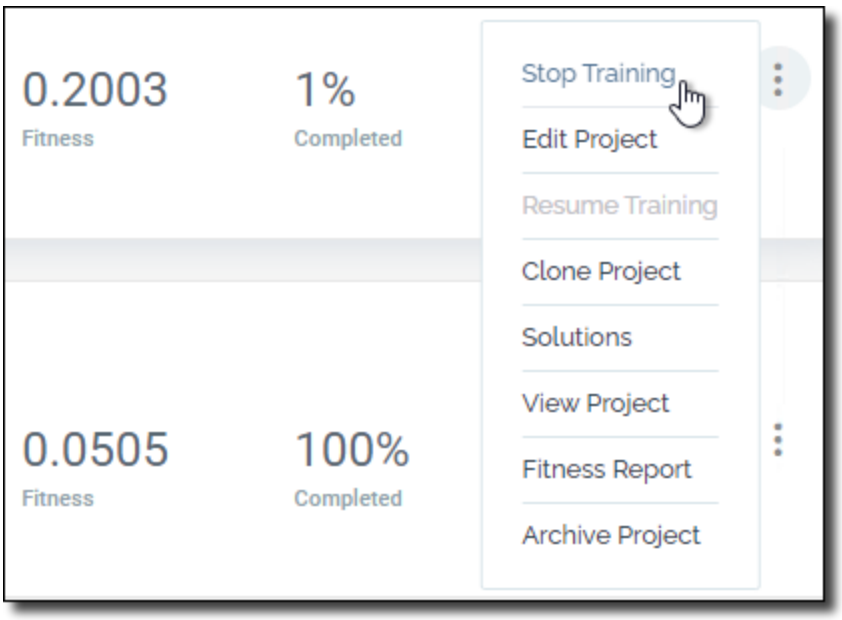
12.5.1 Stop Training

While a project is training (and before completion), you can stop the training process.

Note

Projects in which training is stopped before completion are still usable projects. You could set a project to train for 24 hours, but after 16 hours think "I need this now", stop training, and publish one or more of the project's generated solutions.

1. Hover over the project's pulldown menu and select **Stop Training**.

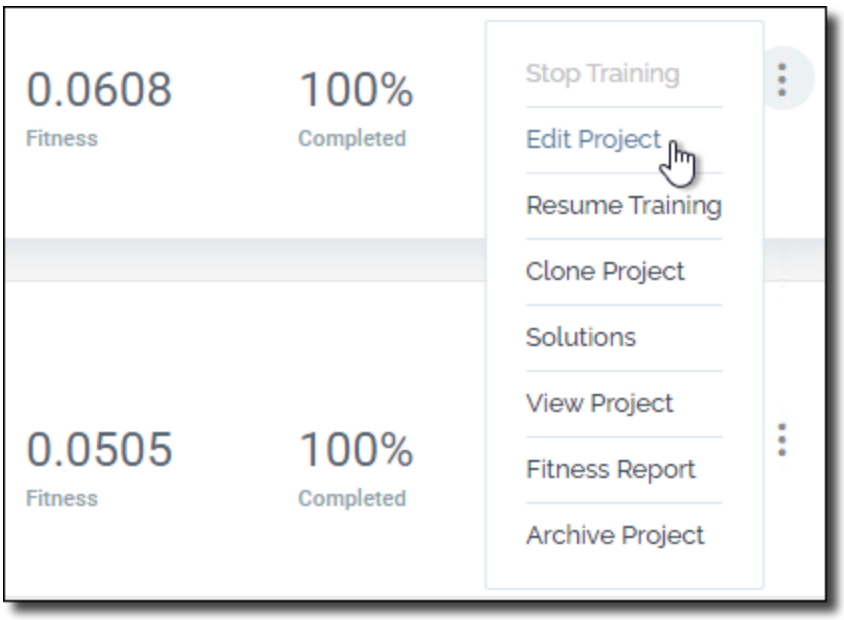


2. The project status bar turns red and displays the status **Canceled**.

12.5.2 Edit Project

Once you've created a project, you can go back and edit it at any time.

1. Hover over the project's pulldown menu and select **Edit Project**.

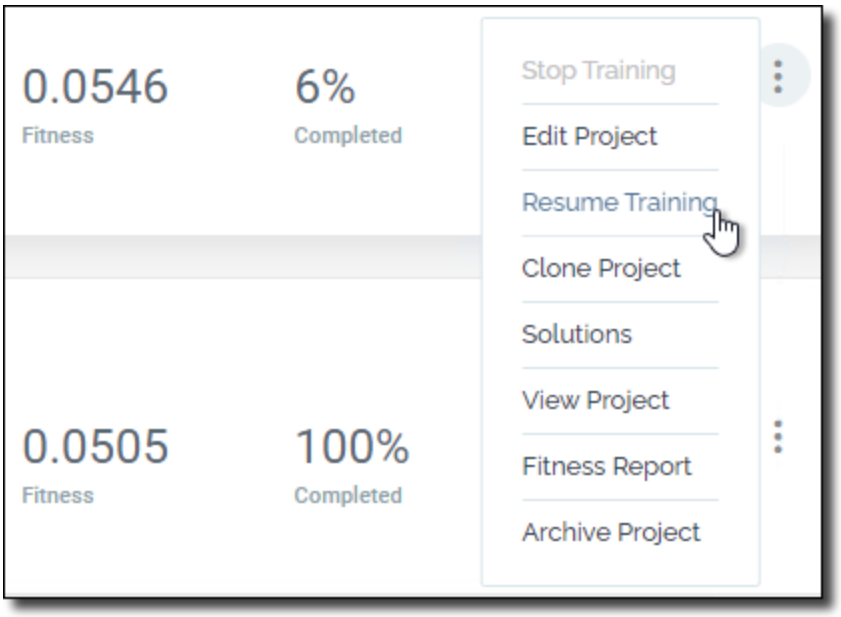


2. The project's **Configure Project** page is displayed. Edit one or more project configuration sections as you see fit.

12.5.3 Resume Training

If you have halted a project's training (by selecting **Stop Training** on the project's pulldown men), you can start training the project from the point at which training was halted.

1. Hover over the project's pulldown menu and select **Resume Training**.

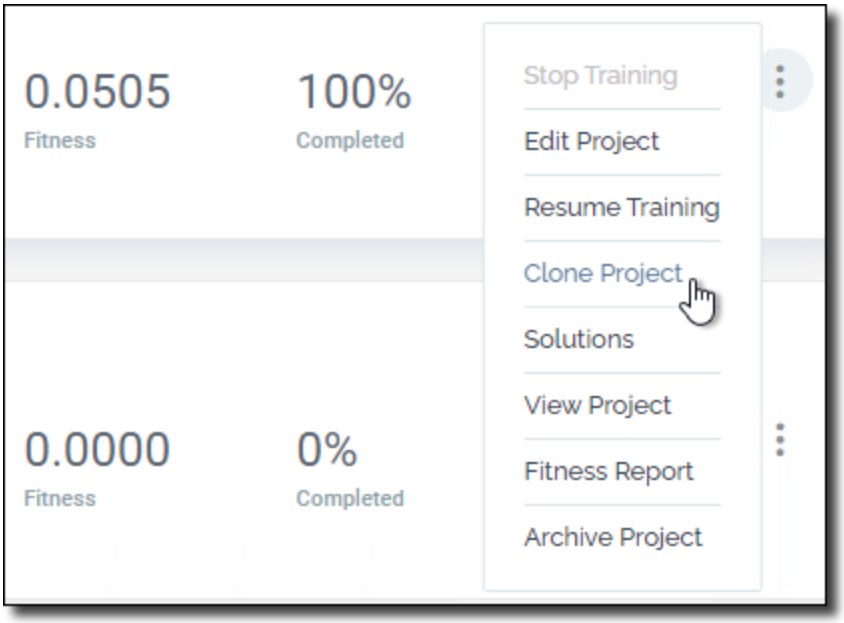


2. The **Project Configuration** page is displayed. Note that when restarting a project, you can edit only the project's **Run Settings** or **Automation Run Settings** sections before clicking **Start Training**.

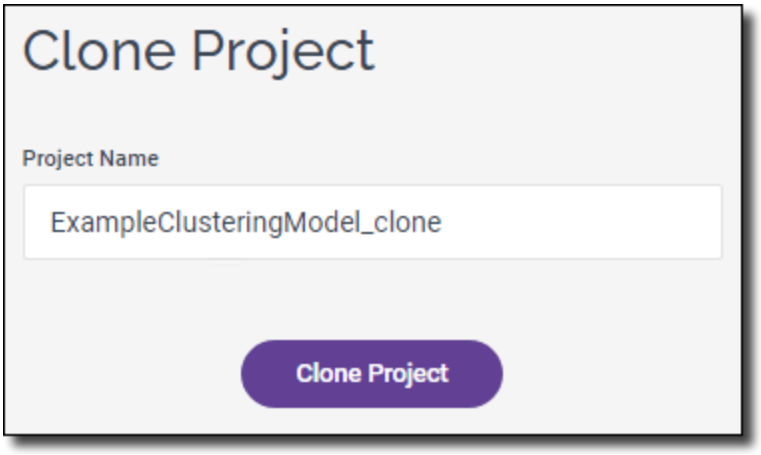
12.5.4 Clone Project

Cloning a project is useful for "what-if" scenarios. You can change any settings in a cloned project, and the original project is not modified. Cloning allows you to easily branch off an existing project.

1. Hover over the project's pulldown menu and select **Clone Project**.



2. The **Clone Project** page is displayed. Note that the default name of the cloned project is *original-project-name*_clone.

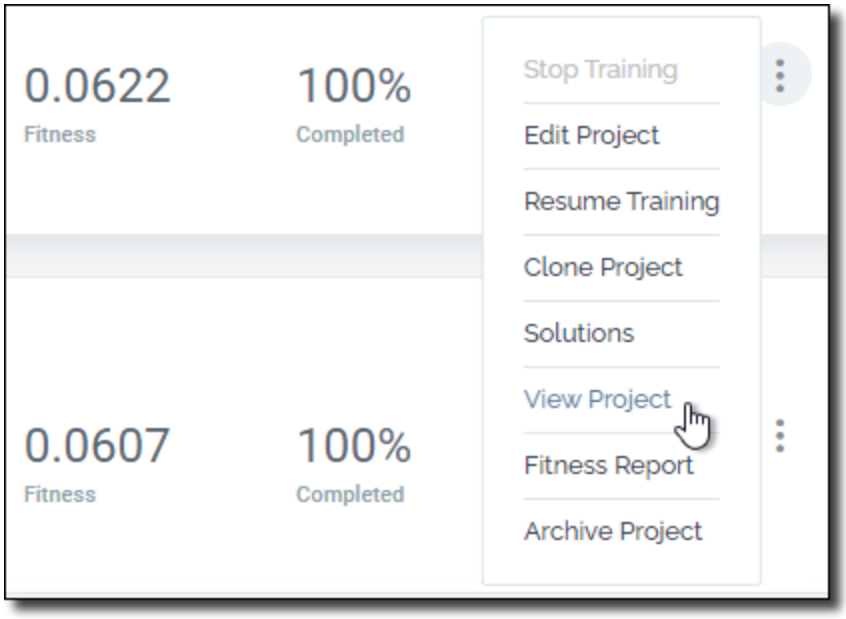


3. Click **Clone Project**. Note that on the **Configure Project** page, you can keep the project section settings from the parent project, or you can change them.

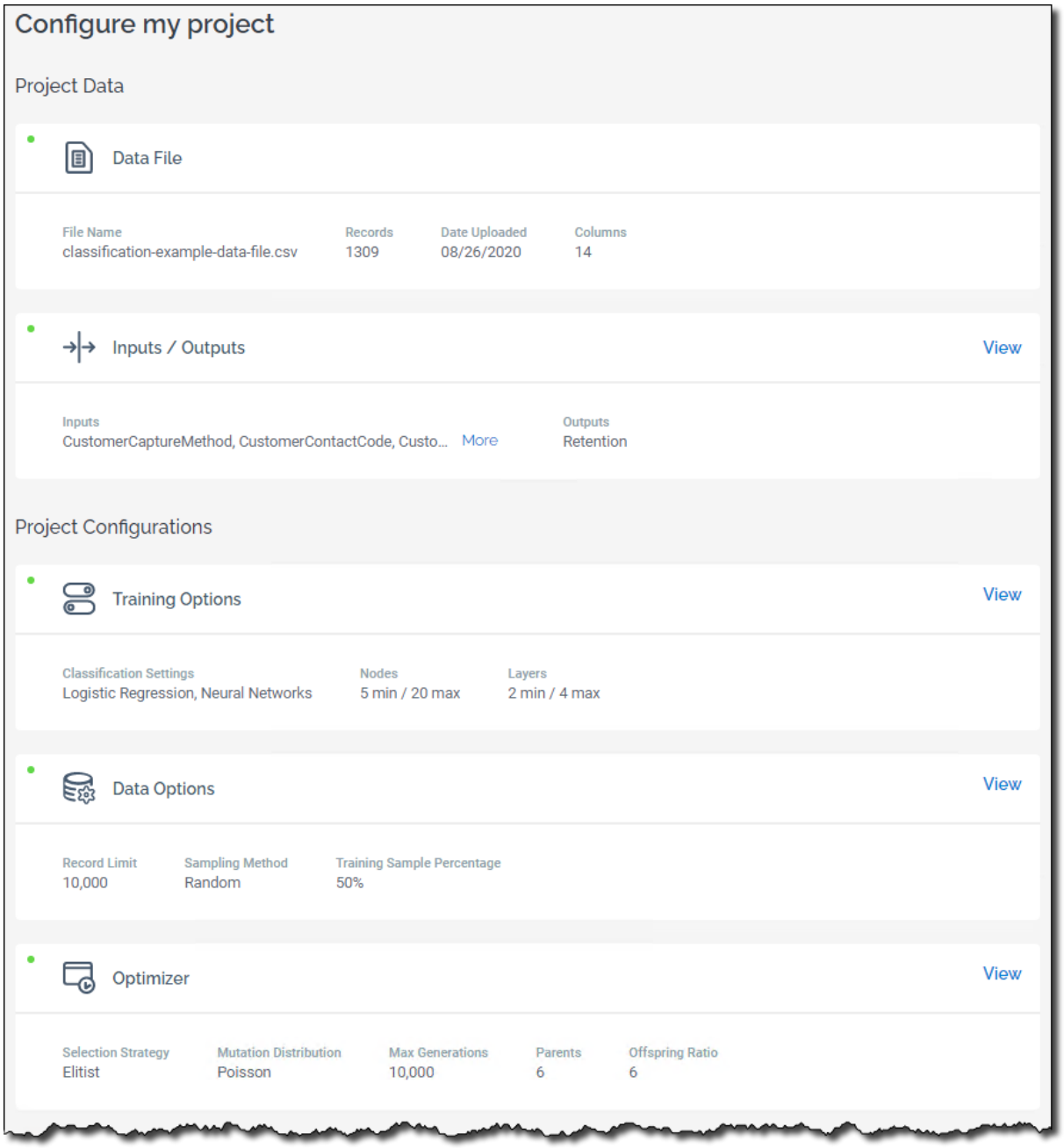
12.5.5 View Project

This option displays the **Configure Project** page in View Project mode.

Hover over the project's pulldown menu and select **View Project**.



The **Configure Project** page is displayed in View Project mode. Clicking **View** on any section (except **Data File**, which already displays all the important information on the main page) displays the section settings, but you are forbidden from changing any settings. Also, you cannot train the project in View Project mode.



12.5.6 Fitness Report

For a description of this report, see Analyze a project's training results 53.

12.5.6.1 Graph options

These options are available from the **Graph Option** pulldown menu.

Download as—Download the graph as an image file.

Save data—Save the graph data as a file in one of several data formats.

Annotate—Allows you to draw lines over the graph. You can use this feature to add text, arrows, and so on.

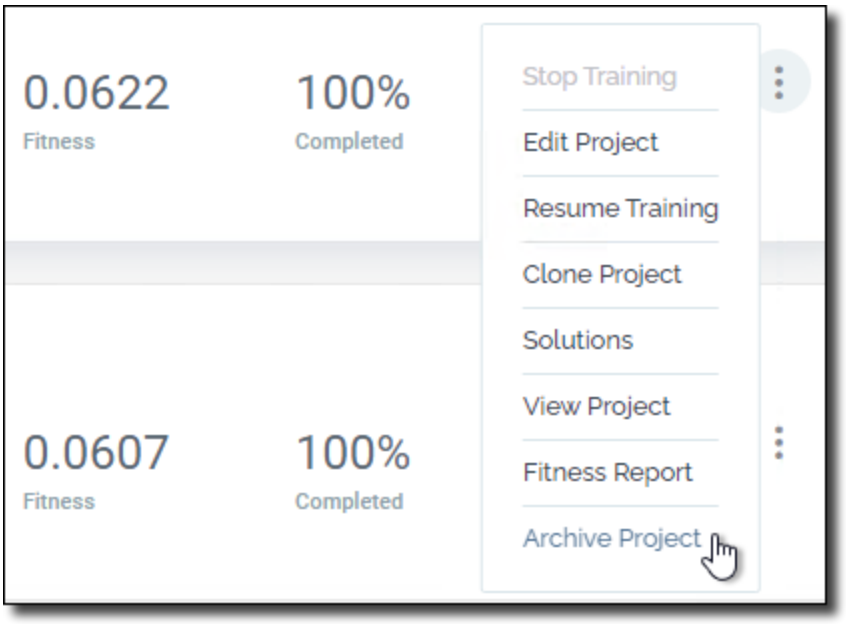
- **Color**—Change the line color.
- **Undo/Redo**—Undo or redo graph annotations.
- **Save as**—Save annotated graph as an image file.
- **Print**—Print the annotated graph.
- **Cancel**—Remove graph annotations.

Print—Print the graph.

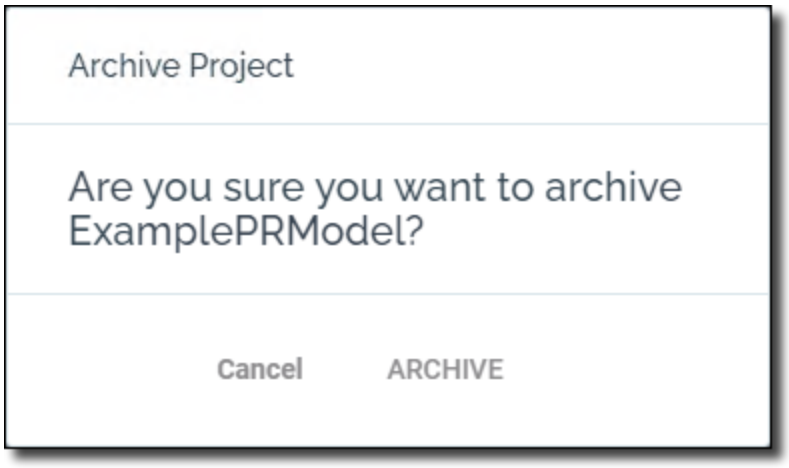
12.5.7 Archive Project

In the AML paradigm, projects are never deleted. You can always go back and view a previously-created project. But if you want to un-deploy a model, you can archive it. Archived projects are not shown by default (but you can change this setting in the project query section).

1. Hover over the project's pulldown menu and select **Archive Project**.



2. A confirmation dialog is displayed. Click **Archive**.



3. The project is archived. You can search for an archived project on the **Project Manager** page by selecting **Archived Projects** in the project search criteria section.

12.6 Solutions page

Rank

See How AML calculates a model rank value 94 for a detailed discussion of this value.

Fitness Score

See How AML calculates a model fitness score 91 for a detailed discussion of this value.

Duration

The total overall time allocated to optimize (through some number of generations) a final “best” solution (that is, continue the optimization process until you run out of allocated time). The best solution in the population at that time becomes the “final best” solution.

Generation

The training generation to which the solution belongs. This value gives you a sense of how far along in the training process it took to calculate the solution.

Number of solutions

The number of solutions displayed is based on the number of parents you set in the project **Optimizer** section. For example, if you change the **Parents** value to 4, the four highest scoring-solutions are displayed on the **Solutions** page.

Project Version

Some notes on project versions:

- A new project version is created:
  - When you create a new project.

  - The first time you start making edits to a previously-trained project.
- This version does not change with subsequent edits you may make. You can save, close, and return to the project and continue making edits without a version change.
- When you start training a project, the version of the configuration at that point is locked and that version can no longer be changed.
- The configuration for each locked version is kept for historical reference.
- Any project edits that occur after a version is locked trigger the creation of a new version of the project.
- When viewing a project, you can select a specific version using the **Version** menu.

12.6.1 Report page

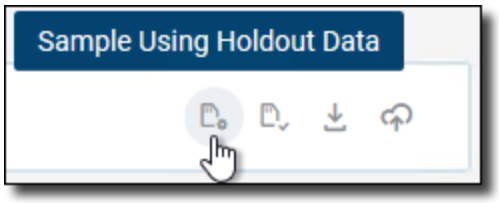
Clicking the **Report** tab on the **Solutions** page displays the project's training score vs. generation graph. This is the same graph that is displayed from the **Project Manager** page if you hover over a project's pulldown menu and select **Fitness Report**.

See Analyze a project's training results 53 for an explanation of this graph.

12.6.2 Solution actions

Click one of the icons on the right side of a solution row to take action on a solution.

12.6.2.1 Sample Using Holdout Data



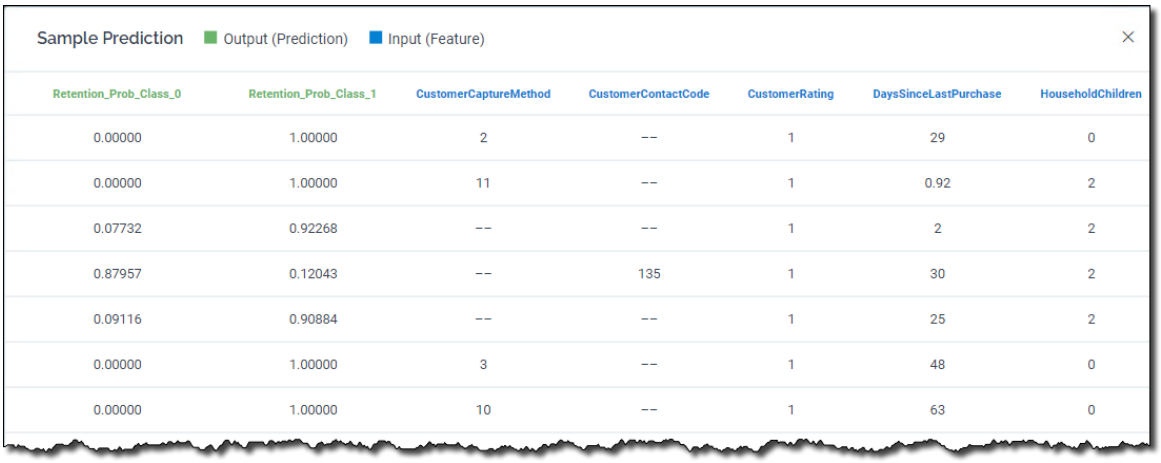
This feature allows you to test a model with data the model has never seen before and answer the question "did the model do what I thought it should do?".

See Model training and testing 90 for an explanation of how your data is used for model training and testing.

See Choosing holdout data 91 for tips on how to best choose your holdout data.

This functionality requires uploading a file.

1. Click the **Sample Using Holdout Data** icon for a solution.
2. Drag your holdout data file to the popup dialog, or browse for it.
3. The **Sample Prediction** dialog is displayed.



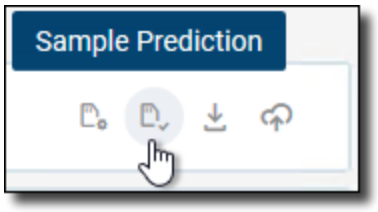
Retention_Prob_Class_0	Retention_Prob_Class_1	CustomerCaptureMethod	CustomerContactCode	CustomerRating	DaysSinceLastPurchase	HouseholdChildren
0.00000	1.00000	2	--	1	29	0
0.00000	1.00000	11	--	1	0.92	2
0.07732	0.92268	--	--	1	2	2
0.87957	0.12043	--	135	1	30	2
0.09116	0.90884	--	--	1	25	2
0.00000	1.00000	3	--	1	48	0
0.00000	1.00000	10	--	1	63	0

4. Click **x** on the upper-right-hand part of the dialog to close it.

The blue column(s) are input features from the holdout data set, and the green column(s) are calculated output features.

In this example, **Retention_Prob_Class_0** is the output feature calculated using the model trained from the original data set, and **Retention_Prob_Class_1** is the output feature calculated using the model trained from the holdout data set.

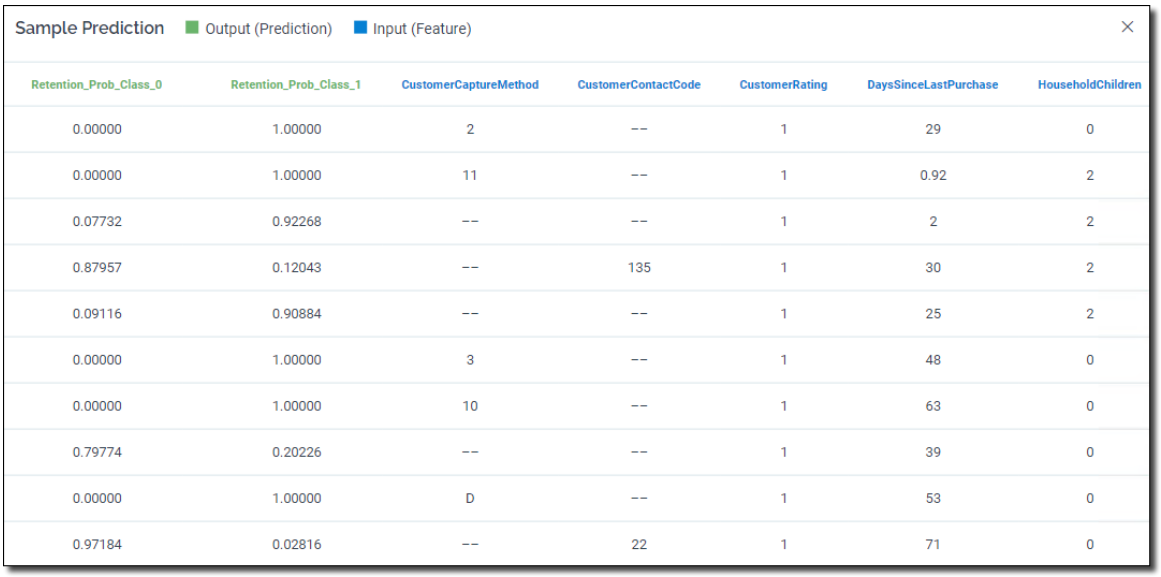
12.6.2.2 Sample Prediction



This is an easier version of the “sample using holdout data” functionality, and answers the same question (“am I getting the output I expected?”). This is a quick way to test if a model is predicting output that looks correct.

This functionality uses data you’ve already uploaded for training, and does not require uploading a separate file. It basically takes the first 20 rows of input data that was used for training the model and calls the predictor service for each row.

1. Click the **Sample Prediction** icon for a solution.
2. The **Prediction Sample** dialog is displayed.



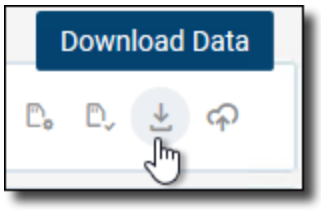
Retention_Prob_Class_0	Retention_Prob_Class_1	CustomerCaptureMethod	CustomerContactCode	CustomerRating	DaysSinceLastPurchase	HouseholdChildren
0.00000	1.00000	2	--	1	29	0
0.00000	1.00000	11	--	1	0.92	2
0.07732	0.92268	--	--	1	2	2
0.87957	0.12043	--	135	1	30	2
0.09116	0.90884	--	--	1	25	2
0.00000	1.00000	3	--	1	48	0
0.00000	1.00000	10	--	1	63	0
0.79774	0.20226	--	--	1	39	0
0.00000	1.00000	D	--	1	53	0
0.97184	0.02816	--	22	1	71	0

3. Click **x** on the upper-right-hand part of the dialog to close it.

The blue column(s) are input features from your data set, and the green column(s) are calculated output features.

In this example, **Retention_Prob_Class_0** is the output feature calculated using the model trained from the entire data set, and **Retention_Prob_Class_1** is the output feature calculated using the model trained from the beginning records of the data set.

12.6.2.3 Download Data



This feature downloads model data as an .xml file. The data includes all the model details (parameters, settings, statistics—basically every decision made in generating the model).

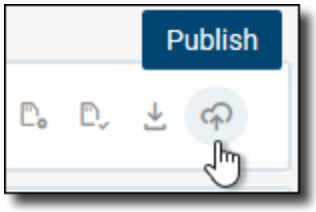
Click the **Download Data** icon for a solution. The .xml file is saved to your default system download directory.

12.6.2.4 Publish

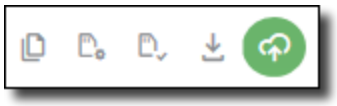
A trained project produces a set of solutions, each with an assigned score. You can publish one of the solutions (the best-scoring one is a good choice) to make it available to the prediction engine. A published solution is called a *model*.

Another way to think of this is publishing a solution makes the solution "live"—puts it in production. Published solutions (models) can be called through the AML API 132 in order to predict

expected outputs based on a set of inputs. AML also tracks model metrics (number of API calls and throughput).

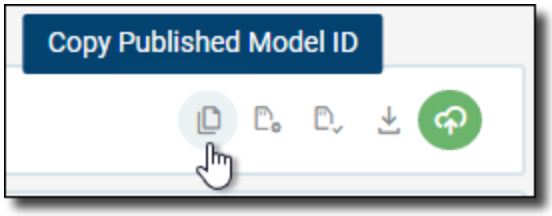


After a solution is published, the icon turns green.



Once a solution is published, you can click the icon again to unpublish it.

Note that after you publish a solution, another icon appears on the left side.



Clicking this icon copies the ID of the published solution (model) to the copy/paste buffer.

You will need the ID of a model when using it in the AML API.

Note

A model ID number is not generated until a solution is published.

12.7 Project settings and UI

This section provides additional information on project settings and project UI functionality.

12.7.1 All project types

The following topics are applicable to all project types.

12.7.1.1 Project Configuration page settings

When you create a new project, you are required to change the settings only in the **Project Configuration** page **Data File** and **Inputs/Outputs** sections. The settings in the other sections are optional. The following topics describe the settings in the optional sections.

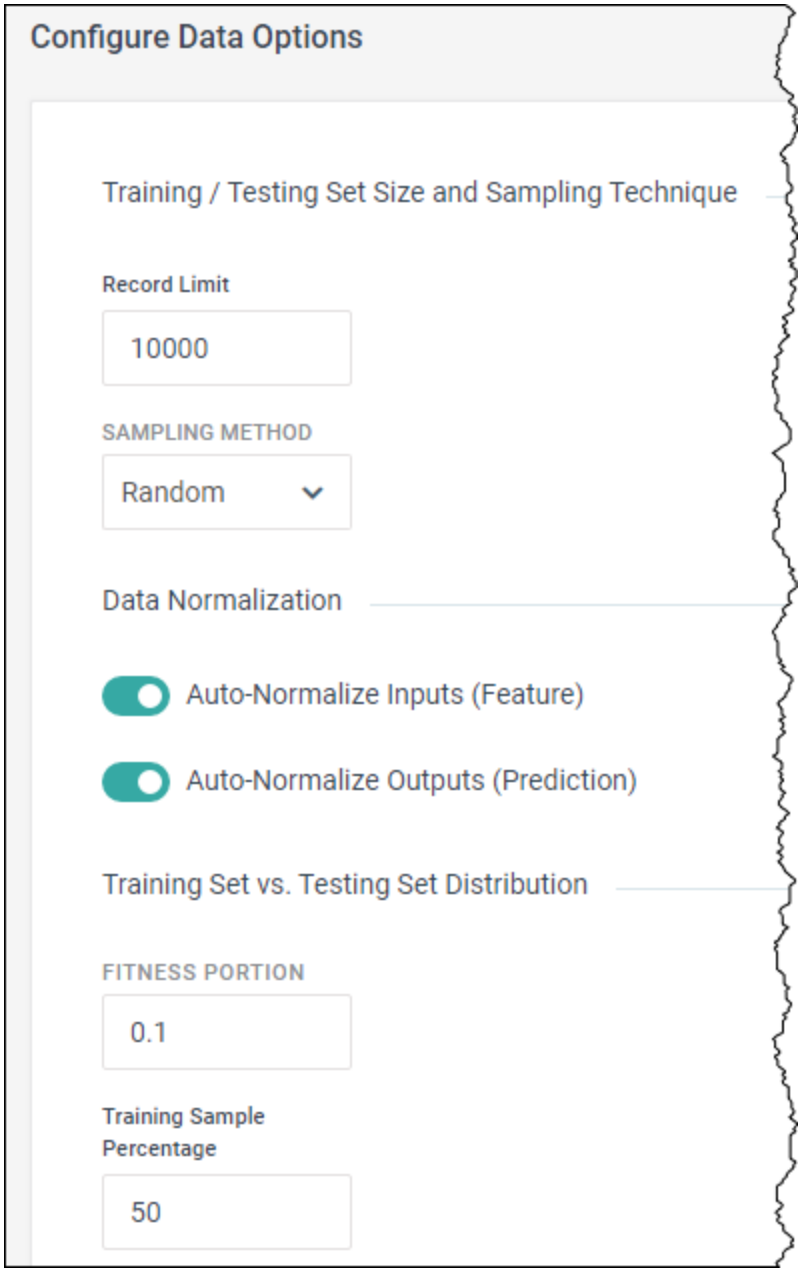
12.7.1.1.1 Training Options

Each model type has a set of associated algorithm options. The AML optimization engine uses each selected algorithm in its search for the best solution. By default, all available, applicable algorithms are selected. At least one algorithm must be selected in order for this project section to be valid, and you can select as many as you like.

Each algorithm has its strengths and weaknesses. Selecting more than one model algorithm for the project gives the optimizer the freedom to find the “best” algorithm (and parameterizations thereof) for the data. (That is, selecting more than one model algorithm gives the optimizer more degrees of freedom to find something that works well.) Selecting a single model algorithm potentially constrains the search to only sub-optimal model algorithms.

See Training algorithms 95 for more information on the algorithm options.

12.7.1.1.2 Data Options



Record Limit

If you want to use less than the full number of records in the data set for training and testing, enter the number of records to use in the **Record Limit** box.

Sampling Method

The **Sampling Method** menu lets you choose how to use records from the data set. The values are:

- **Random**—Use records from the data set in a random order.

- **Nth Record**—Use every nth record from the data set. N is automatically calculated based on the total number of available data records and the number of records selected for use in the model.
- **Sequential**—Use the records from the data set in sequential order (the order the records appear in the file).

Data Normalization

This section lets you decide if you want to auto-normalize data values in input and output features. By default, data normalization is turned on for input and output features.

Situations where you may not want to normalize your data include:

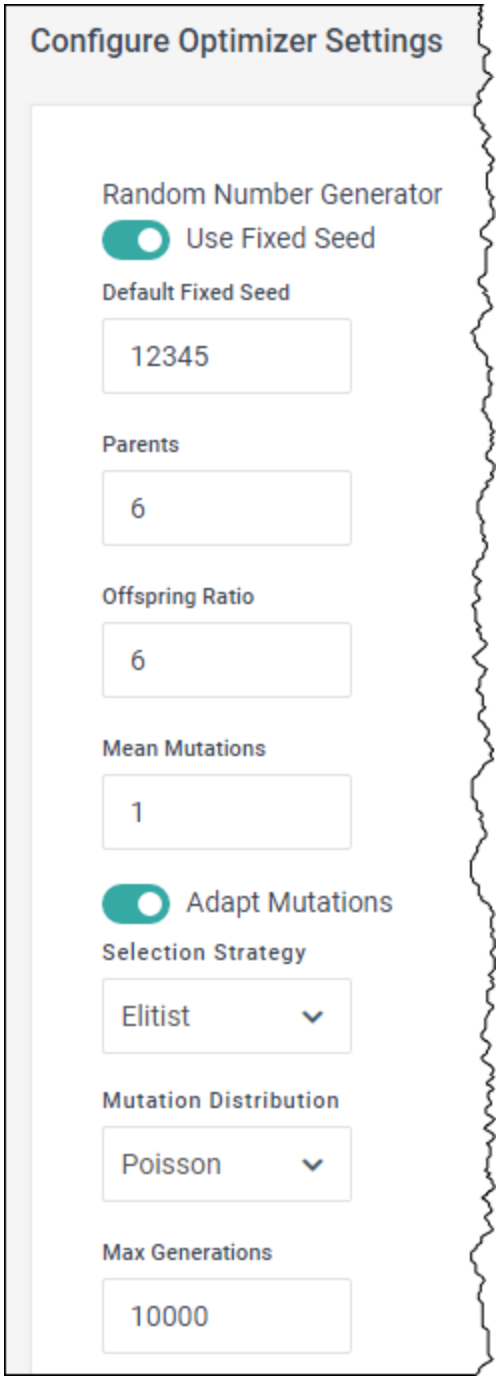
- Re-creating a (pre)existing model's functionality.
- For certain algorithms (such as decision trees and random forests) normalization is not always a good choice as it can seriously distort the resultant model performance.

Training set vs. testing set distribution

This section lets you set the **Fitness Portion** and **Training Sample Percentage** values.

If you want to know the details of what these values are and how they are used, read How AML calculates a solution fitness score 93.

12.7.1.1.3 Optimizer



AML includes an optimizer component that automatically attempts to refine and tweak your model during each iteration of training. The optimizer normally works autonomously, but you can influence the direction of the optimizer refinements by changing the settings in this section.

Random Number Generator

Select **Use Fixed Seed** to ensure that the same sequence of random numbers is used every time the model is trained. This ensures that every model training (that uses the same starting parameters) produces the same results. In addition, you can change the **Default Fixed Seed** value.

There is a lot of random number generation in ML. Using the same fixed seed for random number generation ensures that when comparing solutions from different training runs, you are comparing apples to apples.

Parents

How many parents are used That is, the number of starting solutions used to base offspring on during each iteration.

Offspring Ratio

How many offspring per parent. That is, the number of variant solutions (aka *offspring*) created per parent during each iterative cycle.

Explicitly choosing the number of parents and offspring, or changing the values, won't change the result. It just optimizes how quickly you get the answer. These values should be as large as possible without maxing out the machine's resources.

Mean Mutations

Index of how much mutation you want to occur.

Adapt Mutations

Select **Adapt Mutations** to enable automated meta-level, self-adaptive mutation to speed the evolutionary search.

Selection Strategy

Select **Elitist** to simply pick the top *n* models.

In **Tournament** selection, the best of the best will always win the tournament, but who comes in second, third, fourth (and so on) is dependent on how each pairing competition plays out.

A parent may have a desirable feature, but depending upon how it fares against specific competitors, this desirable feature may not be passed on to offspring.

Mutation Distribution

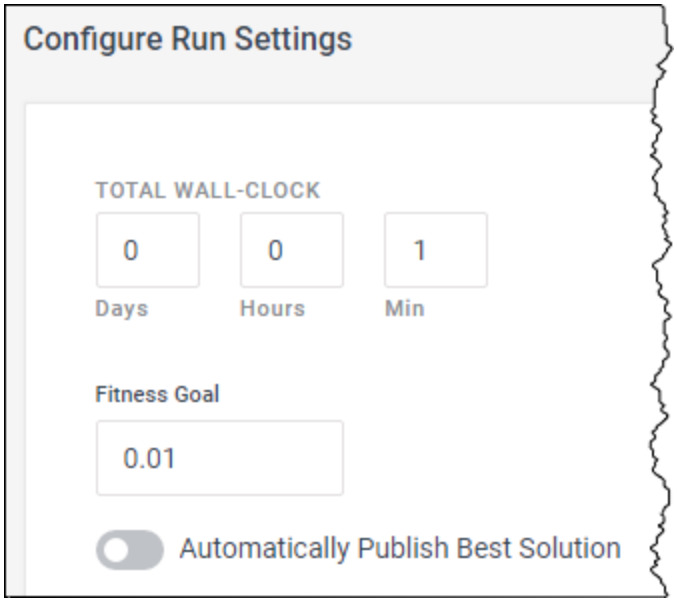
How is the number of mutations chosen (fixed or random value).

Max Generations

The desired maximum number of evolutionary search iterations.

12.7.1.1.4 Run Settings

Non-Product Recommender projects



Total Wall-Clock

The maximum training time limit.

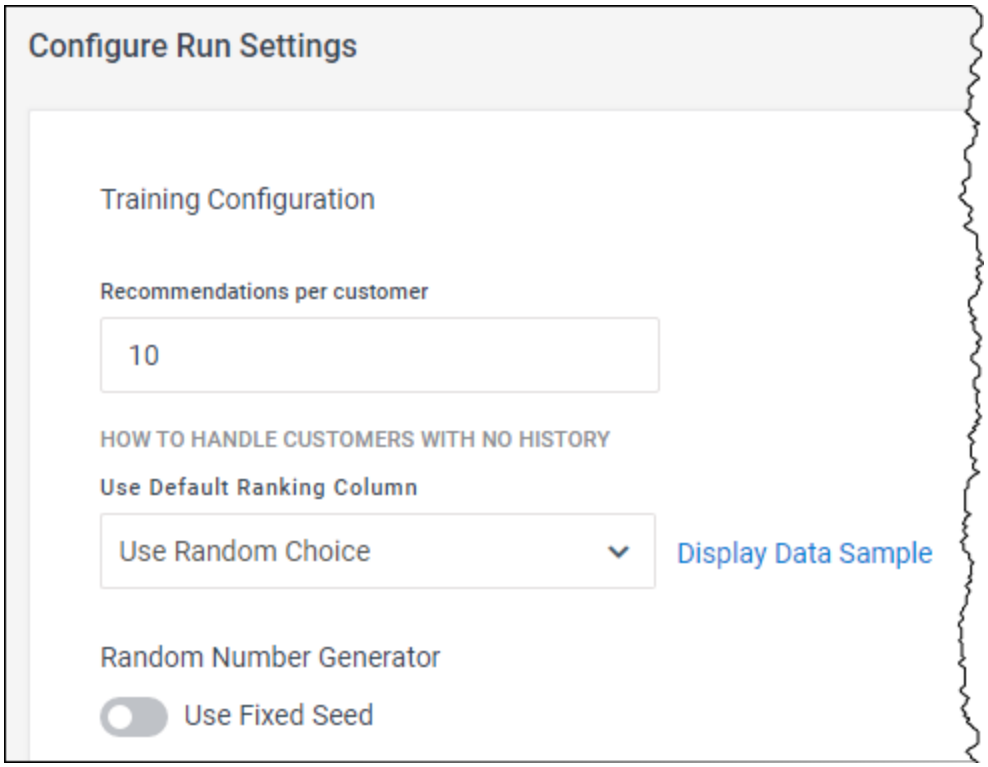
Fitness Goal

This is the solution fitness goal. If a solution's score meets or exceeds this value, the solution is considered "good enough". Since a project usually generates multiple solutions, it's handy to know which solutions are "keepers".

Automatically Publish Best Solution

Select **Automatically publish best solution** if you want AML to automatically publish the best (highest-scoring) solution at the end of model training.

Product Recommender projects



Recommendations per customer

The number of product recommendations generated for each customer.

How to Handle Customers with No History

Determines how you want to generate recommendations for customers with no history data in the model.

If the data set has no integer fields, there are no ranking choices other than **Use Random Choice** (generate customer recommendations randomly).

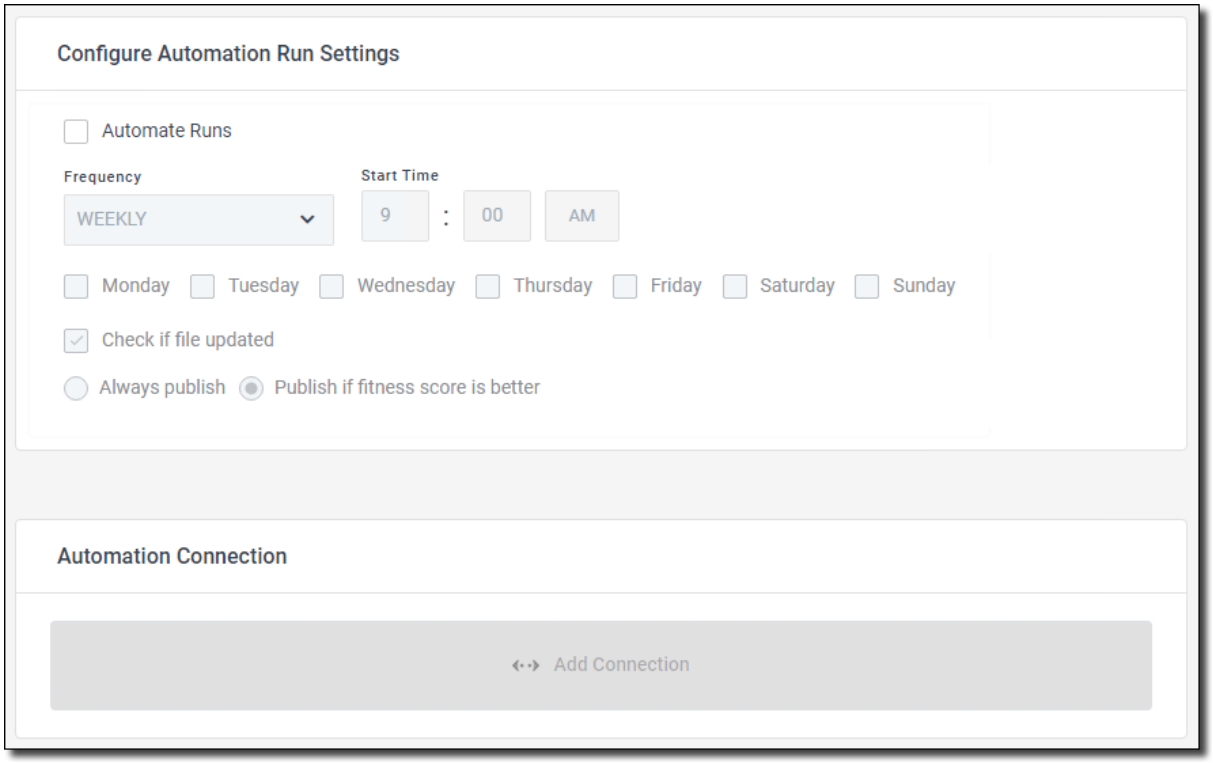
If the data set has integer fields, you can choose one of these fields from the pulldown menu to suggest a ranking of products/offers.

Random Number Generator

Select **Use Fixed Seed** to ensure that the same sequence of random numbers is used every time the model is trained. This ensures that every model training (that uses the same starting parameters) produces the same results.

There is a lot of random number generation in ML. Using the same fixed seed for random number generation ensures that when comparing solutions from different training runs, you are comparing apples to apples.

12.7.1.1.5 Automation Run Settings



This page allows you to automate project training. Instead of triggering a project training manually, you can configure any project to automatically retrain according to the parameters you set on this page.

If you want to automate this project's training, select **Automate Runs** and set the training **frequency**, **start time**, and **day**.

Note

You cannot change any of the settings on this page until you first select **Automate Runs**.

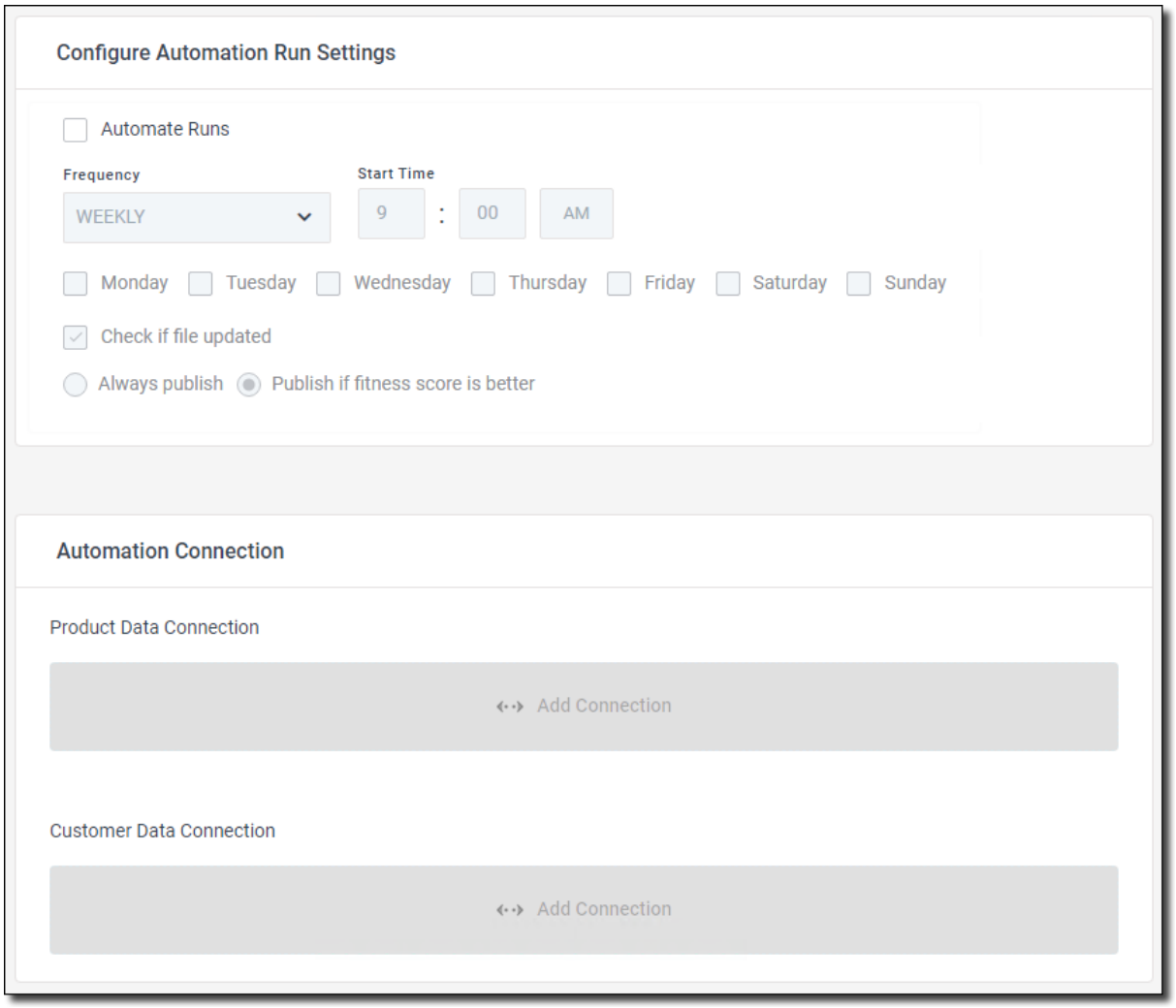
Select **Check if file updated** if you want to check whether the data file has been updated since the last training. If this option is selected, the project will retrain at the specified time only if the data file has been updated.

If you want to publish the highest-scoring solution only if it is better than the previous high-scoring solution, select **Publish if fitness score is better**. Alternately, you can **Always publish** the highest-scoring solution at the end of every training.

Automation Connection

This section allows you to add a predefined Data Connection, or displays the Data Connection already attached to the project. See To create a Data Connection 35 and To automate file uploads via a Data Connection 40 for instructions.

Note that for Product Recommender projects, you can add two Data Connections (one for product data, another for customer data).



12.7.2 Classification projects

This section provides additional information on Classification project settings and UI functionality.

12.7.2.1 The test data set

The file `classification-example-data-file.csv` contains a number of data fields commonly found in client marketing databases. As with any data set, the amount of data (number of records) is usually less important than the information contained within these records. Good information content leads to higher accuracy and reliability.

Some of the records contain missing data elements (missing data is a common problem with real-world data sets). Modeling with missing data is possible, though most machine learning (ML) algorithms require substitution or estimation of these missing data elements to prevent errors in the learning process. There are multiple user-controlled options within AML to perform missing data substitution, and general default values have been established for each data type.

The following table describes the `classification-example-data-file.csv` fields:

Field Name	Type	Comments
RowNum	integer	Row number of the table. Not useful as either an input or output feature.
CustomerRating	integer	Useful as both an input or output feature.
Retention	integer	Particularly useful as an output feature. Possibly could be used as an input feature.
LoyaltyMember	boolean (yes/no)	Useful as an input or output feature.
DaysSinceLastPurchase	float	Useful as an input or output feature.
ImmediateRelatives	integer	Useful as an input or output feature.
HouseholdChildren	integer	Useful as an input or output feature.
CustomerID	alphanumeric text	Useful as a key, not as an input/output feature.
LatestPurchasePrice	float	Useful as an input or output feature.
LatestPurchaseItemID	text	Useful as an input feature.
RegionCode	text	Useful as an input feature.
CustomerCaptureMethod	alphanumeric text	Useful as an input feature.
CustomerContactCode	integer	Useful as an input feature.
Domicile	text	Generally not useful as an input or output feature, though state, city, or county might be used as input features.

12.7.2.2 Creating a predictive model

A common need for marketing companies is to be able to predict customer retention based on historical behaviors. The modeling task becomes a retention/attrition classification problem where the model predicts whether or not a customer is expected to be retained. The model predicts probability of retention using the `Retention` data field as its output (prediction) feature. Results will be mapped to `yes` (indicating predicted retention) or `no` (predicted attrition) for each input row of customer data. For the input features (predictors), many of the other data fields are suitable candidates.

Note

This predictive Classification model does not use the retention/attrition data field as an input feature since it has already been selected as the predicted output.

For this Classification model, we will use the following input features:

Input Feature Name	Rationale
CustomerRating	Often correlated with repeat customers, makes for a good predictor.
LoyaltyMember	Often correlated with long-term customers (a good predictor).
DaysSinceLastPurchase	Indicates recency. Customers with recent purchases are likely to be repeat customers. ***Note*** This may not be a good predictor for durable goods (cars, sofas, washing machines, and so on) although customers may eventually become long-term clients.
ImmediateRelatives	Households with close relatives, spouses, and so on tend to be correlated with long-term product/brand/merchant relationships.
HouseholdChildren	Households with children tend to be correlated with long-term product/brand/merchant relationships.
LatestPurchasePrice	Potentially correlated with long-term merchant relationships.
LatestPurchaseItemID	Potentially correlated with long-term merchant relationships.

Input Feature Name	Rationale
RegionCode	Potentially correlated with long-term merchant relationships.
CustomerCaptureMethod	Often correlated with long-term merchant relationships.
CustomerContactCode	Often correlated with long-term merchant relationships.

12.7.2.3 Feature actions

While adding input and (possibly) output features to a project, there are a number of actions you can take to view feature data, manipulate how feature data is interpreted, and delete a feature.

View and change column enumerated type

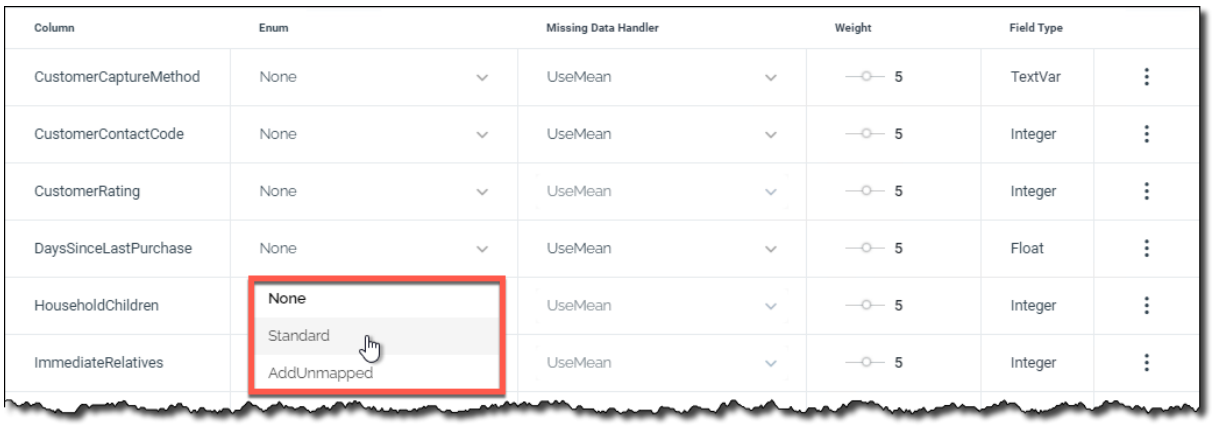
Note

Changing an input feature's enumerated type is an advanced option. In most cases, AML automatically chooses enumerated types that do not need to be changed.

In order for AML to map a relationship between input and output values, these values must be numeric. For input values that are not numbers (for example, a text state abbreviation such as "AL" or "AK"), AML must convert these text values to enumerations.

To continue with the previous example, for state abbreviations "AL" could be assigned a value of 1, "AK" a value of 2, "AZ" a value of 3, and so on. This is an example of *standard enumeration*.

Input features that use numeric values are not converted to enumerations, and the corresponding value in the **Enum** menu is **None**.



If you want to train using an input feature that uses data that may not be representative of the whole world of possible choices, you can choose the **Enum** menu value **Add Unmapped**.

For example, you might have a training data set that contains only the text values "apple", "banana", and "orange", but you know that "lemon" and "grapefruit" are also possible values. In

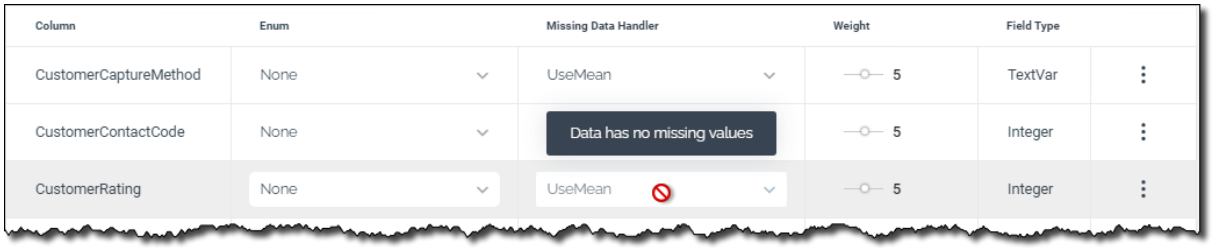
this case, for this input feature you would choose the **Enum** menu value **Add Unmapped**. This setting allows for an optional *other* category (a place to hold something that was not seen in training).

Another use of enumeration is to control the weighting of values. For example, if a feature contains numbers of vastly different size and you want to give all the values equal weight, you would choose the **Enum** menu value **Standard**.

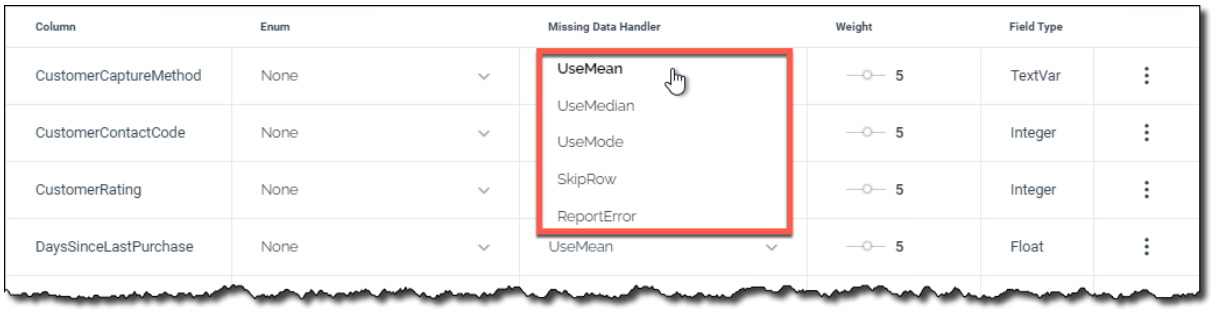
View and change action taken when column data is missing

Sadly, it is not uncommon that data is missing from a data set. However, you can decide what action AML should take when it does find missing data.

If no data is missing from a data column, the **Missing Data Handler** value is greyed out, and hovering over the value displays the message “Data has no missing values”.

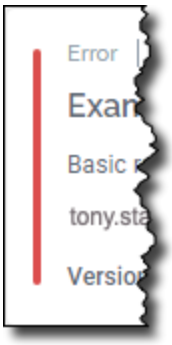


If a column does having missing values, you can change the value of the **Missing Data Handler** column (that is, you can change the action AML takes for missing data).



The **Missing Data Handler** values are:

- **Use Mean**
- **Use Median**
- **Use Mode**
- **Skip Row**
- **Report Error**—If you select **Report Error** for a feature, when AML discovers that data is missing for that feature during project training, the project stops training and on the **Project Manager** page, the project's status bar turns red and displays the status **Error**.

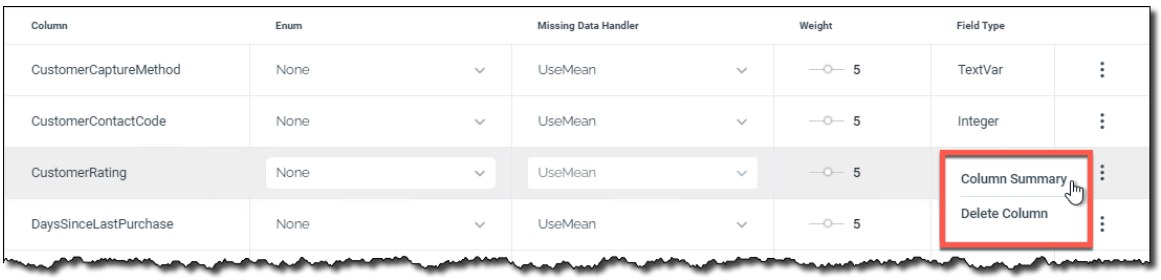


The default **Missing Data Handler** value is **Use Mean**.

View statistical information about a column

When evaluating whether to include a table column in the project, you can view a summary of the column data.

1. At the right side of a column row, hover over the pulldown menu and select **Column Summary**.

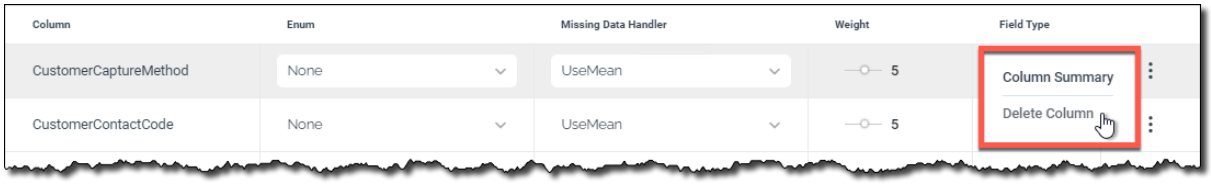


2. A popup dialog of statistics about the column data is displayed.

Delete a feature

After adding a feature to a project, you can later delete it if you change your mind.

At the right side of a column row, hover over the pulldown menu and select **Delete Column**.



View sample data

Sometimes it's useful to look at the actual data in a data set you're working with. For example, a data modeler may want to fine-tune which input or output columns to include in a project.

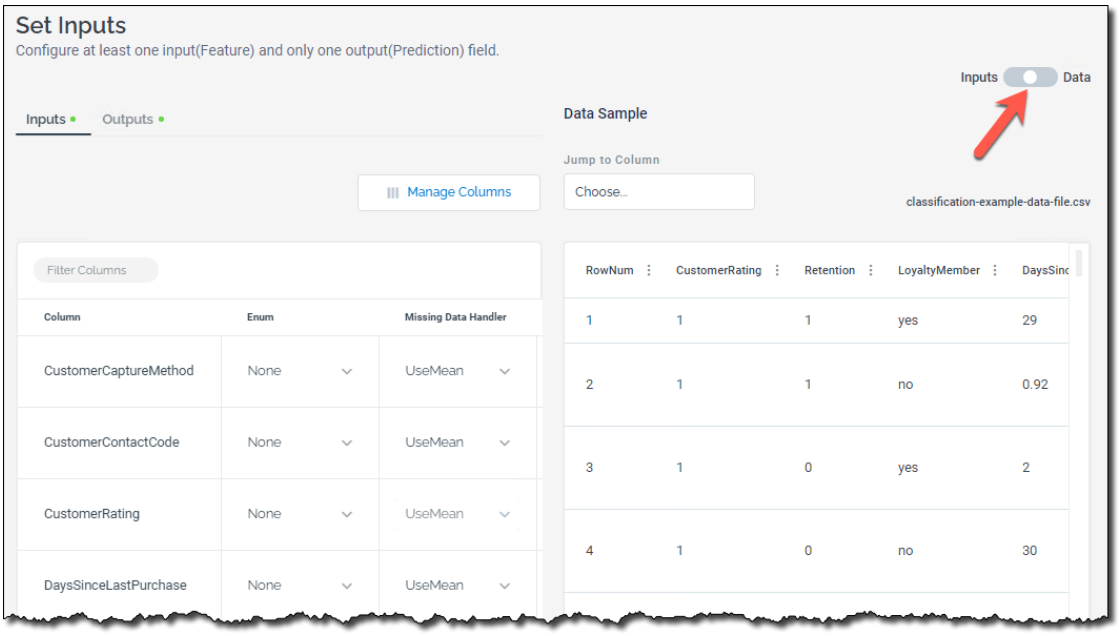
Note

This feature displays the first 200 rows of a data set.

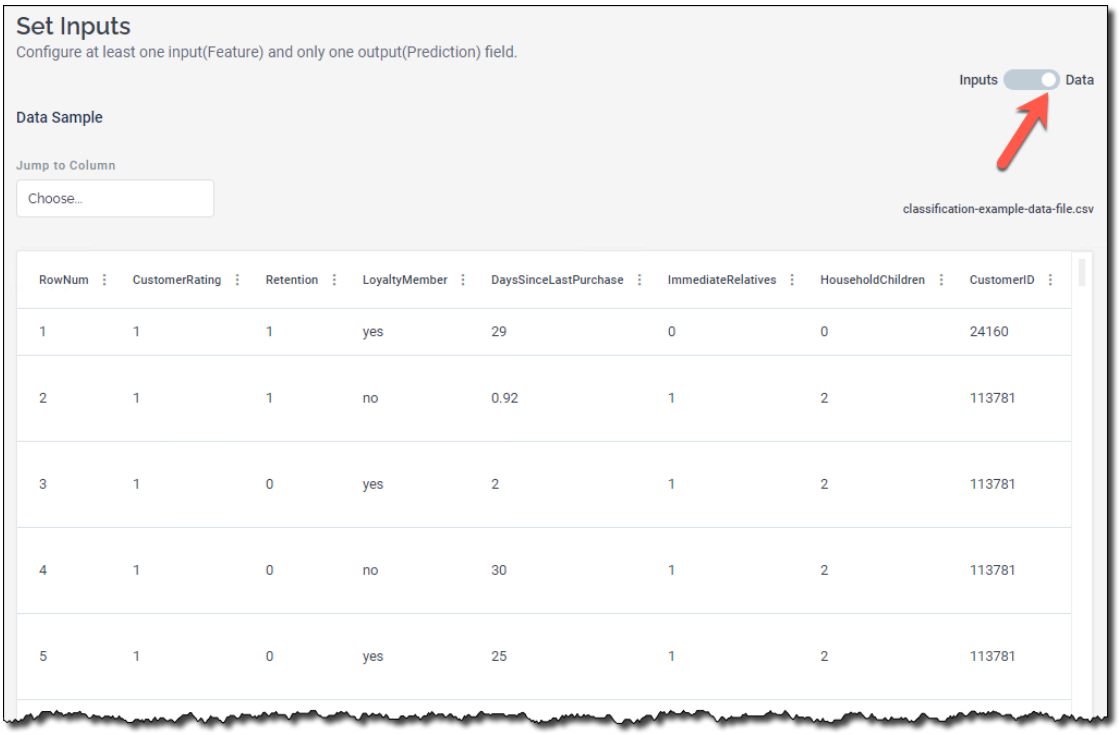
1. On a project's **Configure Project** page, click **Edit** on the **Inputs/Outputs** section.

2. On either the **Inputs** or **Outputs** tab, click the **Inputs/Data** toggle away from **Inputs**.

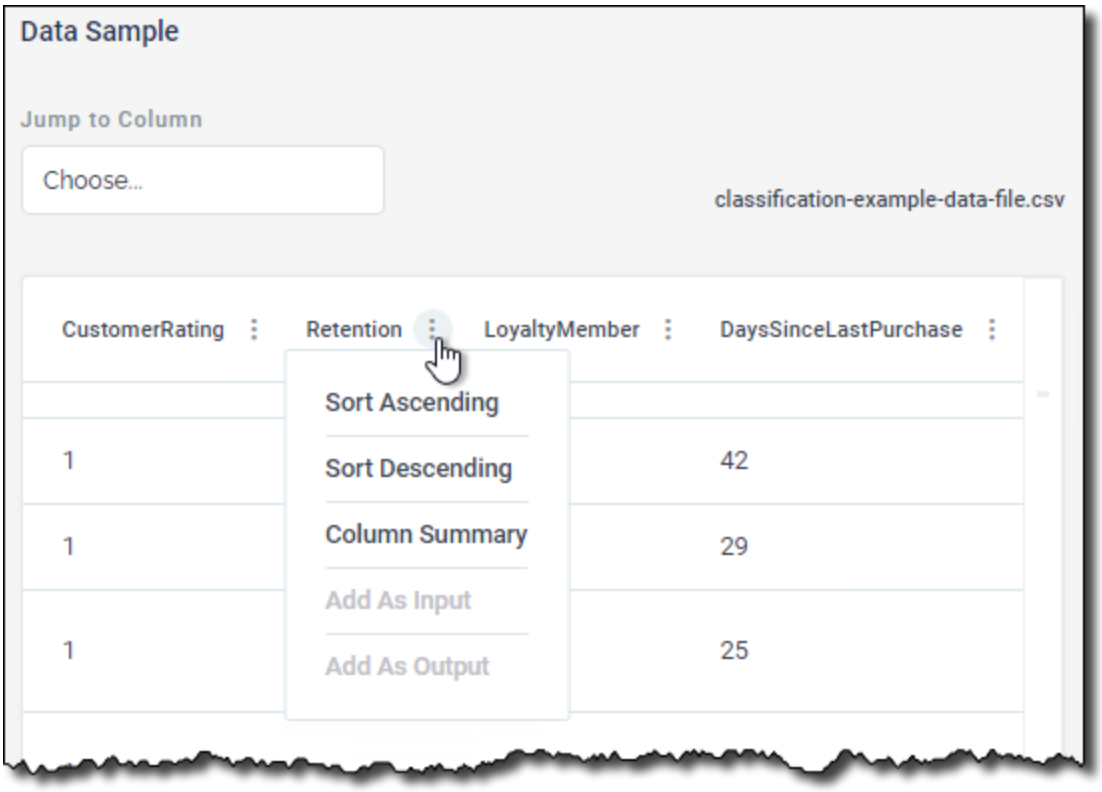
   - If you click the toggle halfway between **Inputs** and **Data**, both the (Input or Output) **Feature** table and the **Data Sample** table (a sample of records from the data file) are displayed.



   - If you click the toggle next to **Data**, only the **Data Sample** table (a sample of records from the data file) is displayed.



3. In the Data Sample table, you can horizontally scroll to view all record columns, or vertically scroll to view different records.
4. To filter columns in the **Feature** table, enter text in the **Filter Columns** box. Columns with names that match the entered text are displayed.
5. To jump to a specific column in the **Data Sample** table, click inside the **Jump to Column** box and either select a column from the popup menu or enter text and select one of the displayed columns whose name matches the text.
6. In the **Data Sample** table you can sort any column according to its data by hovering over the pulldown menu to the right of the column name and selecting **Sort Ascending** or **Sort Descending**.



7. To view a statistical summary of column data, hover over the pulldown menu to the right of a column name and select **Column Summary**.
8. To add a column as an input or output feature, hover over the pulldown menu to the right of a column name and select **Add as Input** or **Add as Output**.

Note

The ability to add a column as an input or output feature is disabled if AML determines the column is ineligible to be considered for inclusion. This usually happens because the column contains text fields and AML enforces a rule in which text columns that contain 25 or more unique values cannot be used as input or output to a project.

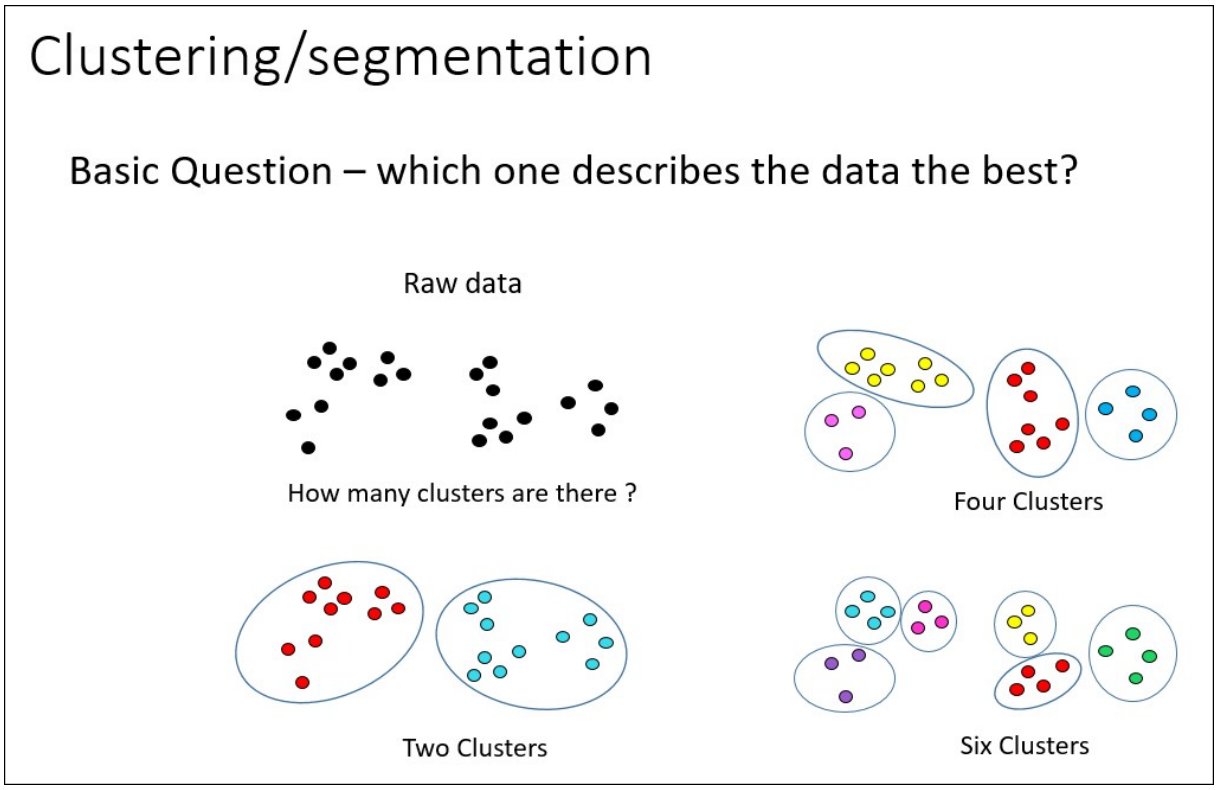
12.7.3 Clustering projects

This section provides additional information on Clustering project settings and UI functionality.

12.7.3.1 What does the Clustering model do?

Cluster/segment analysis (AKA *clustering/segmentation*) is the task of grouping a set of objects in such a way that objects in the same group (called a cluster or segment) are more similar (in some sense) to each other than to those in other groups (clusters/segments).

Clustering/segmentation is done by using *dimensional* data. A data dimension is a set of data attributes pertaining to something of interest. Dimensions are things like customers, products, stores, or preferred color. AML assigns records to clusters based on dimensional data. In AML, a data dimension is also known as an input *feature*.



Cluster/segment analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their understanding of what constitutes a cluster/segment and how to efficiently find them. Popular notions of clusters/segments include groups with small distances between cluster/segment members, dense areas of the data space, intervals or particular statistical distributions.

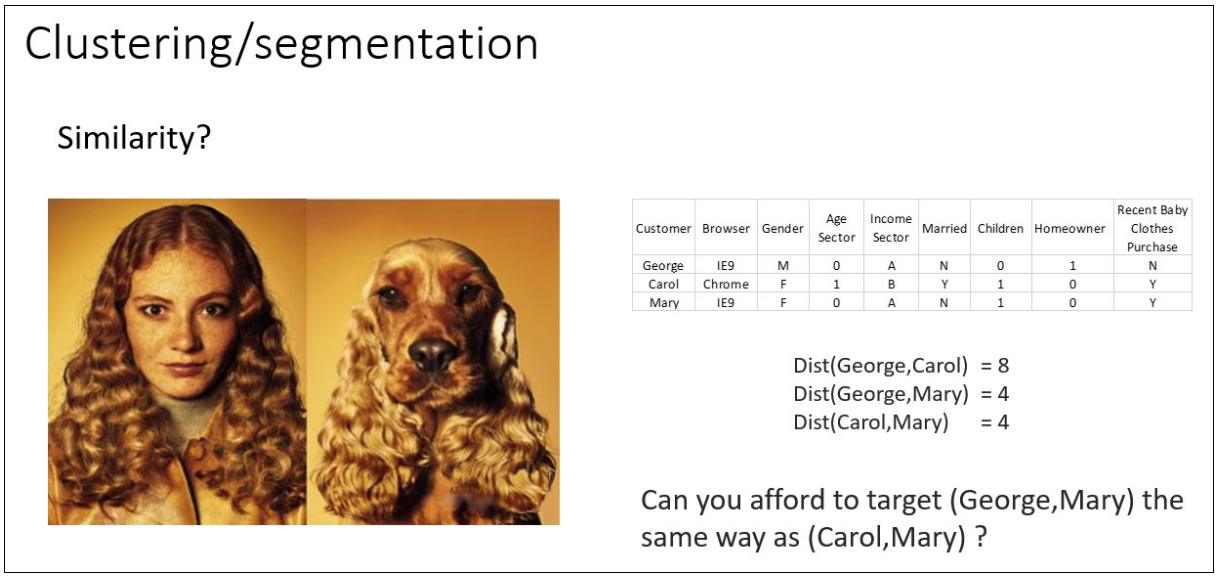
The appropriate clustering/segmentation algorithm and parameter settings (including parameters such as the distance function to use, a density threshold, or the number of expected clusters/segments) depend on the individual data set and intended use of the results. Cluster/segment analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It is often

necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

A common use of clustering/segmentation is dividing users, customers, or subscribers into clusters/segments of individuals based on similarities that may be relevant to your marketing. While traditional clustering/segmentation groups individuals simply by age, gender, income, and so on, cluster/segment analysis can identify clusters/segments using many more dimensions. One can perform cluster/segment analysis on customer attributes to answer questions such as:

- What are the demographic characteristics of my best customers?
- How do customers behave while purchasing?
- What groupings of products do people buy from?
- How many clusters/segments best describe this data? (For example, three clusters/segments, five, six, ten?)

When doing clustering/segmentation without proper analysis (AKA *manual clustering/segmentation*), a common mistake is to use inaccurate, invalid, or inappropriate assumptions about what should and shouldn't be clustered/segmented together. For example, a common demographic assumption is to create standardized age buckets (for example, 18-25, 26-35, 36-55, 55-80, and so on). These age grouping are decided without determining if the data supports them. Clustering/segmentation with machine learning algorithms creates appropriate clusters/segments suggested by the underlying data.



In addition to answering specific questions about your customers, clustering/segmentation can also be used to analyze your customers and their behavior. By looking at the clustering/segmentation of customers based on different dimensions, you can discover underlying connections between customers that may not be immediately obvious.

For example, in marketing standardized age levels such as 20-30 years, 30-40 years, and so on are often used by default. By training a cluster model, you can find more effective age groupings that are suggested by your customer data.

As another example, in sales standardized purchase levels are often used:

- Time—For example, 3, 6, or 9 months
- Money—For example, 0-49 dollars, 50-99 dollars, 100-499 dollars

By training a cluster model, you can find more effective time and money groupings that are suggested by your customer data

Instead of a model suggesting the number of output clusters, you may want to define the number of output clusters. For example, if your company has only four products, you may want only four clusters. In this case, training a cluster model may make a better decision about what goes in each of the four clusters than a human can.

Automation is a useful tool to use with clustering/segmentation. For example, you can set AML to be triggered when a new customer is added to the data set and to immediately calculate what cluster/segment to which the new customer is assigned.

12.7.3.2 Thoughts on cluster size

A cluster size of 1 means that everything is in the same cluster, which doesn't tell you anything.

If the number of clusters of a solution is the same as the maximum number of clusters setting, here are some things to consider:

- The true number of clusters could be much higher than the max cluster setting.
- Increasing the max cluster setting and training the model again could give you a better result.
- The data set may not be a good fit for a Clustering model.

12.7.3.3 Clustering model algorithms reference publications

The following publications describe the cluster model algorithms in detail.

K-Means

Z. Ansari, M.F. Azeem, Waseem Ahmed, A.V. Babu, "Quantitative Evaluation of Performance and Validity Indices for Clustering the Web Navigational Services", *World of Computer Science and Information Technology Journal (WCSIT)*, Vol. 1, No. 5, pp 217-236, 2011

C. Ding, X. He, "k-Means Clustering via Principal Component Analysis", Proc. 21st Int. *Conference on Machine Learning (ICML 04)*, Banff, Canada, 2004

K-Medians

Q. Zhao, "Clustering Validity in Clustering Methods", Masters Thesis, *Publications of the University of Eastern Finland*, No. 77, 2012

K-Modes

Chaturvedi, P. Green, J. Carroll, "K-Modes Clustering", *Journal of Classification*, Vol. 18, pp. 35-55, 2001 DOI: 10.1007/s00357-001-0004-1

N. Sharma, N. Gaud, “K-Modes Clustering Algorithm for Categorical Data”, *Int. Journal of Computer Applications*, Vol. 127, No. 17, Oct. 2015

K-Medoids

Z. Ansari, M.F. Azeem, Waseem Ahmed, A.V. Babu, “Quantitative Evaluation of Performance and Validity Indices for Clustering the Web Navigational Services”, *World of Computer Science and Information Technology Journal (WCSIT)*, Vol. 1, No. 5, pp 217-236, 2011

12.7.3.4 Clustering measure functions

The following topics provide reference information for the Clustering measure functions.

12.7.3.4.1 Semi-supervised Clustering measure functions

These cluster measure functions require the user to specify an output “truth” feature, as during creation of Classification models:

- Rand Index
- Adjusted Rand Index
- Variation of Information (VI)
- Normalized Variation of Information (NVI)
- V-Measure
- Normalized Information Distance (NID)

12.7.3.4.2 Unsupervised Clustering measure functions

These cluster measure functions do not require the user to specify an output “truth” feature:

- Davies-Bouldin Index
- Dunn Index
- Silhouette Index

12.7.3.4.3 Clustering measure functions reference publications

The following publications describe the cluster measure functions in detail.

Adjusted Rand Index (ARI)

L. Hubert and P. Arabie, “Comparing partitions”, *Journal of Classification*, Vol. 2, pp. 193–218, 1985

Davies-Bouldin Index

D. Davies and D. Bouldin, “A Cluster Separation Measure". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. PAMI-1 Vol. 2, pp. 224–227, 1979

Dunn Index

J. C. Dunn, " A Fuzzy Relative of the ISODAT Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics*. Vol. 3 (3), pp. 32–57, 1973

Normalized Information Distance (NID)

S. Terwijn, L. Torenvliet, P. Vitanyi, "Nonapproximability of the normalized information distance", *Journal of Computer and System Sciences*, vol. 77, pp738-742, 2011

R. Cilibrasi and P. Vitanyi, "Clustering by Compression", (corrected version) *IEEE Trans. Information Theory*, Vol.. 51, No. 4, pp 1523-1545, April, 2005

Normalized Variation of Information (NVI)

R. Reichart and A. Rappoport, "The NVI Clustering Measure", Proc. *Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pp. 165–173, Boulder, Colorado, June 2009, Association for Computational Linguistics

Rand Index (RI)

William Medden Rand, "The development of objective criteria for evaluating clustering methods ", PhD dissertation, UCLA, 1969 (Dept. of Biostatistics, UCLA library microfiche).

W. Rand, "Objective criteria for the evaluation of clustering methods ", *Journal of the American Statistical Association*. American Statistical Association. 66 (336): 846–850, 1971

L. Vendramin, R. Campello, E. Hruschka, Relative Clustering Validity Criteria: A Comparative Overview, *Statistical Analysis and Data Mining*, Vol. 3, pp. 209-235, 2010

Silhouette Distance

P.J. Rousseeuw, "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis", *Computational and Applied Mathematics*. Vol. 20, pp. 53–65, 1987

Variation of Information (VI)

M. Melia, "Comparing Clusterings – an information based distance", *Journal of Multivariate Analysis*, Vol. 98, No. 5, pp. 873-895, May, 2007

V-Measure

A. Rosenberg, J. Hirschberg, "V-Measure: A conditional entropy-based external cluster evaluation measure", *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 410-420, 2007

12.7.4 Regression projects

This section provides additional information on Regression project settings and UI functionality.

12.7.4.1 About the Regression model

The Regression model uses a set of input features to predict output features. Input features can be continuous (floating point), discrete (integer), text (turned into enumerated integer values), and binary (0/1). The output features to predict must be continuous values.

Regression models can be used to predict things like:

- profit
- potential sales
- transaction volume
- credit score

A Regression model performs supervised learning, meaning that models are trained with known results that are presumed to be correct (and most importantly, correlated with the input features).

Multiple outputs are independent of each other.

Note

Only certain training algorithms (for example, neural networks) can produce multiple output values simultaneously.

12.7.5 Product Recommender projects

This section provides additional information on Product Recommender project settings and UI functionality.

12.7.5.1 About the Product Recommender model

PR utilizes only a customer's history (that is, a customer's preferences) to recommend other products. It does not use other people's preferences to generate product recommendations (as with Amazon).

For the other model types in AML, training a model generates many possible solutions. However, in the PR model, all the necessary work (training and output creation) is done in one step, so only one solution is generated.

Also (unlike other AML model types) in a PR model, the end results are the same for every customer, no matter how many times the model is trained. Therefore, (unlike other AML model types) the best way to use the PR model is to train it once and create one recommendation file.

Though the PR model works differently from the other AML models, you can call a published solution using the same APIs as with the other models.

Note that customer preference information is optional (customer buying history for the products is all that is absolutely necessary).

12.8 AML API

Anything that you can do through the web app, you can do by calling AML API endpoints. Furthermore, the AML API includes functionality not included in the UI.

12.8.1 How to access the AML API documentation

We use the open-source Swagger toolset to automatically generate the AML API documentation.

Your URL for the AML API docs is:

`https://linux-hostname:8983/docs/index.html`

or

`http://linux-hostname:8980/docs/index.html`

12.8.2 Add an app admin user account

When an instance of AML is deployed, it has only two associated user accounts:

- An app admin user with the username "admin".
- A system admin user with the username "system".

Additional app admin and system admin users can be created only through the AML API. This topic explains how to add an app admin user account using the API endpoints.

Prerequisites

Before doing this procedure, deploy (successfully) all of the DCC containers.

Access DCC service APIs to configure user accounts with your local web browser

Navigate to the AML API docs using a web browser:

`https://linux-hostname:8983/docs/index.html`

or

`http://linux-hostname:8980/docs/index.html`

When you drill in to a specific API endpoint, a "Try Me" option (in which you can manually exercise the API) is displayed.

Log in to AML

Log in as the user "system" and use the password "system":

```
POST /api/v1/auth/signon-admin
  {
    "username": "system",
```

```
    "password": "system"
  }
```

Copy returned authorization token

An authorization token is returned. Copy this token for use in the next two endpoint calls.

Create a new user account

Use the following JSON payload to create the user "testuser". Use the authorization token that was previously returned.

```
POST /api/v1/auth/users
  {
    "username": "testuser",
    "password": "testuser!",
    "email": "testuser@nomail.com",
    "description": "",
    "firstname": "Test",
    "lastname": "User",
    "authType": "Default"
  }
```

Assign the new user to a group

Assign the user "'testuser'" as an app admin for AML.

The general call is:

```
POST /api/v1/auth/clients/clientid/apps/appname/add-admin/username
```

For this step, use the following values:

```
POST /api/v1/previously-granted-
token/clients/dcc_client_id_global/apps/machine_learning/add-admin/testuser
```

12.8.3 Add a system admin account

When an instance of AML is deployed, it has only two associated user accounts:

- An app admin user with the username "admin".
- A system admin user with the username "system".

Additional app admin and system admin users can be created only through the AML API. This topic explains how to add a system admin account using the API endpoints.

Prerequisites

Before doing this procedure, deploy (successfully) all of the DCC containers.

Access DCC service APIs to configure user accounts with your local web browser

Navigate to the AML API docs using a web browser:

`https://linux-hostname:8983/docs/index.html`

or

`http://linux-hostname:8980/docs/index.html`

When you drill in to a specific API endpoint, a "Try Me" option (in which you can manually exercise the API) is displayed.

Log in to AML

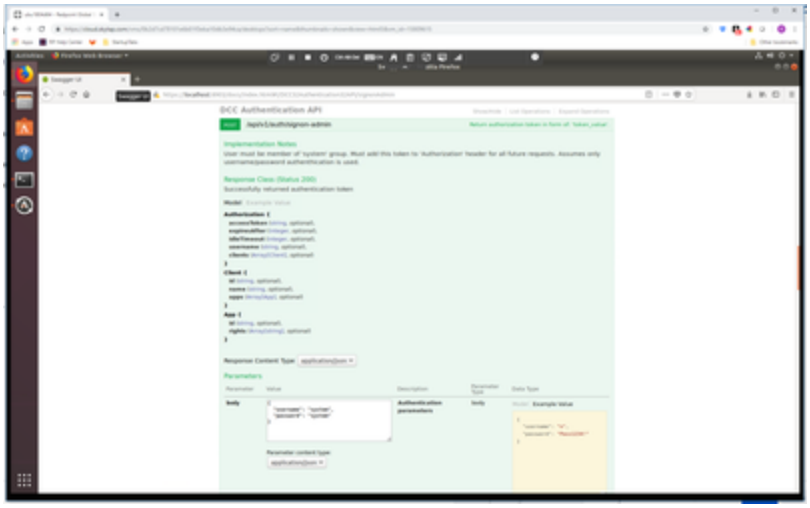
Log in as the user "admin". You can also log in as the default user "system".

If logging in as the user "system", use the password "system":

```
POST /api/v1/auth/signon-admin
  {
    "username": "system",
    "password": "system"
  }
```

Copy returned authorization token

An authorization token is returned. Copy this token for use in the next two endpoint calls.



Create a new user account

Note

You can skip this step if you are simply enabling system admin access for a pre-existing user account.

Use the following JSON payload to create the user "newadmin". Use the authorization token that was previously returned.

```
POST /api/v1/auth/users
  {
    "username": "newadmin",
    "password": "NewAdmin123!",
    "email": "newadmin@nomail.com",
    "description": "",
    "firstname": "New",
    "lastname": "Admin",
    "authType": "Default"
  }
```

Assign the new user to a group

Assign the user "newadmin" to the group "system" group. This gives the user access to the system admin page.

The general call is:

```
POST /api/v1/auth/groups/groupname/add-user/username
```

For this step, use the following values:

```
POST /api/v1/previously-granted-token/groups/system/add-user/newadmin
```

Grant the new user access to the AML app

Assign the user "newadmin" to the group "machine_learning". This gives the user access to the AML app.

The general call is:

```
POST /api/v1/auth/clients/clientid/apps/appname/add-user/username
```

For this step, use the following values:

```
POST /api/v1/previously-granted-
token/clients/dcc_client_id_global/apps/machine_learning/add-user/newadmin
```

Index

- A -

- C -

- D -

- F -

- G -

- H -

- I -

- L -

- M -

- O -

- P -

- R -

- S -

- T -

- U -

Redpoint Global Inc.

36 Washington Street, Suite 120
Wellesley Hills, MA 02481
+1 781 725 0250 Phone
+1 781 235 3739 Fax
www.redpointglobal.com