



# Data Science and Machine Learning Essentials

Lab 1 – Getting Started with Azure Machine Learning

## Overview

In this lab, you will learn how to open and navigate the Microsoft Azure Machine Learning (Azure ML) Studio. You will also learn how to create and run experiments in Azure ML.

**Note:** The goal of this lab is to familiarize yourself with the Azure ML environment and some of the modules and techniques you will use in subsequent labs. Details of how to visualize and manipulate data, and how to build and evaluate machine learning models will be discussed in more depth later in the course.

## What You'll Need

To complete this lab, you will need the following:

- An Azure ML account
- A web browser and Internet connection
- The files for this lab
- Optionally, the Anaconda Python distribution or R and R Studio if you want to edit the code examples give in this lab

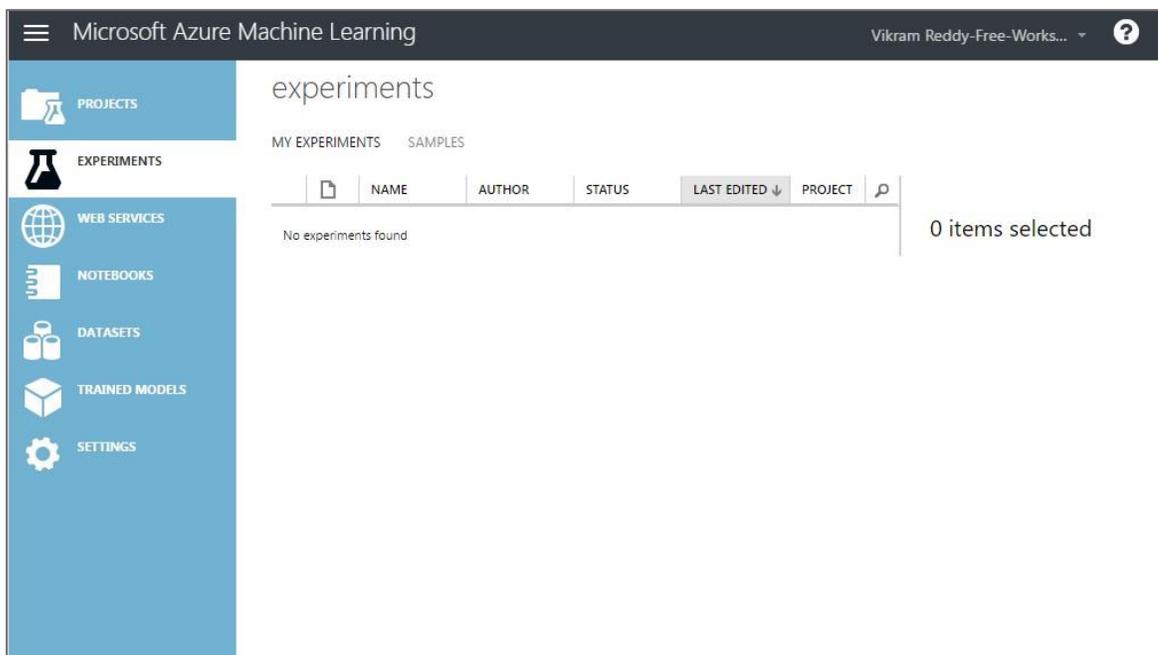
**Note:** To set up the required environment for the lab, follow the instructions in the Setup Guide for this course.

## Creating an Azure ML Experiment

Azure ML enables you to create experiments in which you can manipulate data, create predictive models, and visualize the results. In this exercise, you will create a simple experiment to explore a sample dataset that contains data on bank customers. Your goal is to predict the creditworthiness of these customers.

Sign into Azure ML Studio

1. Browse to <https://studio.azureml.net> and sign in using the Microsoft account associated with your free Azure ML account.
2. If the **Welcome** page is displayed, close it by clicking the **OK** icon (which looks like a checkmark). Then, if the **New** page (containing a collection of Microsoft samples) is displayed, close it by clicking the **Close** icon (which looks like an X).
3. You should now be in Azure ML Studio with the **Experiments** page selected, which looks like the following image (if not, click the **Studio** tab at the top of the page).



- **Tip:** To organize the labs for this course you can click **Projects** and create a new *project*. You can add your datasets and experiments to this project so they are easy to find in the future.

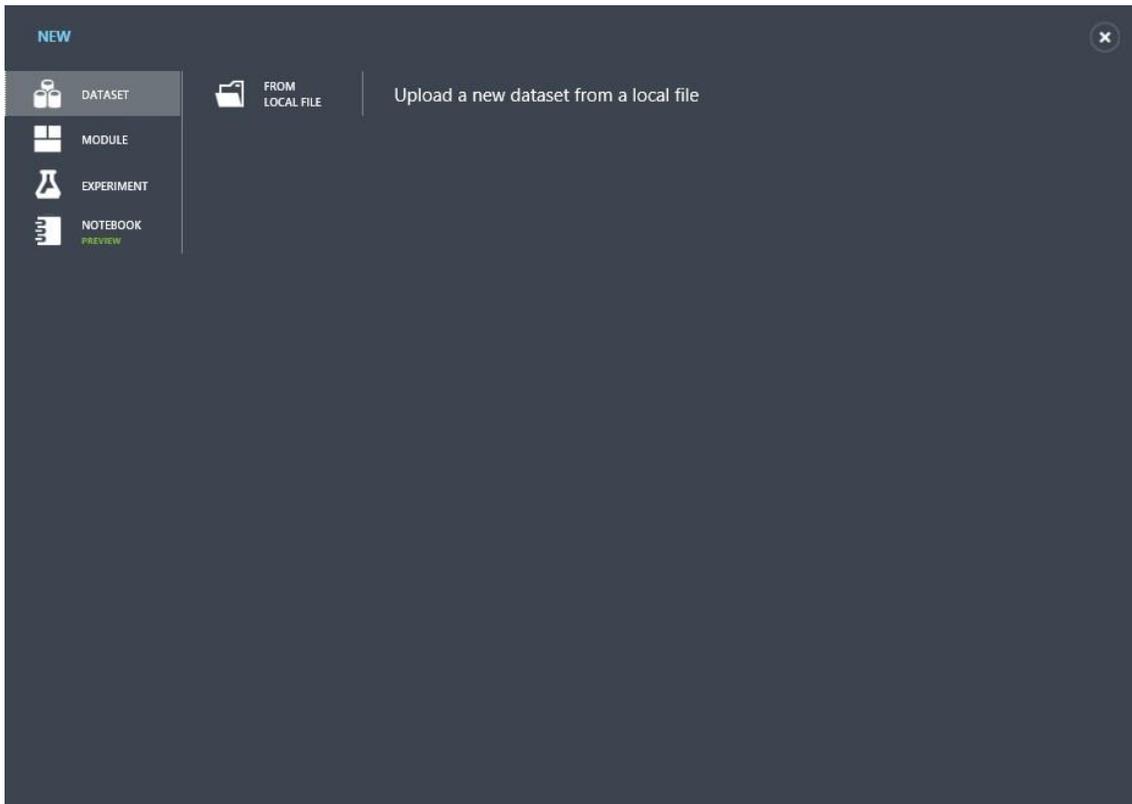
Upload and Visualize the Dataset

**Note:** The data set you will use in this lab has been cleaned and adjusted to make life easy for you while performing this lab. Later in this course, you will learn the important techniques required to clean and adjust data sets, and prepare them for analysis.

- From the folder where you extracted the lab files for this module (for example, C:\DAT203.1x\Mod1), open the **Credit-Scoring-Semi-Clean.csv** file, using either a spreadsheet application such as Microsoft Excel, or a text editor such as Microsoft Windows Notepad.
- View the contents of the **Credit-Scoring-Semi-Clean.csv** file, noting that it contains data on 944 customer cases. You can see the column headers for 21 columns; 20 features (data columns which can be used to train a machine learning model) and one label (the column indicating the actual credit status of the customers). Your data file should appear as shown here:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	CheckingAc	Duration	CreditHisto	Purpose	CreditAmo	Savings	Employer	Installment	SexAndStat	OtherDeter	PresentRes	Property	Age
2	A11	0.205882	A32	A43	0.16177	A61	A73	0.333333	A92	A101	0.333333	A121	0.0892
3	A14	0.294118	A32	A43	0.05838	A62	A74	1	A94	A101	0.666667	A122	0.1
4	A14	0.205882	A32	A43	0.069055	A61	A73	1	A93	A101	0.333333	A124	0.2321
5	A11	0.647059	A32	A43	0.358094	A61	A73	0.666667	A92	A101	0.333333	A123	0.2142
6	A13	0.029412	A33	A43	0.023825	A61	A72	0.333333	A92	A101	0	A122	0.1785
7	A11	0.029412	A32	A43	0.131892	A63	A73	0.333333	A93	A101	0.666667	A121	0.4464
8	A14	0.205882	A34	A46	0.088808	A62	A73	1	A92	A101	0.333333	A121	0.1964
9	A11	0.117647	A32	A48	0.035875	A61	A74	1	A94	A101	1	A122	0.0357
10	A11	0.029412	A34	A40	0.02289	A64	A74	0.666667	A92	A101	1	A121	0.3571
11	A14	0.161765	A32	A42	0.108452	A63	A73	0.333333	A92	A101	1	A123	0.0178
12	A14	0.470588	A32	A40	0.036261	A63	A75	1	A93	A101	1	A122	0.3035
13	A12	0.294118	A34	A43	0.053153	A62	A72	1	A93	A101	1	A124	0.3392
14	A12	0.205882	A32	A42	0.15137	A61	A74	0.333333	A92	A101	1	A121	0.3
15	A14	0.044118	A33	A43	0.032794	A65	A75	0.666667	A93	A101	1	A124	0.3035
16	A13	0.470588	A32	A43	0.217894	A61	A73	1	A93	A101	0.333333	A123	0.1
17	A11	0.294118	A32	A40	0.036591	A65	A75	1	A92	A101	0.333333	A123	0.1785
18	A11	0.161765	A34	A42	0.067569	A61	A75	1	A93	A101	1	A123	0.4464
19	A12	0.823529	A33	A43	0.490096	A65	A73	0.333333	A93	A101	0.333333	A124	0.1428
20	A14	0.205882	A32	A49	0.09354	A61	A74	1	A93	A101	0	A123	0.2678
21	A14	0.117647	A34	A46	0.024816	A61	A73	1	A93	A101	0.333333	A123	0.2321
22	A11	0.029412	A32	A46	0.010895	A61	A72	1	A92	A101	1	A122	0.0714
23	A14	0.294118	A32	A40	0.393034	A61	A73	0	A93	A101	1	A122	0.4285
24	A13	0.117647	A32	A42	0.110102	A61	A73	0	A92	A101	0.333333	A123	0.4821
25	A14	0.029412	A34	A40	0.100693	A63	A73	0	A94	A101	0.333333	A123	0.0892
26	A13	0.117647	A32	A43	0.17327	A65	A75	0.333333	A93	A101	0.666667	A123	0.3214

- Close the text file and return to your browser where your experiment is displayed. At the bottom left, click **NEW**. Then in the **NEW** dialog box, click the **DATASET** tab as shown in the following image.



4. Click **FROM LOCAL FILE**. Then in the **Upload a new dataset** dialog box, browse to select the **Credit-Scoring-Clean.csv** file from the folder where you extracted the lab files on your local computer. Enter the following details as shown in the image below, and then click the ✓ icon.
- **This is a new version of an existing dataset:** Unselected
  - **Enter a name for the new dataset:** Credit-Scoring-Semi-Clean
  - **Select a type for the new dataset:** Generic CSV file with a header (.csv) □ **Provide an optional description:** Bank credit scoring data.

Upload a new dataset

SELECT THE DATA TO UPLOAD:  
C:\Users\dmansor\SOUTHPACIFIC\OneDrive - Browse...

This is the new version of an existing dataset

ENTER A NAME FOR THE NEW DATASET:  
Credit-Scoring-Semi-Clean.csv

SELECT A TYPE FOR THE NEW DATASET:  
Generic CSV File with a header (.csv)

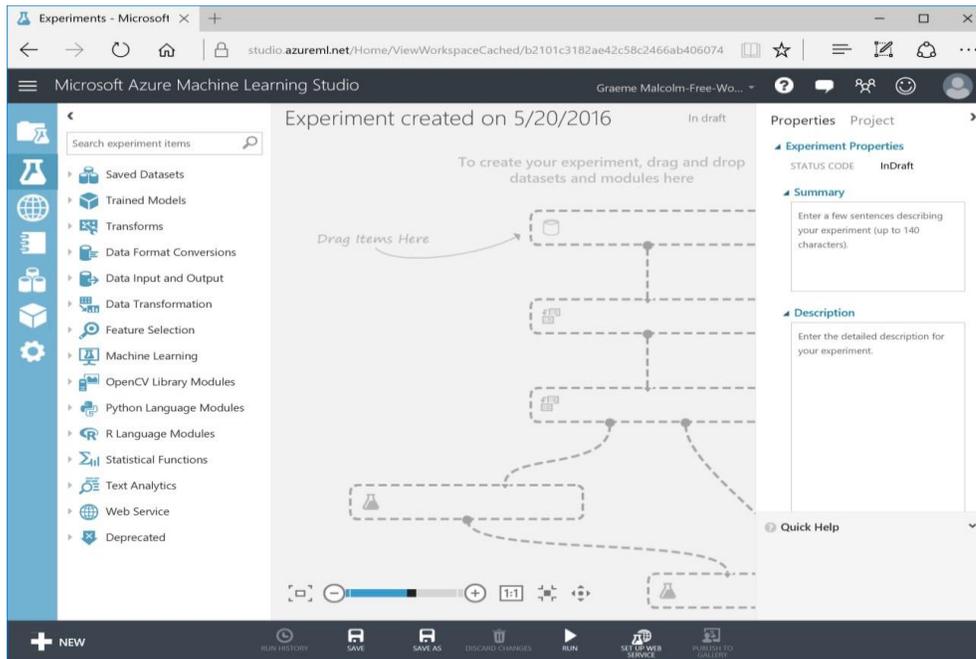
PROVIDE AN OPTIONAL DESCRIPTION:

✓

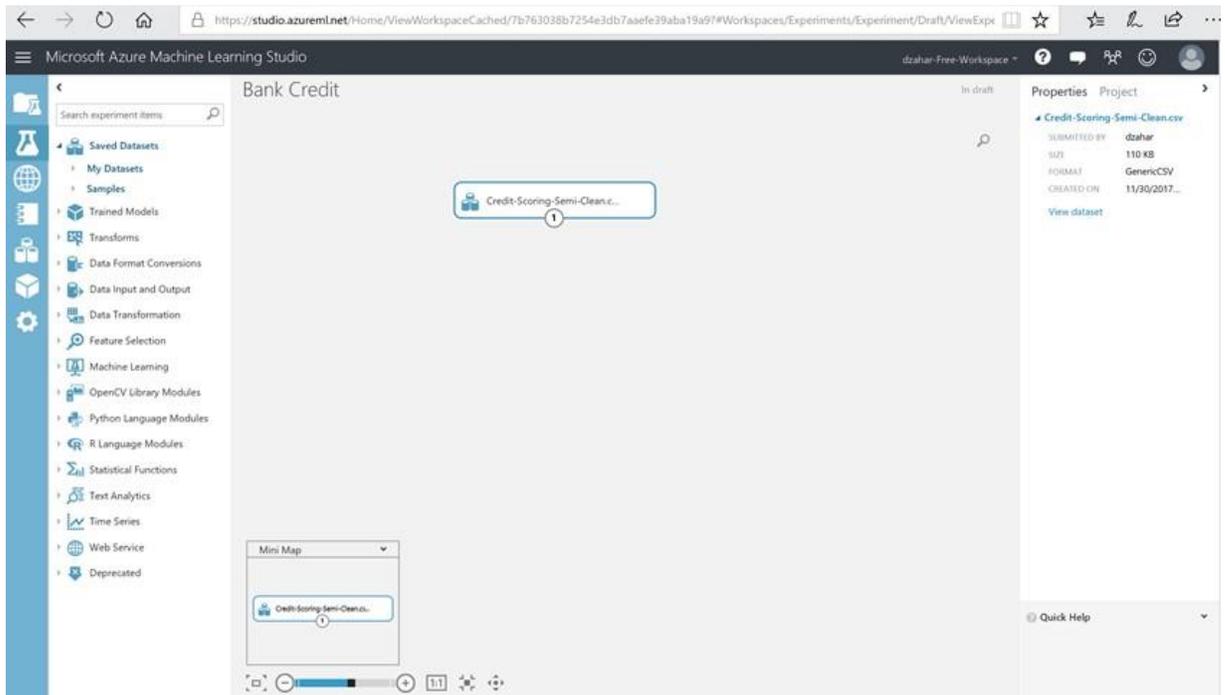
5. Wait for the upload of the dataset to complete, then click **OK** on the status bar at the bottom of the AML Studio screen.

## Create an Experiment

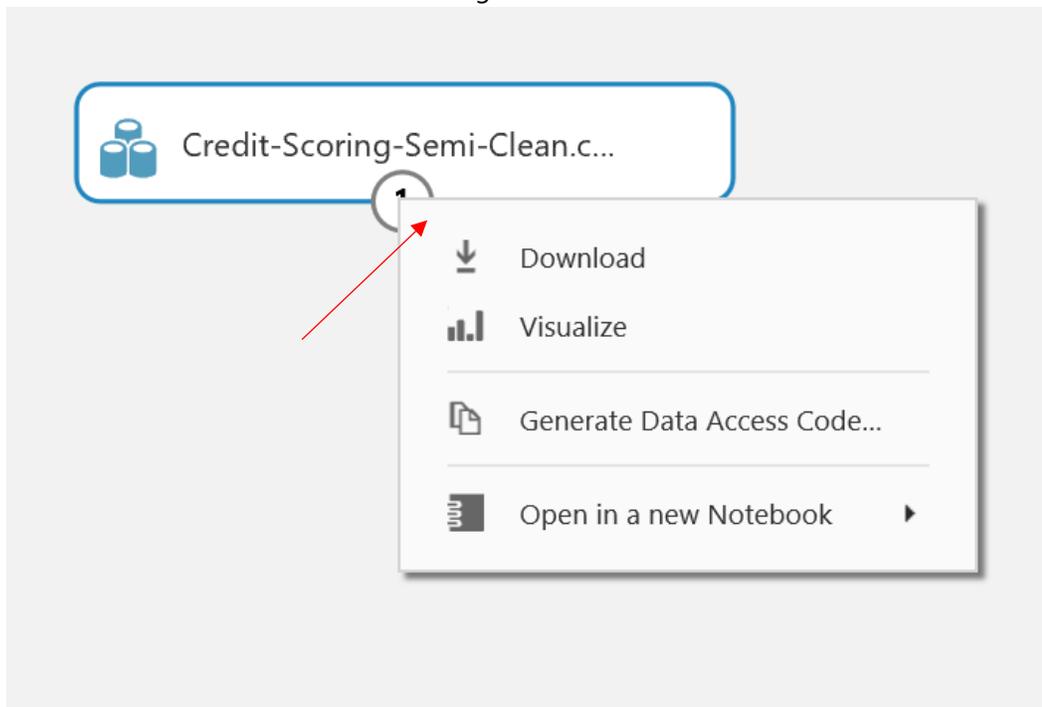
1. In the studio, at the bottom left, click **NEW**. Then in the **Experiment** category, in the collection of Microsoft samples, select **Blank Experiment**. This creates a blank experiment, which looks like the following image.



2. Change the title of your experiment from "Experiment created on *today's date*" to "**Bank Credit**".
3. On the experiment items pane, expand **Saved Datasets > My Datasets** to verify that the **Credit-Scoring-Semi-Clean** dataset is listed.
4. Drag the **Credit-Scoring-Semi-Clean** dataset to the canvas for the **Bank Credit** experiment.
5. Verify that the Azure ML screen, which shows your experiment, now looks like the figure shown here:

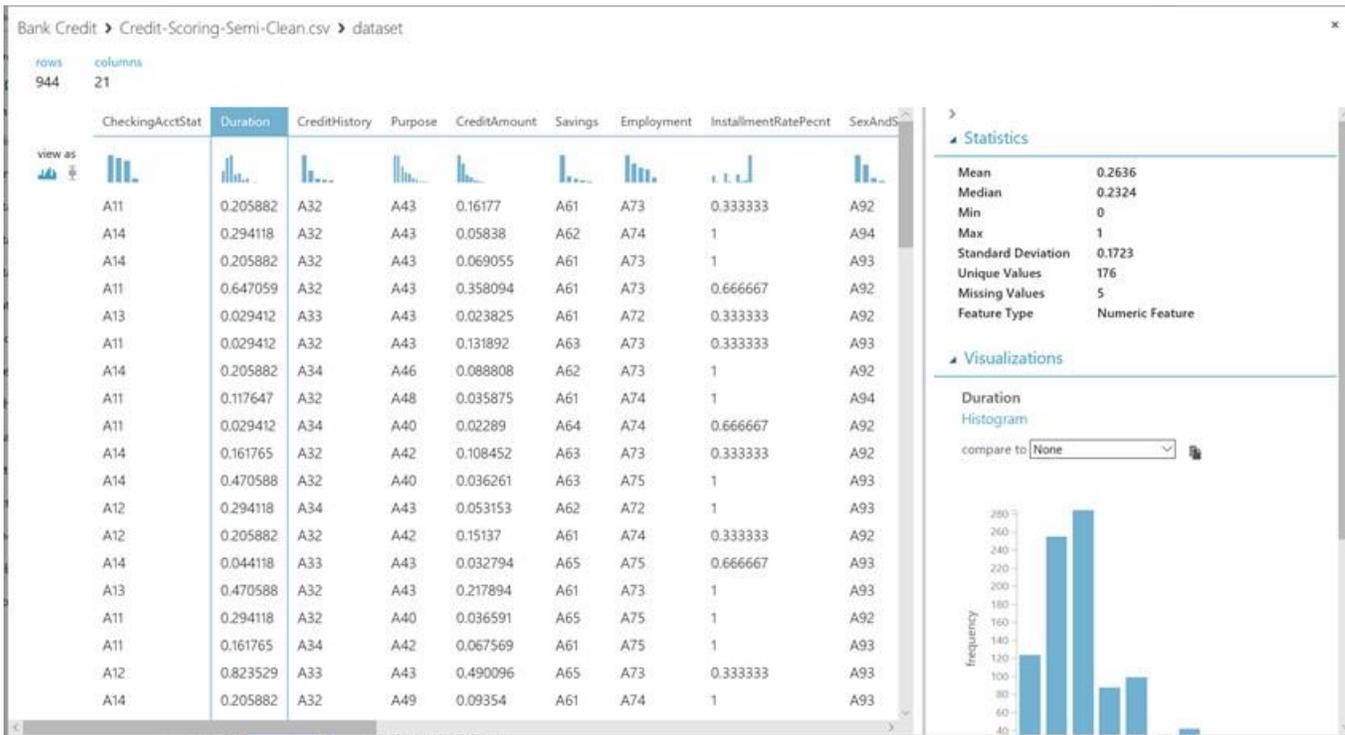


6. Click the output port for the **Credit-Scoring-Clean** dataset on the canvas and click **Visualize** to view the data in the dataset as shown in the figure:

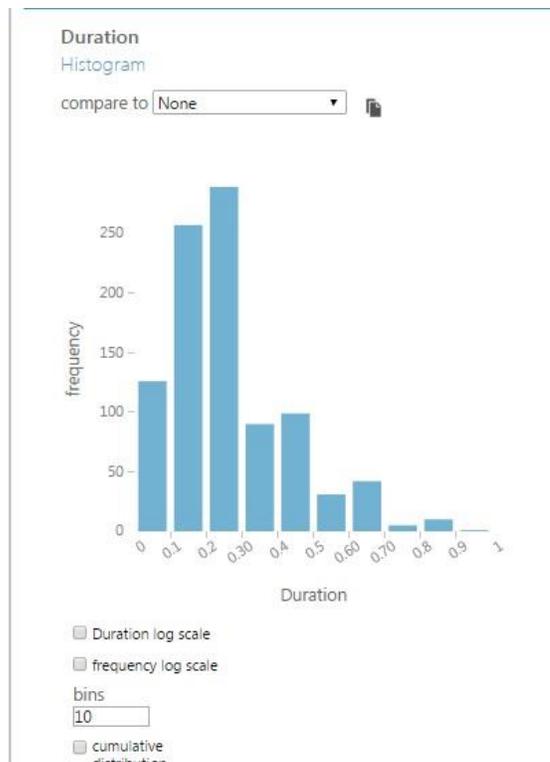


**Note:** The output port can be accessed by clicking on the small circle on the bottom of the module boxes, pointed to by the red arrow in the figure.

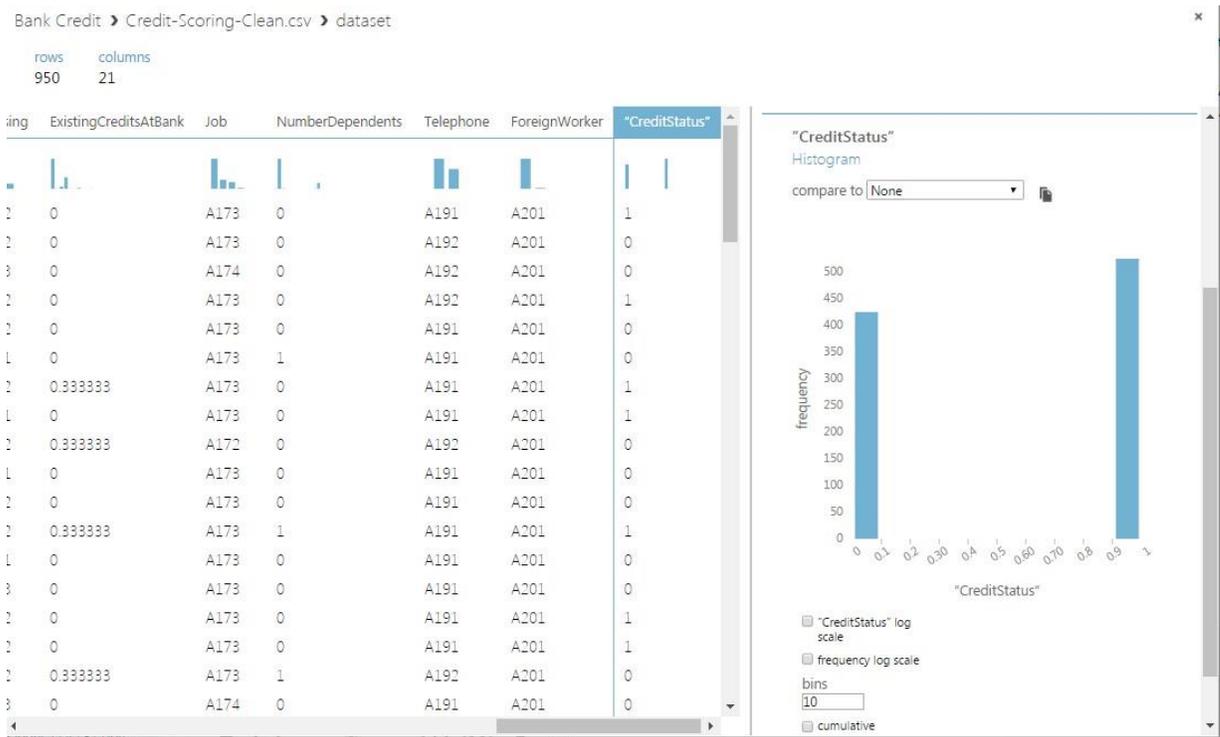
7. Click on the second column labeled **Duration**, which will display some properties of that feature (data column) on the right side of the display. These properties include summary statistics and the data type, as shown below. **Take note on the number of missing values.**



8. Verify that the dataset contains the data you viewed in the source file.
9. Using the scroll bar on the right side of the display, scroll down until you can see the histogram of the **Duration** feature as shown here:



- On the data display, scroll to the right and click **CreditStatus**. Scroll down in the pane on the right and observe the histogram, which should appear as shown below. Note that **CreditStatus** has two values, {0,1}, and that the number of cases with each value are approximately balanced.

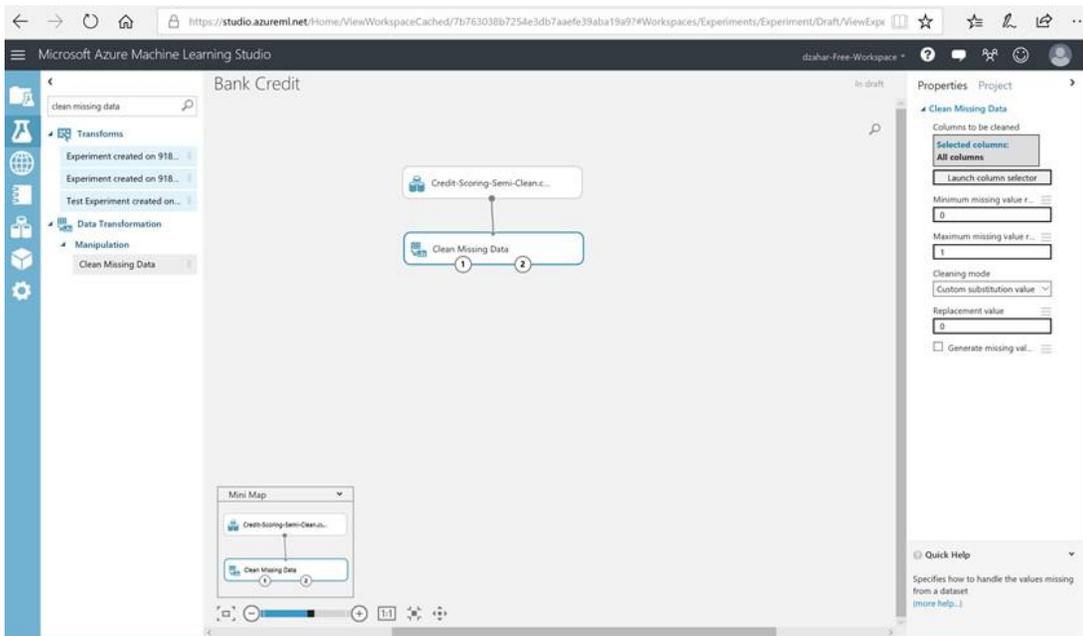


- Click the 'x' in the upper right corner to close the visualization.

## Clean Missing Data

Recall that the **Duration** column had missing data.

- In Azure ML Studio, search (search bar is near the top left hand corner of the canvas) for the **Clean Missing Data** module, which is in the **Manipulation** category under **Data Transformation**, and drag it onto the canvas.
- Connect the output of the **Credit-Scoring-Semi-Clean** dataset to the **Dataset** input of **Clean Missing Data** module as shown here:

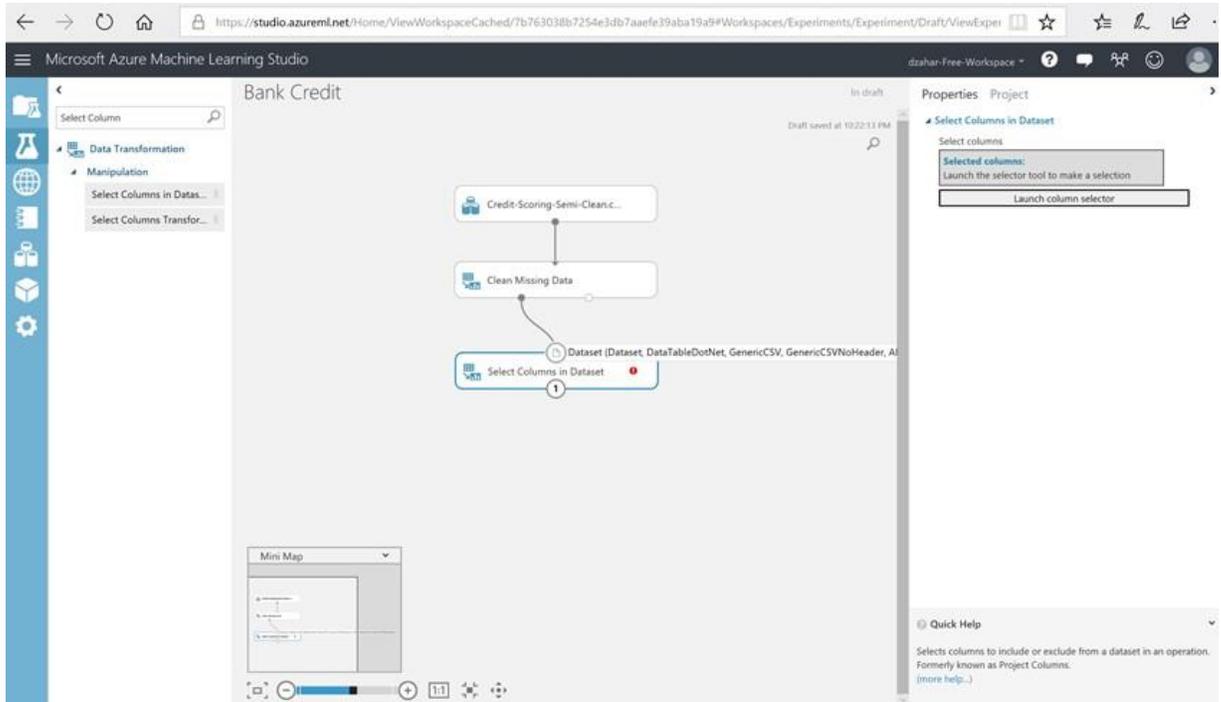


3. With the **Clean Missing Data** module selected, in the **Properties** pane, click **Launch Column** selector.
4. In the **Select columns** dialog box, note that on the **By Name** page, you can select individual columns by name; or alternatively, on the **With Rules** page you can specify rules to filter the columns. Many of the modules in Azure ML use this column selector, so you should familiarize yourself with it.
5. On the **With Rules** page, create a rule that starts with no columns and then include the **Duration** column as shown here; then click the ✓ icon to apply the filter.
6. Set the Minimum missing value ratio to be 0.
7. Set the Maximum missing value ratio to be 1.
8. Select mean for the Cleaning mode.
9. Set Cols with missing all values to Remove.
10. Leave Generate missing value indicator column unchecked.
11. **Save** and **Run** your experiment by clicking on the icons at the bottom of the studio.
12. When your experiment has finished running, visualize the **Results Dataset** output of the **Clean Missing Data** module. Check that there are no more missing values in the **Duration** column.

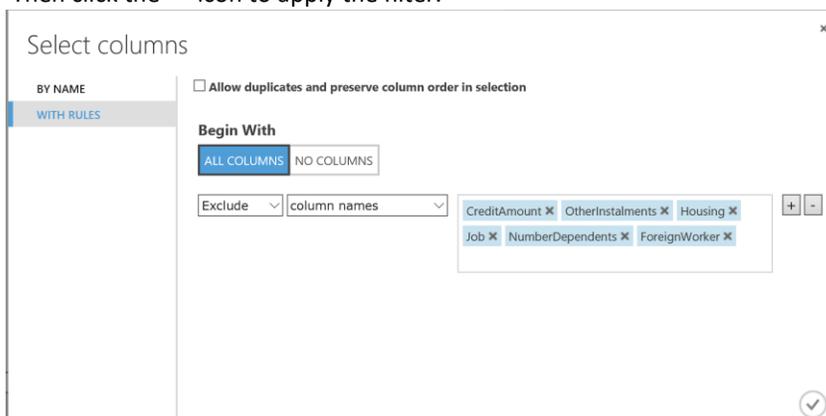
### Select Columns from the Dataset

1. In Azure ML Studio, search for the **Select Columns in Dataset** module, which is in the **Manipulation** category under **Data Transformation**, and drag it onto the canvas.

2. Connect the output of the **Clean Missing Data** Cleaned dataset output to the **Dataset** input of the **Select Columns in Dataset** module as shown here:



3. With the **Select Columns in Dataset** module selected, in the **Properties** pane, click **Launch Column** selector.
4. In the **Select columns** dialog box, note that on the **By Name** page, you can select individual columns by name; or alternatively, on the **With Rules** page you can specify rules to filter the columns.
5. On the **With Rules** page, create a rule that starts with all columns and then **exclude** the following columns: CreditAmount, OtherInstalments, Housing, Job, NumberDependents, ForeignWorkers
6. Then click the ✓ icon to apply the filter.



7. **Save** and **Run** your experiment by clicking on the icons at the bottom of the studio.

- When your experiment has finished running, visualize the **Results Dataset** output of the **Select Columns in Dataset** module. Note that there are now 15 columns, because the 6 columns have been removed.

## Using Custom Code in Azure ML (R or Python – Select ONE)

In this exercise you will use the Python, R and SQL to filter more columns from the **Credit-Scoring-Clean** data set.

### Add an Execute R Script Module

- Search for the **Execute R Script** module, which is under the **R Language Modules**, and drag it onto the canvas.

**Note:** R is a commonly used scripting language in data science experiments, and it enables you to include custom logic in an Azure ML experiment. You can learn more about using R elsewhere. For now, use a simple R script to remove some more columns from the dataset.

- Connect the **Results Dataset1** output of the **Select Columns in Dataset** module to the **Dataset1** (left most) input of the **Execute R Script** module.
- Select the **Execute R Script** module, set the **R Version** to the latest available version of **Microsoft R Open**, and then replace the existing R code in the code editor window of the **Execute R Script** module with the following code drops the **SexAndStatus** and **OtherDetorsGuarantors** columns.

```
credit.frame <- maml.mapInputPort(1)
drop.cols <- c('SexAndStatus', 'OtherDebtorsGuarantors')
out.frame <- credit.frame[, !(names(credit.frame) %in% drop.cols)]
maml.mapOutputPort("out.frame")
```

The screenshot displays the Microsoft Azure Machine Learning Studio interface. The main workspace shows a workflow titled "Bank Credit" with the following modules: "Credit-Scoring-Semi-Clean...", "Clean Missing Data", "Select Columns in Dataset", and "Execute R Script". The "Execute R Script" module is selected, and its properties are visible on the right. The R script code is as follows:

```
1 credit.frame <- maml.mapInputPort(1)
2 drop.cols <- c('SexAndStatus', 'OtherDebtorsGuarantors')
3 out.frame <- credit.frame[, !(names(credit.frame) %in% drop.cols)]
4 maml.mapOutputPort("out.frame")
5
```

The properties panel on the right shows the R Version set to "CRAN R 3.3.0". The execution details are as follows:

Property	Value
START TIME	11/30/2017 10:57:37 PM
END TIME	11/30/2017 10:57:51 PM
ELAPSED TIME	0:00:13.962
STATUS CODE	Finished
STATUS MESSAGE	None

At the bottom of the properties panel, there is a "Quick Help" section with the text: "Executes an R script from an Azure Machine Learning experiment (more help...)"

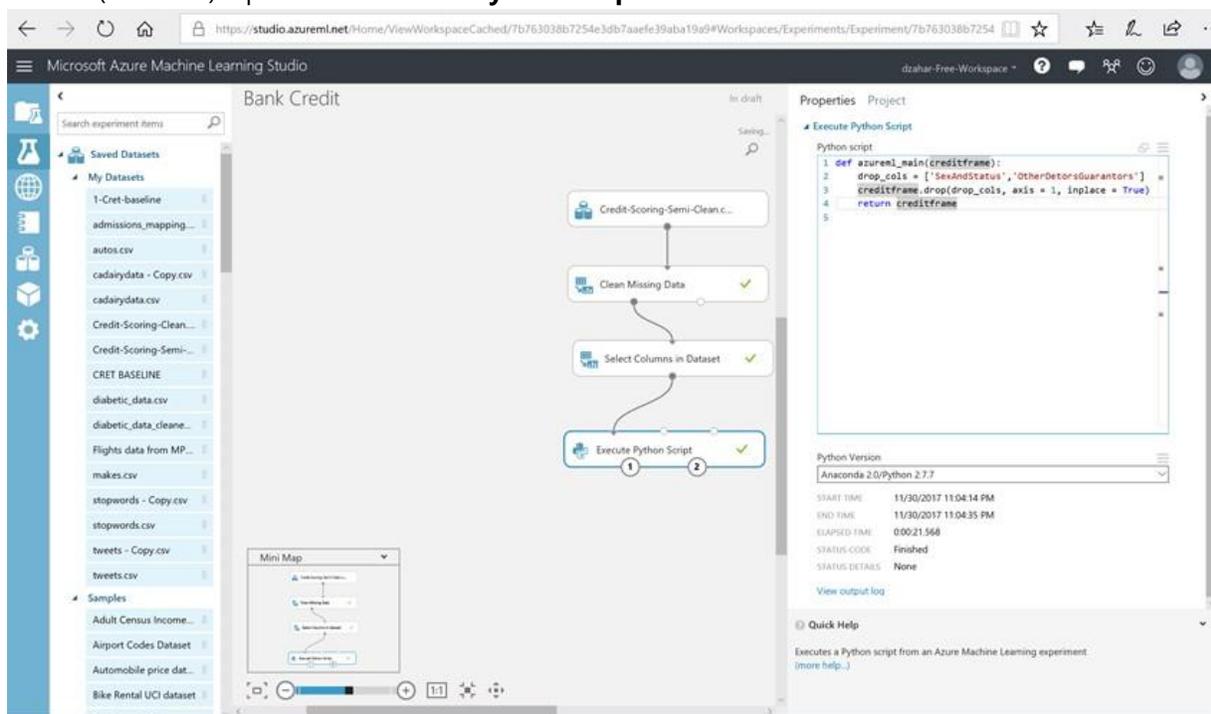
4. **Save** and **Run** the experiment. Then, when it has finished running, visualize the **Results Dataset** (left hand) output of the **Execute R Script** module. Note that there are now 18 columns.

#### Add an Execute Python Script Module

1. In Azure ML Studio, search for the **Execute Python Script** module, which is under **Python Language Modules**, and drag it onto the canvas.

**Note:** Python is a commonly used scripting language in data science experiments; and like R, it enables you to include custom logic in an Azure ML experiment. You can learn more about using Python elsewhere. For now, use a simple Python script to remove some columns from the dataset.

2. Connect the **Results Dataset** output of the **Select Columns in Dataset** module to the **Dataset1** (left most) input of the **Execute Python Script** module as shown here:



3. Select the **Execute Python Script** module, set the **Python Version** to the latest available version of Python, and then replace the existing code in the code editor pane with the following code, which drops the **SexAndStatus** and **OtherDebtorsGuarantors** columns. You can copy and paste this code from **dropcols.py** in the lab files folder for this lab.

```
def azureml_main(creditframe):  
    drop_cols = ['SexAndStatus', 'OtherDebtorsGuarantors']  
    creditframe.drop(drop_cols, axis = 1, inplace = True)  
    return creditframe
```

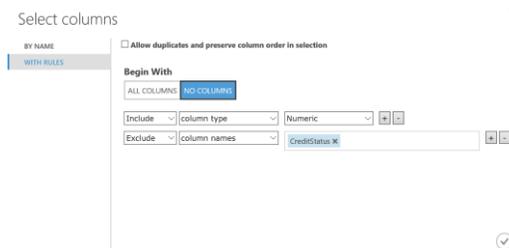
**Tip:** To paste code from the clipboard into the code editor in the Azure ML **Properties** pane, press **CTRL+A** to select the existing code, and then press **CTRL+V** to paste the code from the clipboard, to replace the existing code.

4. **Save** and **Run** the experiment, and when it has finished running, visualize the **Results Dataset** (left hand) output of the **Execute Python Script** module. Note that there are now 18 columns, as another two have been removed.

## Normalizing Numeric Data

Most datasets used in data science contain multiple numeric features, and these features are often expressed in different units of measurement. For example, age may be in years while credit amount in dollars which may have different magnitude range that makes it difficult to compare the multiple numeric features in the data set. It is always a good practice to ensure that the numeric features are normalized or scaled. Note that normalization should be applied only after the data has been cleansed (remove missing values, duplicates and outliers) and should not include the label should this be numeric.

1. Search for the **Normalize Data** module, under **Data Transformation > Scale and Reduce**, and drag it onto the canvas.
2. Connect the **Results Dataset** output of the **Execute R Script** or **Execute Python Script** module to the **Dataset** input of the **Normalize Data** module.
3. With the **Normalize Data** module selected, in the **Properties** pane:
  - a In the Transformation method drop down, select ZScore.
  - b Check Use 0 for constant values when checked.
  - c Click **Launch Column** selector, start with NO COLUMNS, then
    - i Include, Column Type, Numeric.
    - ii Click the “+” button, and then Exclude, Column Names, Credit Status



## Converting Strings to Categories

In data science, data should be numbers (numeric) or categories (names). Typically, string values are considered as categories, and should be converted into category type before being processed by the model.

1. Search for the **Edit Metadata** module, under **Data Transformation > Manipulation**, and drag it onto the canvas.
2. Connect the **Transformed Dataset** (left) output of the **Normalize Data** module to the **Dataset** input of the **Edit Metadata** module.
3. With the **Edit Metadata** module selected, in the **Properties** pane:
  - a Click **Launch Column** selector, start with NO COLUMNS.

- i Include, Column Type, String.
- b Data type drop down – leave **Unchanged** selected.
- c Categorical drop down – select **Make categorical**.
- d Fields – leave **Unchanged** selected.
- e New column names – leave blank.

4. Ensure that your model looks like this:



5. Save and run your experiment.

6. When the experiment finished running, visualize output of the Edit Metadata module and ensure that the string columns are now categorical.

## Creating and Evaluating a Machine Learning Model.

Now that you have created a simple experiment that processes data, you can use the data to train a predictive model. In this exercise, you will use the data to create a model that tries to predict if a particular bank customer is a good or bad credit risk.

**Note:** The purpose of these exercises is to give you a feel for working with machine learning models in Azure Machine Learning. In subsequent chapters we will explore the evaluation for machine learning models.

### *Split the Data*

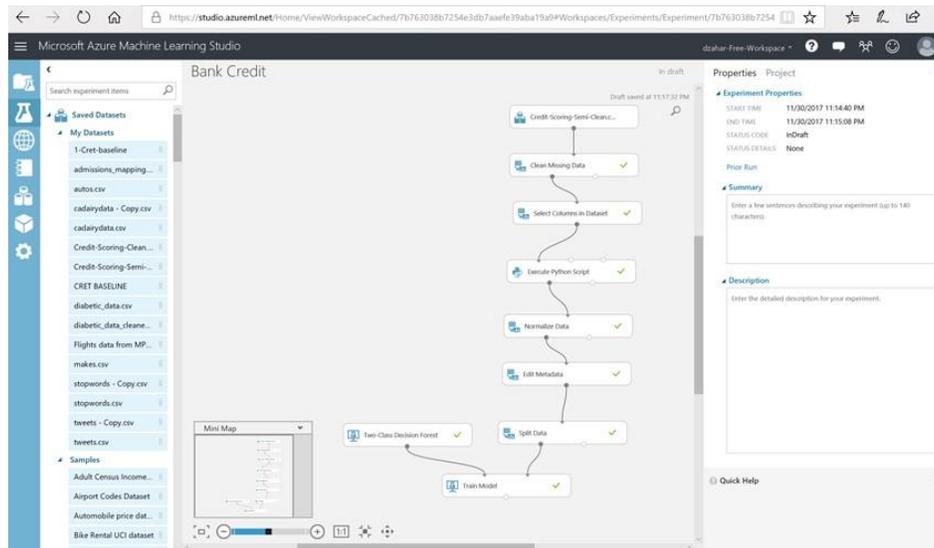
1. Search for the **Split Data** module and drag it onto the canvas under the existing modules.

**Note:** The data are split to create independent, non-overlapping, randomly sampled subsets of the data to train and evaluate the performance of the machine learning model.

2. Connect the output of the **Edit Metadata** module to the input of the **Split Data** module.
3. Select the **Split Data** module, and in the **Properties** pane, view the default split settings, which split the data randomly into two datasets. Set these properties as follows:
  - **Splitting mode:** Split Rows
  - **Fraction of rows in the first output dataset:** 0.7
  - **Randomized split:** checked
  - **Random seed:** 876
  - **Stratified split:** False

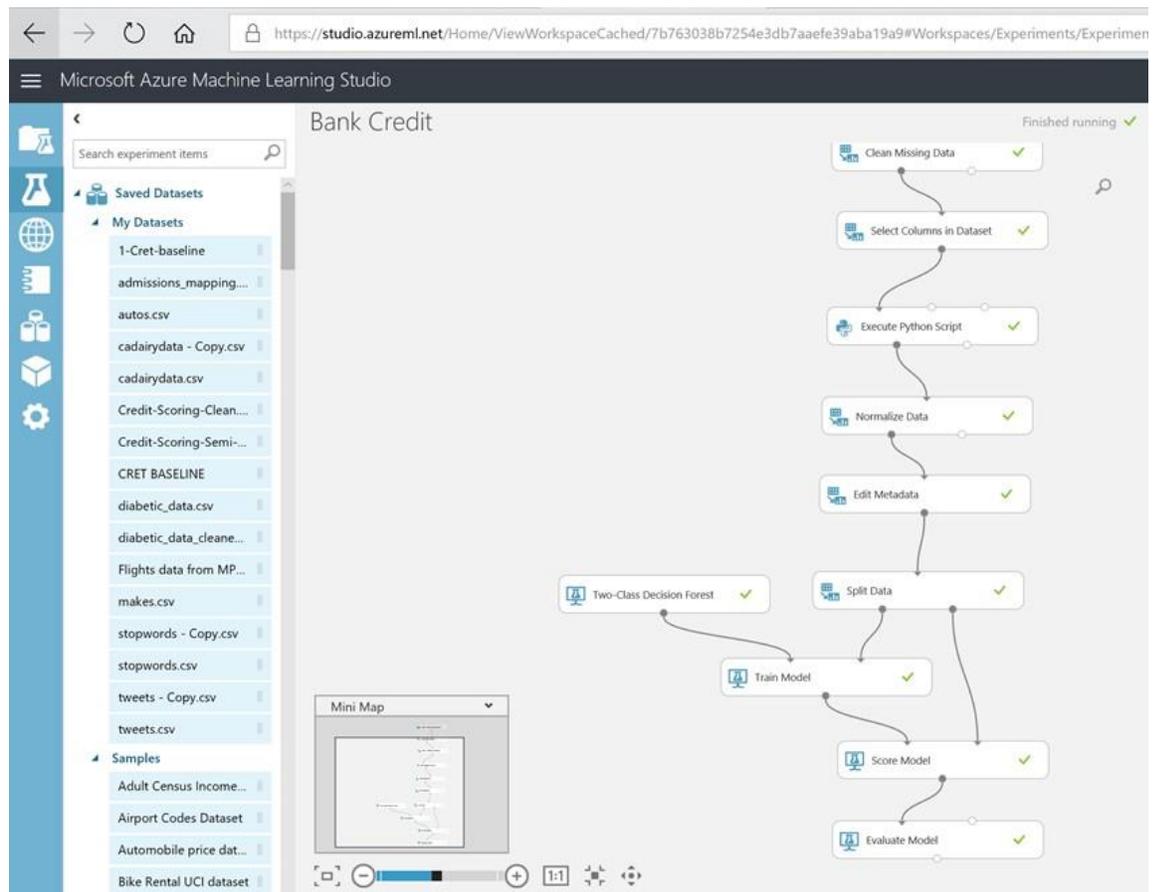
#### *Add an Algorithm and a Train Model Module*

1. In the experiment items pane, search for "Classification", and view the range of multi-class and two-class models that are supported in Azure ML. Find the **Two Class Decision Forest** module, which is under **Machine Learning > Initialize Model > Classification**, and drag it onto the canvas to the left of the **Split Data** module.
2. Select the **Two Class Decision Forest** module, and in the **Properties** pane configure the model parameters as follows:
  - **Resampling method:** Bagging
  - **Create trainer mode:** Single Parameter
  - **Number of Decision trees:** 50
  - **Maximum depth of the decision tree:** 32
  - **Number of random splits per node:** 32
  - **Minimum number of samples per leaf node:** 4
  - **Allow unknown values for categorical features:** checked
3. Search for the **Train Model** module, which is under **Machine Learning > Train**, and drag it onto the canvas beneath the existing modules.
4. Connect the output from the **Two-Class Decision Forest** module to the **Untrained model** (left) input of the **Train Model** module, and connect the **Results dataset1** (left-most) output port of the **Split Data** module to the **Dataset** (right) input of the **Train Model** module.
5. On the properties pane for the **Train Model** module, use the column selector to include only the **CreditStatus** column as the label for the model.
6. Verify that you experiment now looks like this:



### *Add Modules to Score and Evaluate the Trained Model*

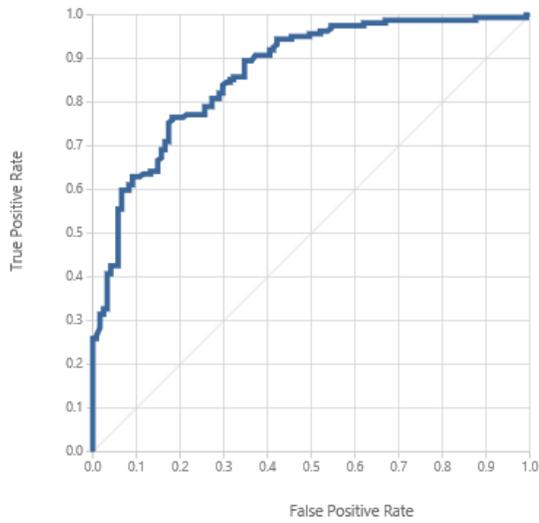
1. Search for the **Score Model** module, which is under **Machine Learning > Score**, and drag it onto the canvas under the existing modules.
2. Connect the output of the **Train Model** module to the **Trained model** (left) input of the **Score Model** module.
3. Connect the **Results Dataset2** (right) output of the **Split Data** module to the **Dataset** (right) input of the **Score Model** module.
4. Search for the **Evaluate Model** module, which is under **Machine Learning > Evaluate**, and drag it onto the canvas under the existing modules.
5. Connect the output of the **Score Model** module to the **Scored dataset** (left) input of the **Evaluate Model** module.
6. Verify that your experiment now looks like this:



### *Train and Evaluate the Model*

1. **Save** and **Run** your experiment.
2. When the experiment has finished running, visualize the output of the **Score Model** module. Note the values of the **CreditStatus** column (the known label in the test dataset) and **Scored Labels** column (the prediction computed by the model). In most cases, the values in these columns are identical, indicating that the model has correctly predicted the label value. Cases where the value of the label and the prediction differ are errors.
3. Visualize the output of the **Evaluate Model** module. Scroll down until you see performance metrics, including values for **True Positive**, **False Negative**, **False Positive**, **True Negative**, **Accuracy**, **Precision**, **Recall**, **F1 Score**, and **AUC**. These metrics are used to measure the effectiveness of the model, and will be discussed later in this course.

ROC PRECISION/RECALL LIFT



Scored dataset



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
<b>139</b>	<b>23</b>	<b>0.774</b>	<b>0.772</b>	<b>0.5</b>	<b>0.867</b>
False Positive	True Negative	Recall	F1 Score		
<b>41</b>	<b>80</b>	<b>0.858</b>	<b>0.813</b>		
Positive Label	Negative Label				
<b>1</b>	<b>0</b>				

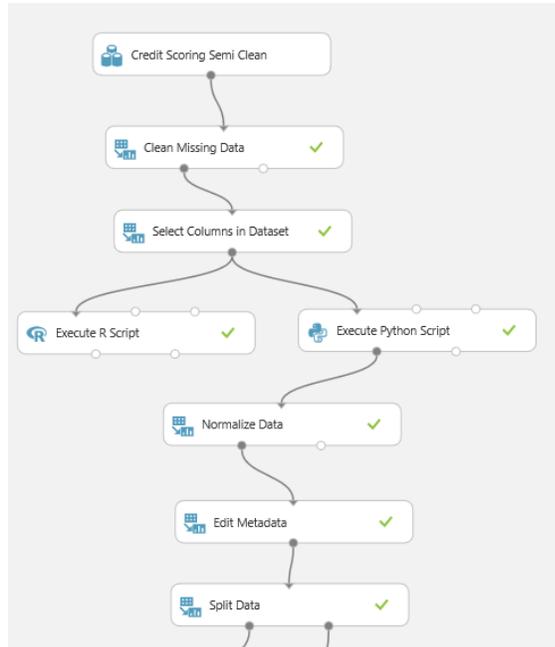
	Positive	Negative
TRUE	139	23
FALSE	41	80

Accuracy	Precision
0.774	0.772
$(TP+TN)/(TP+TN+FP+FN)$	$TP/(TP+FP)$

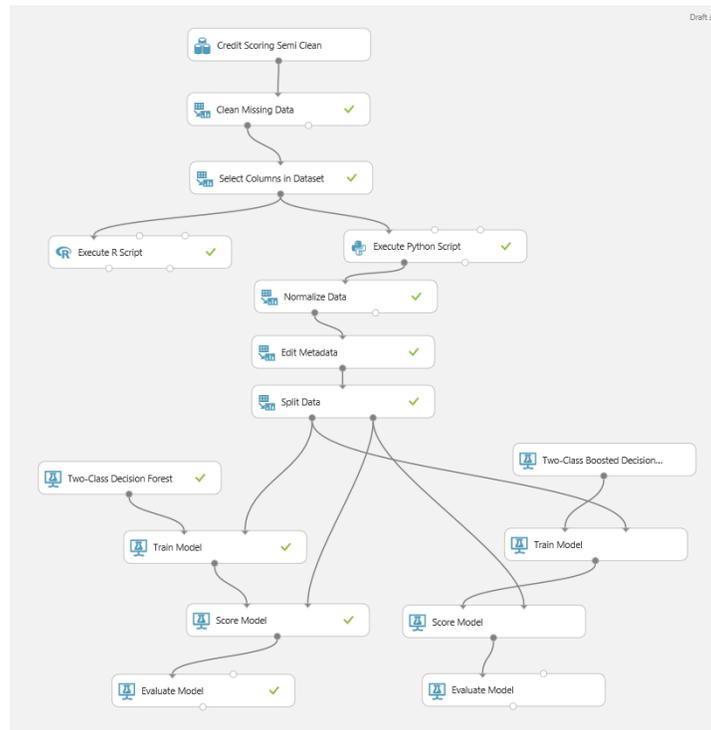
Recall	F1
0.858	0.813
$TP/(TP+TN)$	$2*(precision*recall)/(precision+recall)$

## Selecting Alternative Models

As an option, try different classification models and compare the different results. You can reuse most of the existing experiment right up to the Split Data module as shown below.



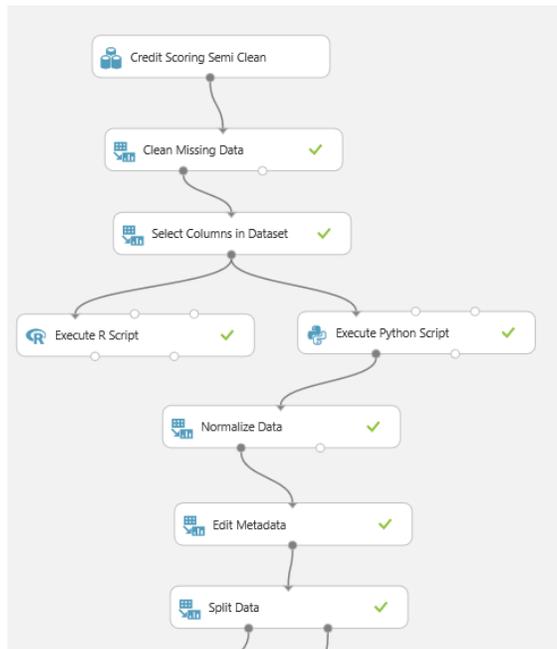
The subsequent portions of the experiment need to be arranged such that there will be a dedicated set of Train Model, Score Model and Evaluate modules for each model we would like to experiment with. The experiment shown below illustrates this where we added a Two-Class Boosted Decision Tree to be evaluated alongside the original Two-Class Decision Forest.



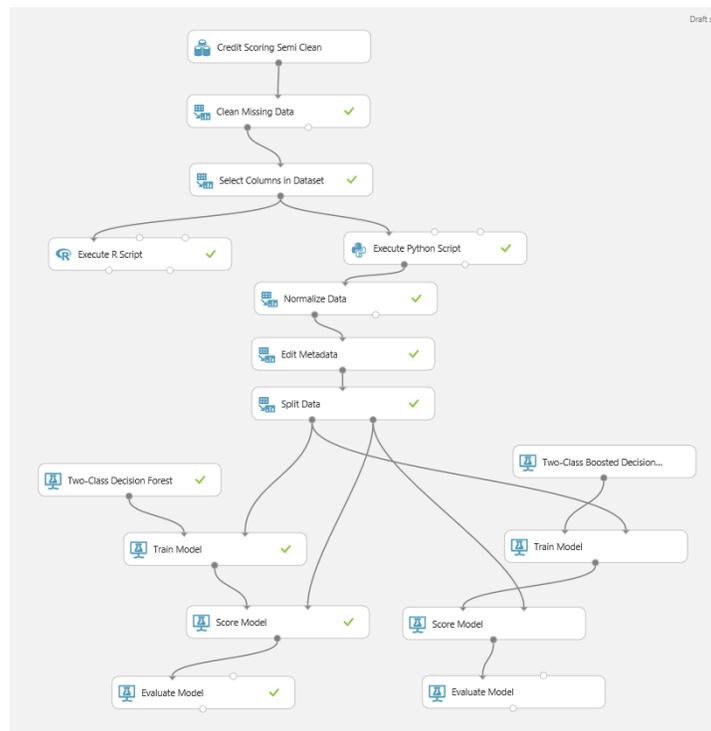
**Tip:** To reduce the effort of doing this is to use the **right mouse button** and then drag the mouse to select the existing Train Model, Score Model and Evaluate modules, press **Control A** to copy these, and then press **Control C** to paste this set of modules on the canvas. The canvas works very much like power pint when it comes to editing.

## Selecting Alternative Models

As an option, try different classification models and compare the different results. You can reuse most of the existing experiment right up to the Split Data module as shown below.



The subsequent portions of the experiment need to be arranged such that there will be a dedicated set of Train Model, Score Model and Evaluate modules for each model we would like to experiment with. The experiment shown below illustrates this where we added a Two-Class Boosted Decision Tree to be evaluated alongside the original Two-Class Decision Forest.



**Tip:** To reduce the effort of doing this is to use the **right mouse button** and then drag the mouse to select the existing Train Model, Score Model and Evaluate modules, press **Control A** to copy these, and then press **Control C** to paste this set of modules on the canvas. The canvas works very much like power point when it comes to editing.

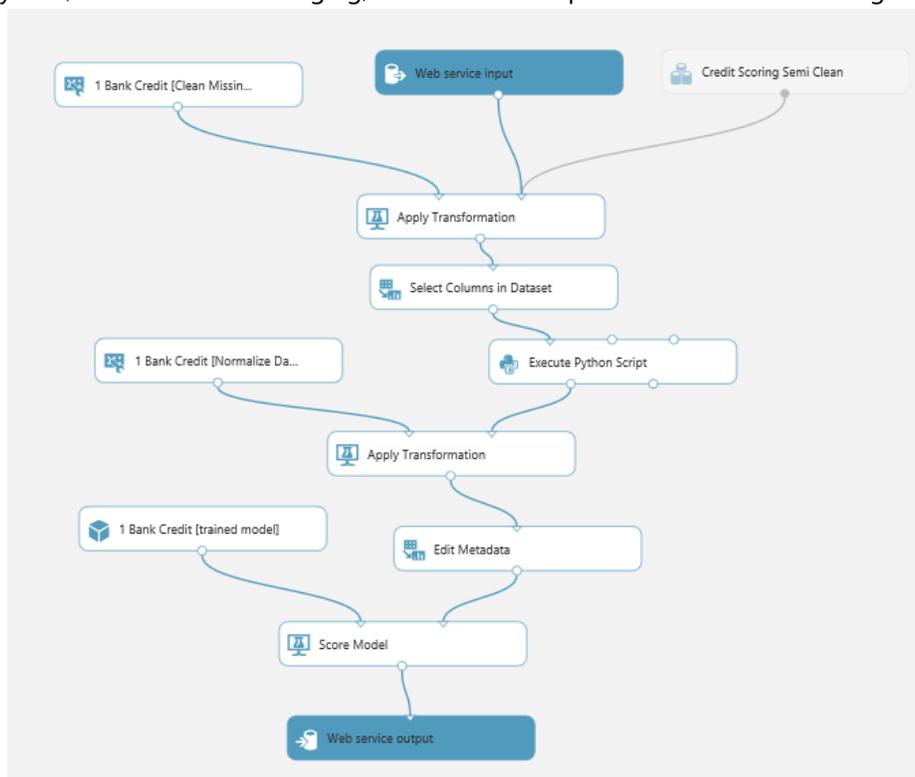
## Model Deployment.

In this lab, you will use Azure Machine Learning to train, evaluate, and publish a classification model, a regression model, and a clustering model. The point of this lab is to introduce you to the basics of creating machine learning models in Azure ML, it is not intended to be a deep-dive into model design, validation and improvement.

**Note:** This lab builds on knowledge and skills developed in previous labs described in the previous sections. Also if the option to test multiple models were done, select the best model and remove the other (also by selecting the unwanted modules, and then deleting them).

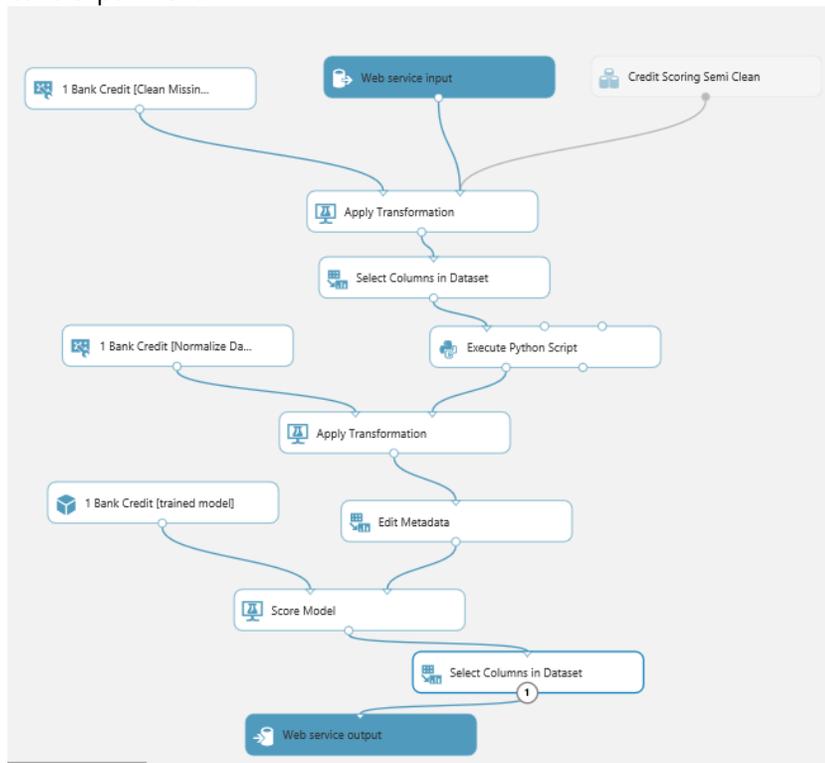
### Publish the Model as a Web Service

1. With the **previously built <Bank Credit>** experiment open, click the **SET UP WEB SERVICE** icon at the bottom of the Azure ML Studio page and click **Predictive Web Service [Recommended]**. A new **Predictive Experiment** tab will be automatically created.
2. Verify that, with a bit of rearranging, the Predictive Experiment resembles this figure:



3. Delete the connection between the **Score Model** module and the **Web service output** module.
4. Add a **Select Columns in Dataset** module to the experiment and connect the **Score Model** module output to its input. Then connect the output of the **Select Columns in Dataset** module to the input of the **Web service output** module.

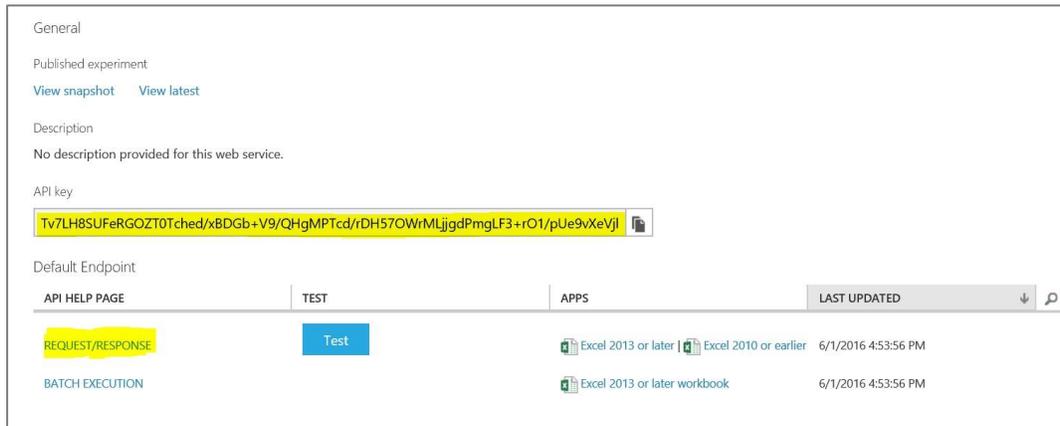
5. Select the **Select Columns in Dataset** module and use the column selector to select only the **Scored Labels** column. This ensures that when the web service is called, only the predicted value is returned.
6. Ensure that the predictive experiment now resembles like the following, and then save and run the predictive experiment:



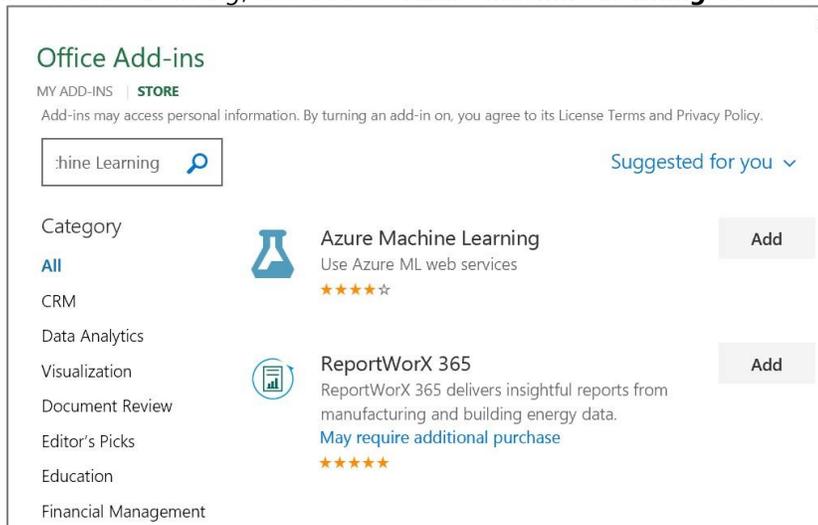
7. When the experiment has finished running, visualize the output of the last **Select Columns in Dataset** module and verify that only the **Scored Labels** column is returned.

### Deploy and Use the Web Service

1. In the **Previously built experiment <Bank Credit> [Predictive Exp.]** experiment, click the **Deploy Web Service** icon at the bottom of the Azure ML Studio window.
- 2.
3. Wait a few seconds for the dashboard page to appear and note the **API key** and **Request/Response** link. You will use these to connect to the web service from a client application.



4. Leave the dashboard page open in your web browser, and open a new browser tab.
5. In the new browser tab, navigate to <https://office.live.com/start/Excel.aspx>. If prompted, sign in with your Microsoft account (use the same credentials you use to access Azure ML Studio.)
6. In Excel Online, create a new blank workbook. |
7. On the **Insert** tab, click **Office Add-ins**. Then in the **Office Add-ins** dialog box, select **Store**, search for *Azure Machine Learning*, and add the **Azure Machine Learning** add-in as shown below:



8. After the add-in is installed, in the **Azure Machine Learning** pane on the right of the Excel workbook, click **Add Web Service**. Boxes for the URL and API key of the web service will appear.
9. On the browser tab containing the dashboard page for your Azure ML web service, right-click the **Request/Response** link you noted earlier and copy the web service URL to the clipboard. Then return to the browser tab containing the Excel Online workbook and paste the URL into the URL box.
10. On the browser tab containing the dashboard page for your Azure ML web service, click the **Copy** button for the **API key** you noted earlier to copy the key to the clipboard. Then return to the browser tab containing the Excel Online workbook and paste it into the **API key** box.

- Verify that the **Azure Machine Learning** pane in your workbook now resembles this, and click **Add**.

### Azure Machine Learning

Web Services

- + Titanic Survivor Predictor (Excel Add-in Sa...
- + Text Sentiment Analysis (Excel Add-in Sam...

URL ?

```
https://studio.azureml.net/apihelp/workspaces/b2101c3182ae42c58c2466ab40607479/webservices/4d98abd657a449a895374b16fa2722aa/endpoints/66ce37974550469ba9b9c3d90eb25814/score
```

API key ?

```
Tv7LH8SUFerGOZT0Tched/xBDGb+V9/QHgMPJcc/rDH57OWrMljgdPmgLF3+rO1/pUe9vXevJlU7dHjU8TPyg==
```

Cancel
Add

Auto-predict Predict All

- After the web service has been added, in the **Azure Machine Learning** pane, click **1. View Schema** and note the *inputs* expected by the web service (which consist of the fields in the original Adult Census dataset) and the *outputs* returned by the web service (the **Scored Labels** field).
- In the Excel worksheet select cell A1. Then in the **Azure Machine Learning** pane, collapse the **1. View Schema** section and in the **2. Predict** section, click **Use sample data**. this enters some sample input values in the worksheet.
- Modify the sample data in row 2 onwards (or copy and paste the contents of **Credit-Scoring-Semi-Clean-Test.csv** into the online spreadsheet as shown below).

The screenshot shows the Excel Online interface with a spreadsheet containing data from the 'CheckingAccount' dataset. The data includes columns for checking duration, credit limit, purpose, credit history, savings, employment, treatment, sex, education, other debts, property, age, other debts housing, existing credit, number of telephone, foreign, and credit status. The Azure Machine Learning pane on the right is open to the 'Predict' section, where 'Use sample data' is selected, and the 'Input output' section is visible.

CheckingDuration	CreditLimit	Purpose	CreditHistory	Savings	Employment	Treatment	Sex	Education	Other Debts	Property	Age	Other Debts Housing	Existing Credit	Number Of Telephone	Foreign	Credit Status
A12	0.34718	A85	0.05855	A61	A73	1.482	A021	0.33333	A122	A122	0.18512	A145	A122	0.4173	0.4139	A201
A12	0.34171	A82	0.15487	A61	A73	1.484	A021	0.33333	A122	A122	0.39193	A145	A122	0.4173	0.4139	A201
A11	0.17026	A83	0.13409	A61	A73	1.483	A021	0.17046	A122	A122	0.17026	A145	A122	0.24701	A178	0.4139
A11	0.17369	A84	0.12347	A61	A73	1.482	A021	0.09348	A124	A124	0.22083	A145	A122	0.4173	0.4139	A201
A11	0.08239	A84	0.04761	A62	A72	1.482	A021	0.44762	A122	A122	0.08851	A145	A122	0.4173	0.4139	A201
A11	0.06431	A82	0.01406	A62	A71	1.482	A021	0.69716	A122	A122	0.05478	A145	A122	0.4173	0.4139	A201

15. Select the cells containing the input data (e.g cells A1 to U7 if the contents of the file Credit-Scoring-Semi-Clean-Test.csv was used ), and in the **Azure Machine Learning** pane, click the button to select the input range and confirm that it is something like < 'Sheet1'!A1:U7' >
16. Ensure that the **My data has headers** box is checked if there are headers in the first row.
17. In the **Output** box type <V1>. Check the **Include header** check box.
18. Click the **Predict** button, and after a few seconds, view the predicted label in cell P2.
19. Try changing a few of the values in the feature columns and click **Predict** again. Then review the updated label values provided by the web service.

## Summary

This lab has familiarized you with the essentials of using the Azure ML Studio environment. In this lab you have used built-in Azure ML functionality, Python and/or R select the features used for training a machine learning model. You then created, trained, and evaluated a first machine learning model to classify bank customers as good or bad credit risks.

As an option, try different classification models and compare the different results.