

Learn how MicroStrategy developers incorporated Azure Machine Learning into USHER

By Beth Zeranski, Microsoft - what’s norm for the show me then Kevin or Kevin then me or delete this?

By Kevin Do, MicroStrategy

Workshop storyline

New

MicroStrategy leveraged Azure Machine Learning to automatically make effective predictions in their application. In this workshop, we will use flight data to demonstrate how to make predictions using Azure Machine Learning, ML, and to highlight how to think about machine learning that is available today.

Haven’t you always wondered about where machine learning could actually be used? In this tutorial, you will learn the magic of Azure Machine Learning that enables MicroStrategy developers to add significant functionality in very little time by consuming Azure Machine Learning as a web service and quickly developing it into dashboards in MicroStrategy Workstation.

In under one hour, you will gain an understanding of how anyone can use publicly available data to make predictions and the thinking behind this effort. When complete, you will have a deeper understanding of Machine Learning and gain insight into additional possibilities.

Overview

The main sections of this workshop are:

1. Prerequisites
   1. Azure Machine Learning only requires network connectivity to the internet, no sign-up, no account required
   2. Download files from GitHub to your C: Drive
   3. MicroStrategy Desktop – Download and Install MicroStrategy Desktop, include R installation
2. Step-by-step Azure Machine Learning for Flight Prediction - how to use Machine Learning, ML, for predicting if a flight will be late
3. Step-by-step MicroStrategy Workstation – Integrate MicroStrategy with Azure ML to make predictions and easily create effective dashboards
4. **Pre-requisites:**

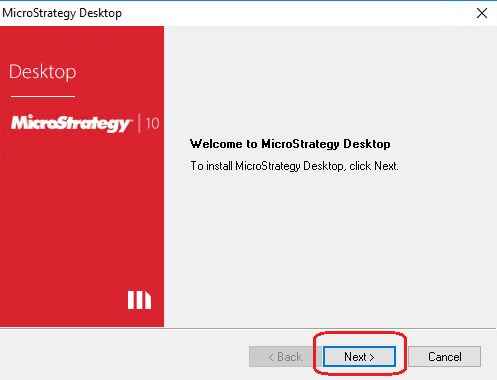
**Download files from GitHub to your C: Drive**

Download the excel and R files to your C drive from:

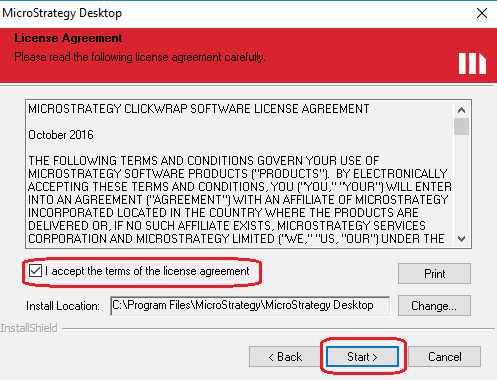
<https://github.com/bethz/AzureML-FlightPrediction/tree/master/MicroStrategy>

Download and Install MicroStrategy Desktop

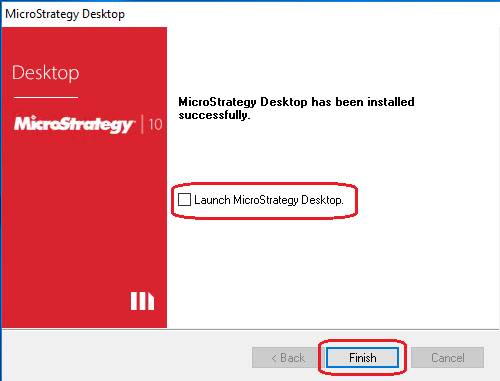
1. If you do not have MicroStrategy Desktop installed on your laptop yet, you can download it from the below link for free: <https://www.microstrategy.com/us/desktop>
2. Go to your Downloads folder to find the zip file you just downloaded, unzip it and run the “MicroStrategyDesktop-64bit.ext” to start the installation.
3. **Click on Next button on the installation wizard.**

****

1. **Accept the terms of the license agreement, and click Star button.**

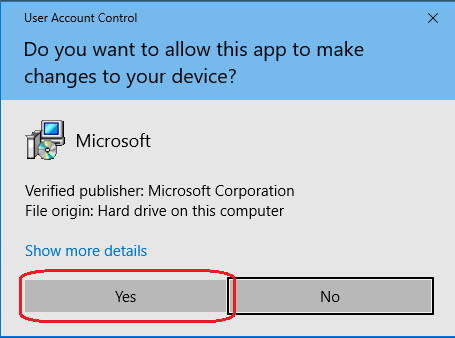
****

1. **Uncheck “Launch MicroStrategy Desktop” and click Finish.**

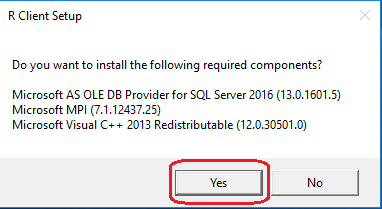
****

Download and installing R on Windows.

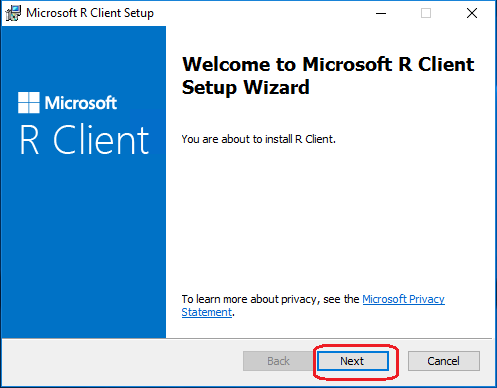
1. Download Microsoft R Client by click the following link: <http://aka.ms/rclient/download>
2. In the Downloads folder, double click on “RClientSetup.exe” to install Microsoft R Client.
3. On the pop-up below, click on Yes.

****

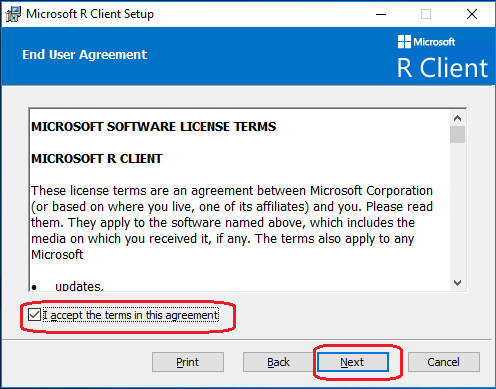
1. Click on Yes on following pop up window.



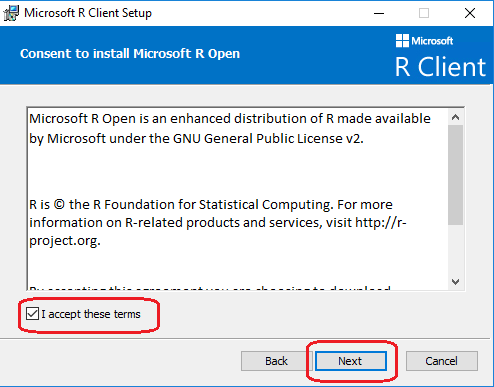
1. Click Next on the setup wizard.



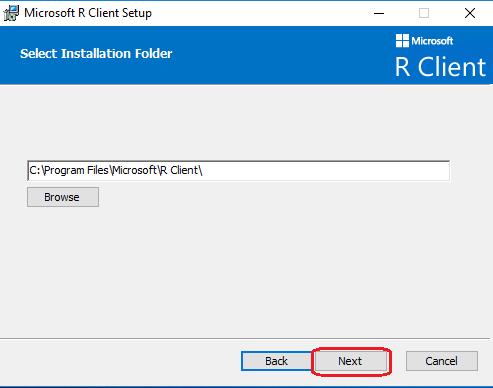
1. Accept the terms, and click Next.



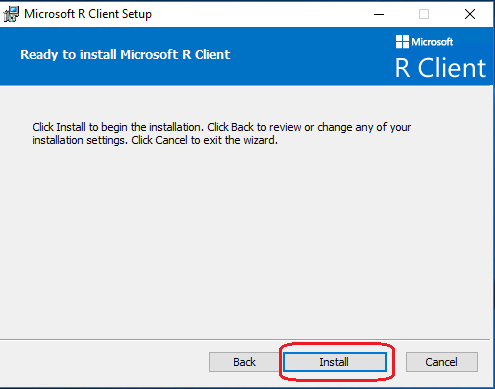
1. Accept the R Open terms and click Next.



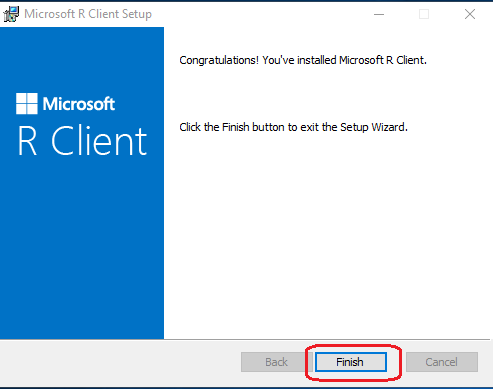
1. Keep the default installation path, and click next.



1. Click on Install.



1. Click Finish

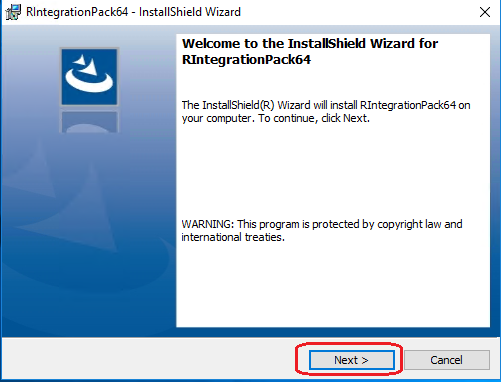


Downloading and installing R Integration Pack

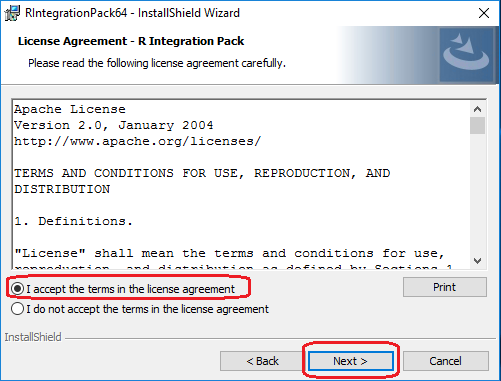
1. Go to the link: <http://rintegrationpack.codeplex.com/releases/view/630028>
2. On the web page, locate and click on the version as shown below.



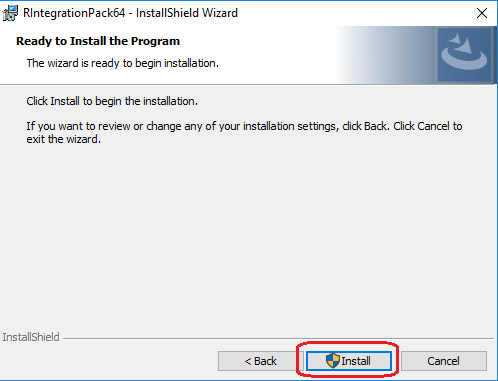
1. In Downloads folder, double click on the file “RIntegrationPack64.msi” to start installing R Integration Pack.
2. Click Next.



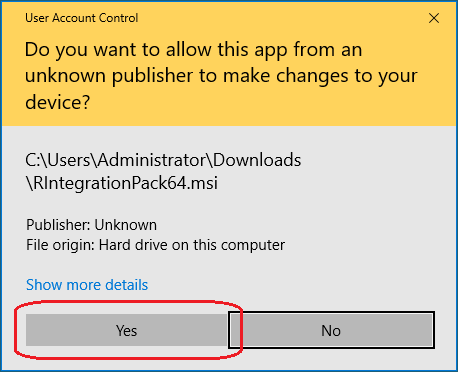
1. Accept the terms in the license agreement, and click Next.



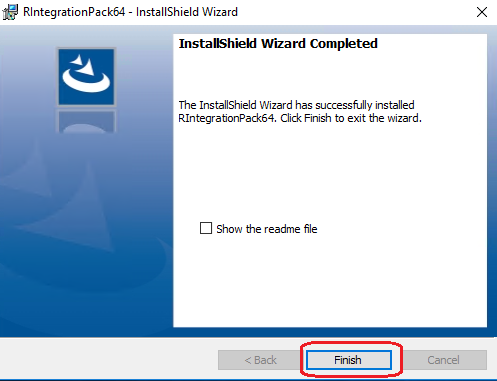
1. Click next until you see the screen below.
2. Click on Install.



1. On the User Access Control (UAC) pop-up, click Yes.



1. Click on Finish to complete the installation.



1. **Step-by-step Azure Machine Learning for Flight Prediction**

Next, we will open two items on your screen, one on the left and one on the right. On the left, open this:

<https://github.com/bethz/AzureML-FlightPrediction>/MicroStrategy/2017MSTRWorldAMLWorkshop.docx

You can also save the file for future use. A paper manual will be provided at the conference.

On the right of your screen, open a browser.

A simple example in Azure Machine Learning will be used to demonstrate concepts that can be leveraged for others uses. The example uses some publicly available data to use machine learning to predict when a flight will be late. We begin by accessing Azure ML.

# Azure Machine Learning for Flight Prediction

This workshop uses publicly available Flight Data to predict when a flight will be late. We begin by accessing Azure ML. Next steps:

[2.1. Access Azure ML](https://github.com/bethz/AzureML-FlightPrediction#1-access-azure-ml)  
[2.2 Create an Experiment](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#10-run-experiment)  
[2.3 Import, Review and Clean Data](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#3-import-review-and-clean-data)  
[2.4 Specify Columns to Use](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#4-specify-columns-to-use)  
[2.5 Split The Data Into A Training And Test Set](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#5-split-the-data-into-a-training-and-test-set)  
[2.6 Train the model](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#6-train-the-model)  
[2.7 Select Algorithm](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#7-select-algorithm)  
[2.8 Score the Model](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#8-score-the-model)  
[2.9 Evaluate Model](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#9-evaluate-model)  
[2.10 Run Experiment!](https://github.com/bethz/AzureML-FlightPrediction/blob/master/README.md#10-run-experiment)

## 2.1 Access Azure ML

This Azure Machine Learning Challenge can be completed with:

\* [Guest Access](http://studio.azureml.net/home/anonymous) - <http://studio.azureml.net/home/anonymous>

This does not require an Azure subscription or a credit card. It is anonymous access. After 8 hours the workspace gets reset. This is a great option for evaluation and this workshop.

\* [Your Own Account](https://studio.azureml.net/Home) - <https://studio.azureml.net/Home>

Sign in and use a work, school or Microsoft account. Please do not take a tour now, explore the tour later. There are two types:

\* Free Account ([Microsoft Account](https://signup.live.com/signup) required) - <https://signup.live.com/signup>

10 GB of Storage, R and Python Scripts and Web Service access

\* Enterprise Grade ([Azure Subscription](https://azure.microsoft.com/en-us/free/) required) - <https://azure.microsoft.com/en-us/free/>

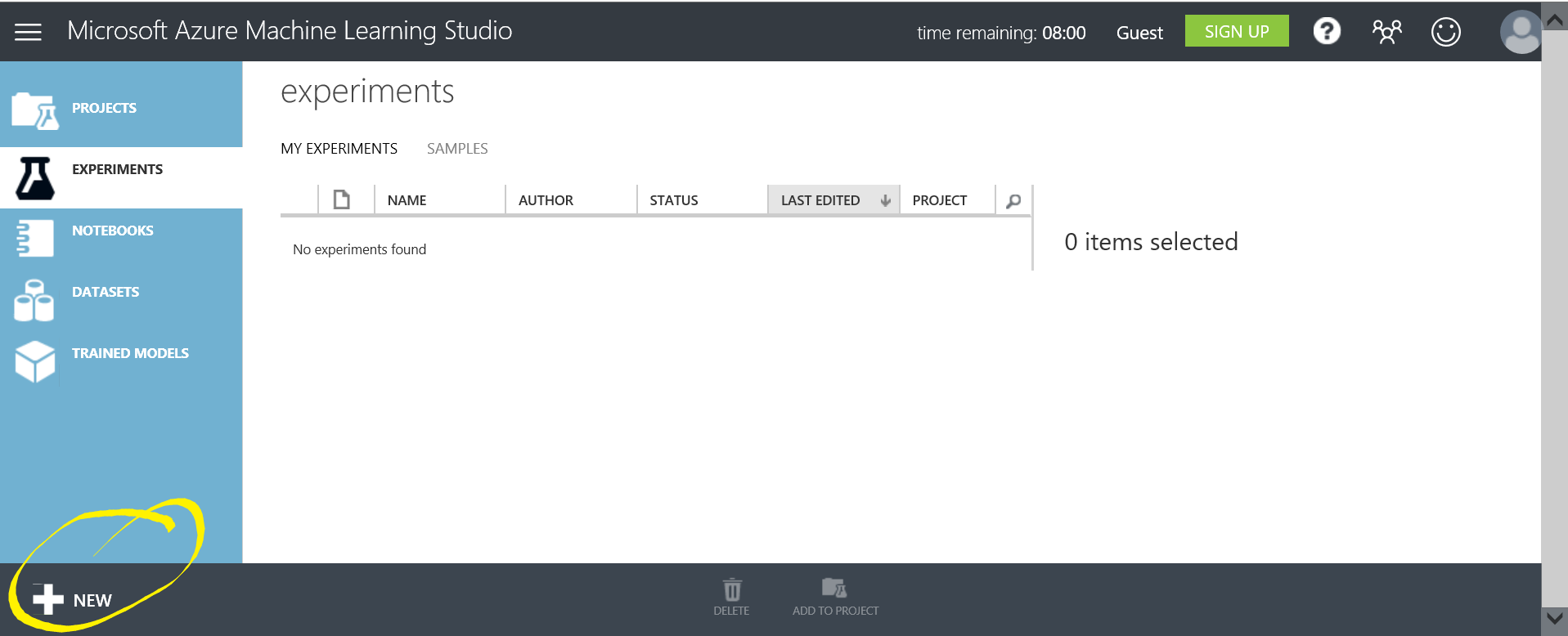
Full SLA, Bring your own Azure Storage, parallel graph execution, Elastic Web Service Endpoints

## 2.2 Create an Experiment

Let's get started by making a new experiment.

### A. Make a new experiment

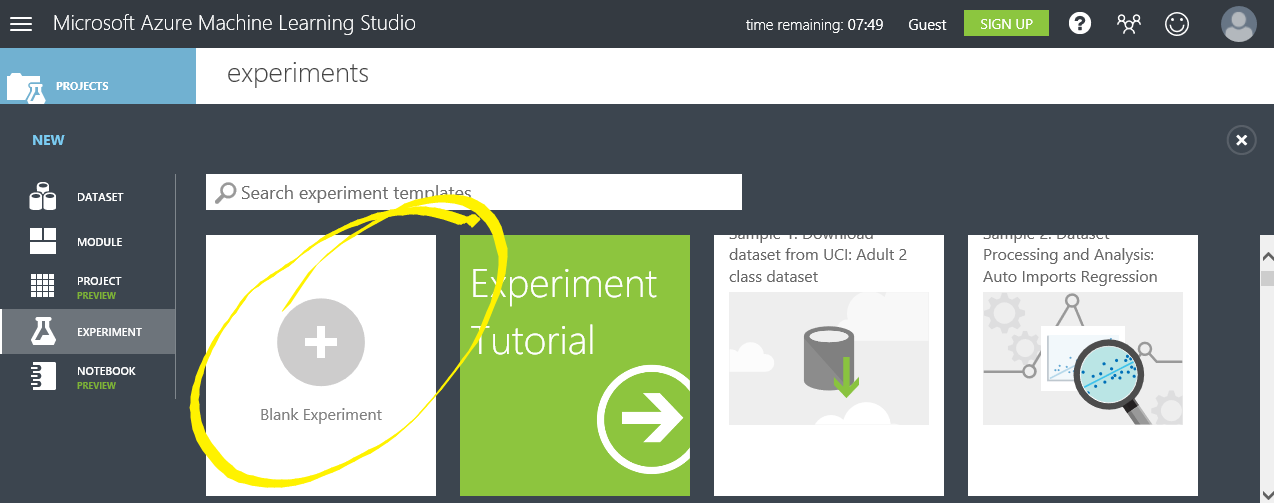
Select +New in the lower left corner.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/2ANewExperiment.png)



### B. Select Blank Experiment

To the right of Experiment, you will see a tile with a plus sign and the words Blank Experiment. Select + Blank Experiment.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/2bBlankExperiment.png)

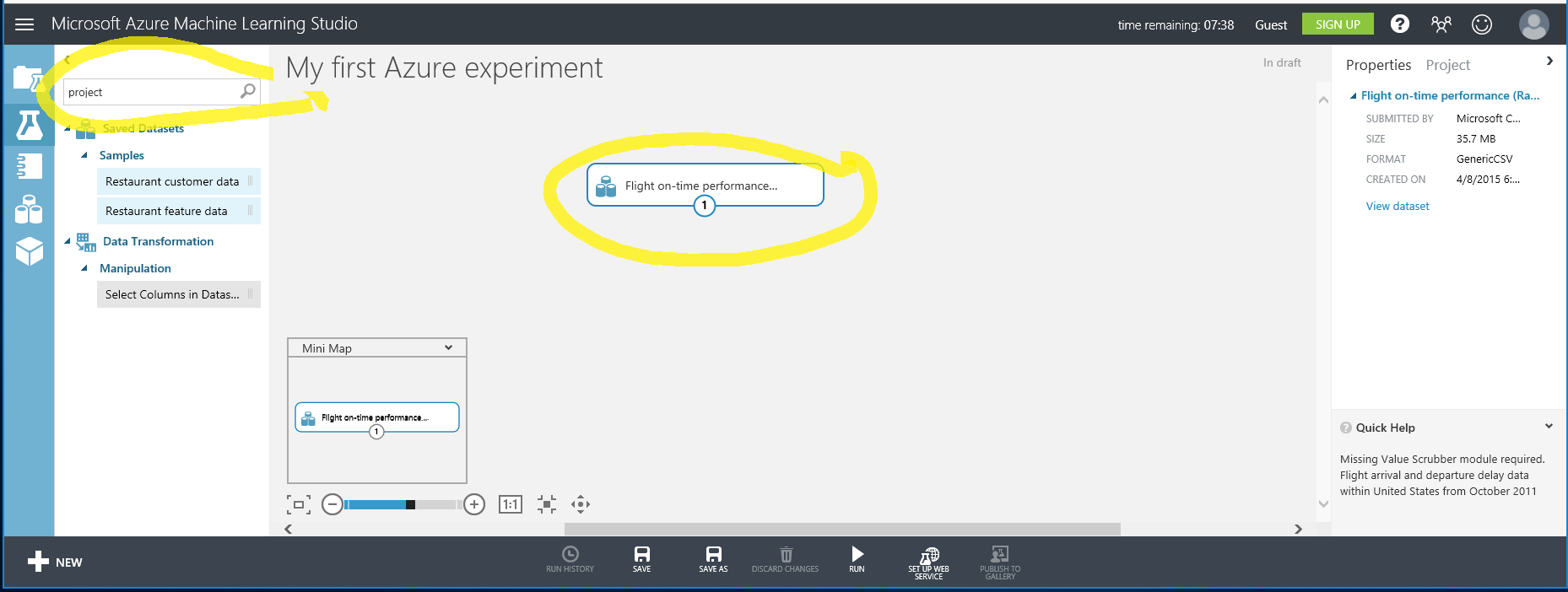


### C. Give the experiment a title

By default, a title is created with a name like "***Experiment created on 9/24/2016***". Change the title to "My first Azure ML experiment" by editing the provided title.

**B. Import data**

Drag the Flight on-time performance Dataset to the workspace as show in the image. This is one of many sample datasets built into Azure Machine Learning Studio designed to help you learn and explore the tool.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/3b.png)



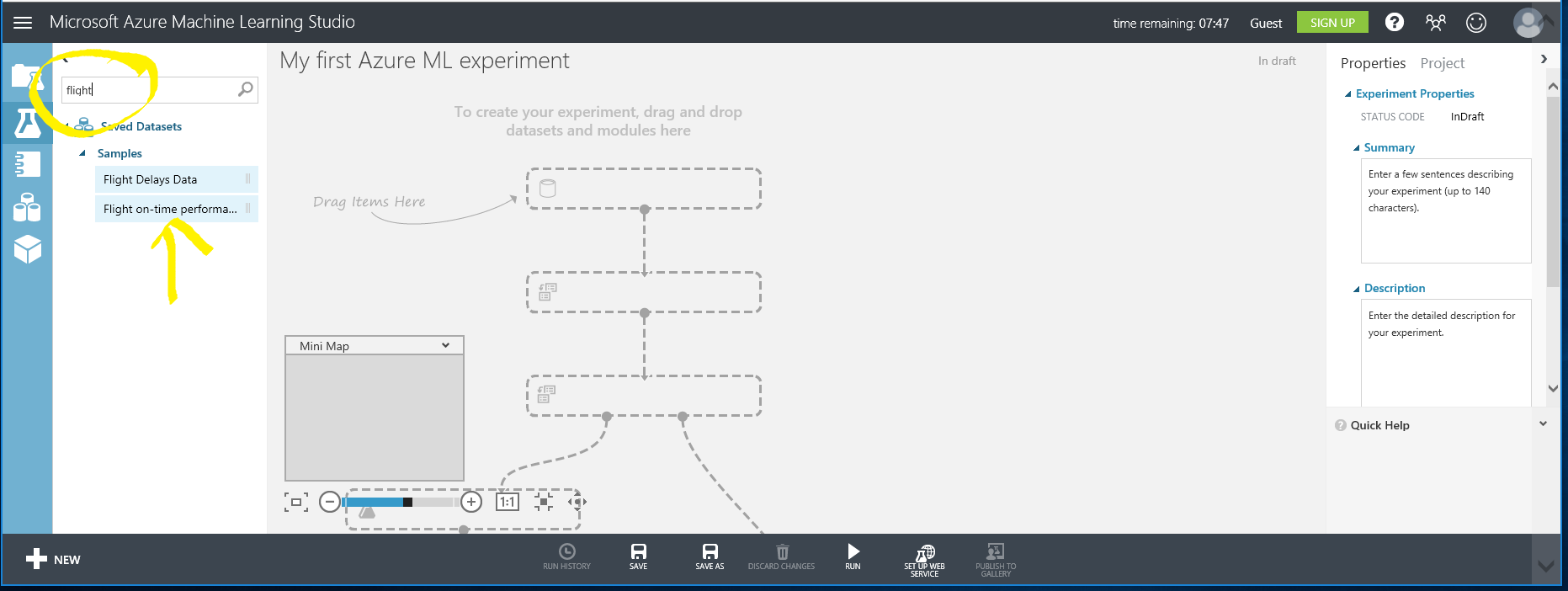
## <https://cloud.githubusercontent.com/assets/6098674/18649061/7e817444-7e8b-11e6-815d-2a7dd7b154ac.png>



## 2.3 Import, Review and Clean Data

### A. Search for flight data

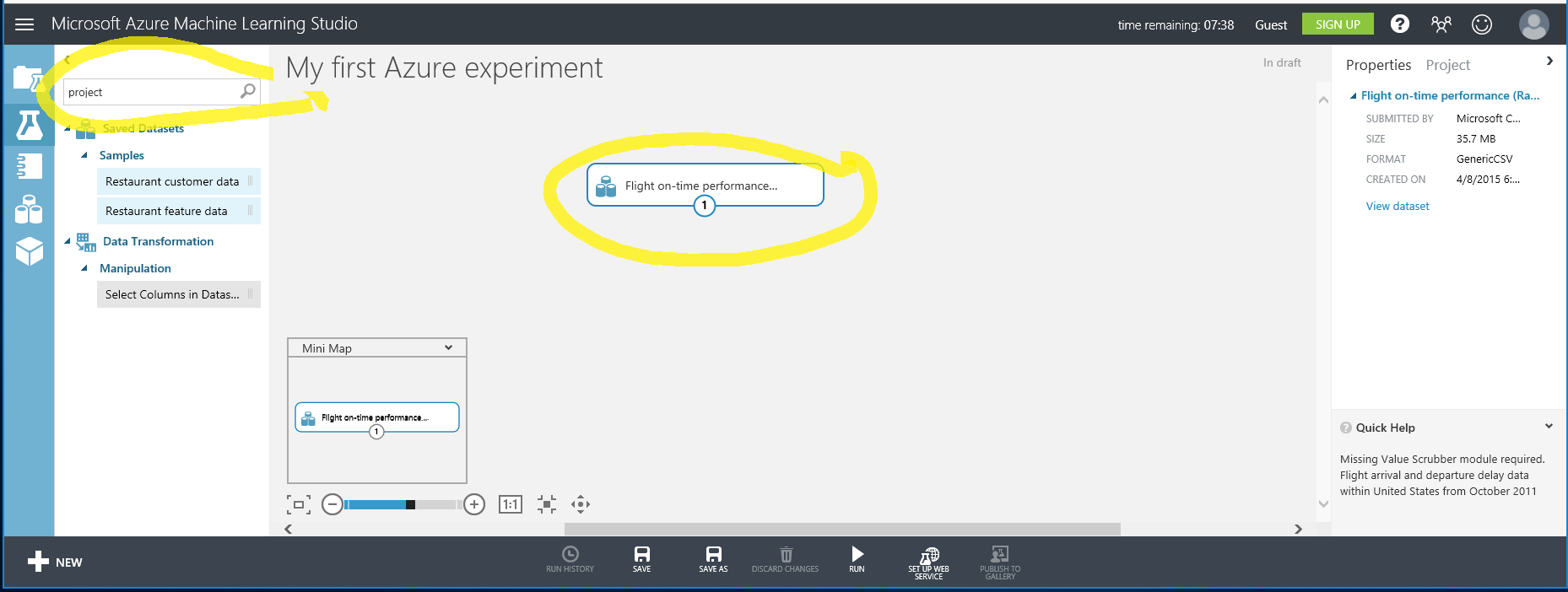
Type “flight” into the search bar.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/3A.png)



### B. Import data

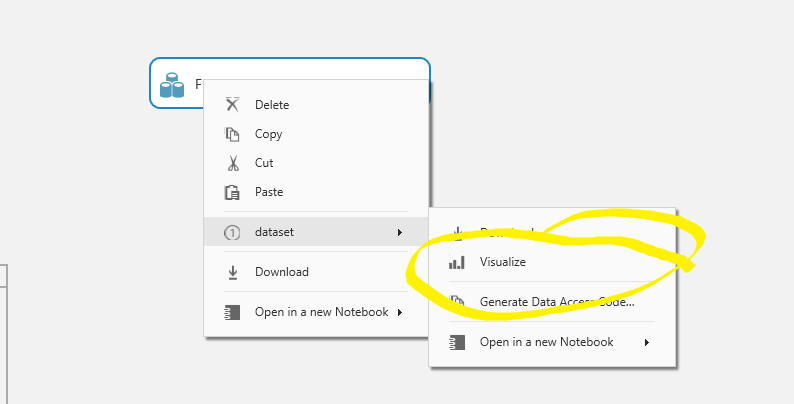
Drag the Flight on-time performance Dataset to the workspace as show in the image. This is one of many sample datasets built into Azure Machine Learning Studio designed to help you learn and explore the tool.

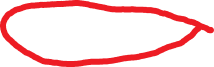
[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/3b.png)



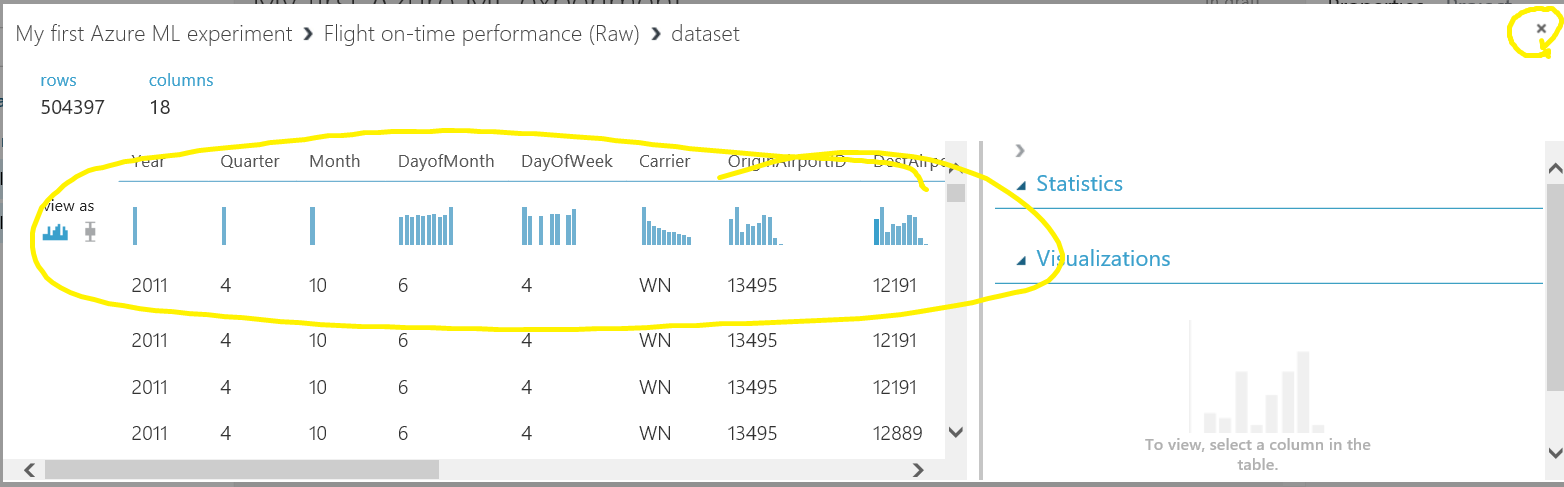
### C. Review Data

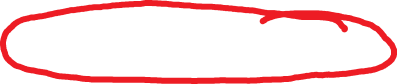
Right click on the dataset on your worksheet and select **dataset | visualize** from the pop-up menu.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/3cvis.png)



Notice the graphs or charts at the top of each data column. Explore the dataset by clicking on different columns. It’s essential in Machine Learning to be familiar with your data and visualizing your dataset is a great first step. This dataset provides a great deal of information about flights and whether or not they arrived on time. We are going to use Machine Learning to use this data to create a model that predicts whether a given flight will be late.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/3cdataset.png)



Note: In an actual data science experiment, it is likely going to be necessary to data wrangle or clean dirty data. For this example, the data set is clean enough for our use.

You can find the column definitions for this data on the [US Department of Transportation site](http://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236&DB_Short_Name=On-Time), http://www.transtats.bts.gov/DL\_SelectFields.asp?Table\_ID=236&DB\_Short\_Name=On-Time.

### D. Close the data visualization window

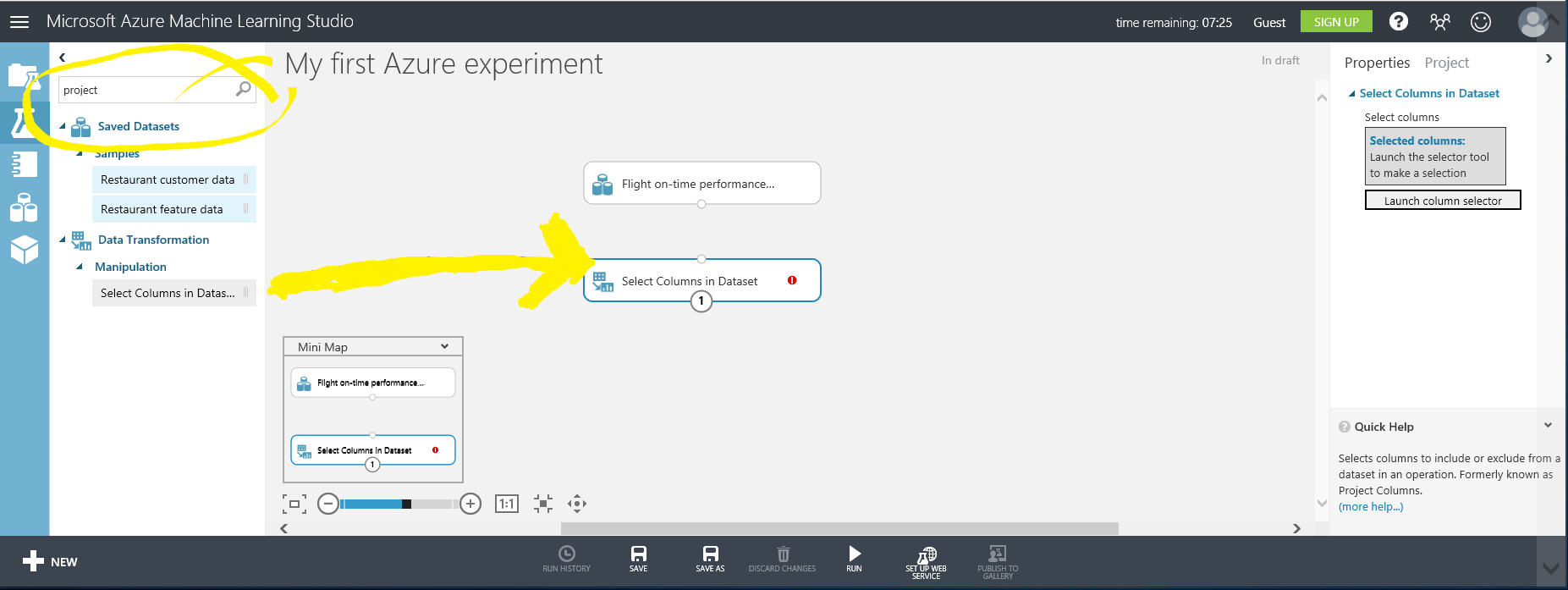
Click on the X in the top right corner of the window to close the data visualization window..

## 2.4 Specify Columns to Use

You need to review the data in the dataset and decide which columns represent data that you think will affect whether or not a flight is delayed. You also need to select the column that you want to predict. In this case, we are going to predict the value of ArrDel15. This is a binary state, 0/1, that indicates whether a flight arrival was delayed by more than 15 minutes.

### A. Add Manipulation to Select Columns in Dataset

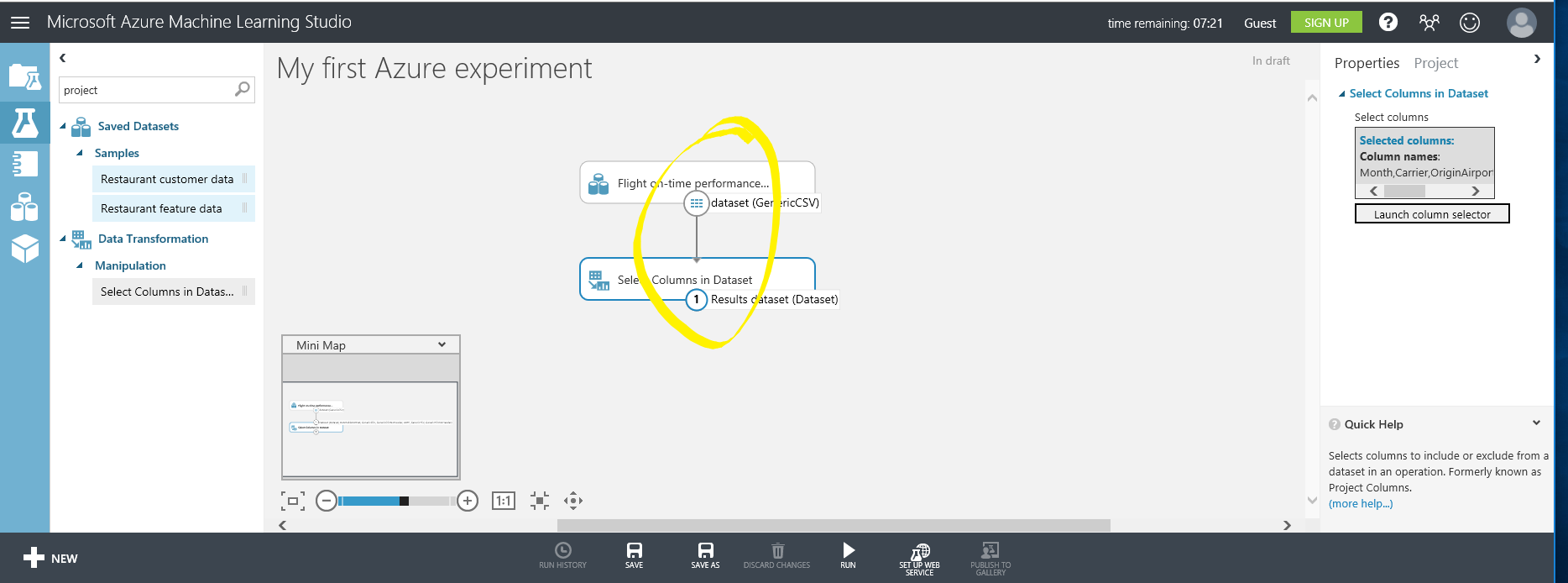
First, type "**project**" into the search bar and drag the Select Columns in Dataset manipulation to the workspace. This manipulation enables you to specify which columns in the data set you think are significant to the prediction.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/4a.png)



### B. Connect Flight on-time performance task to Select Columns in Dataset task

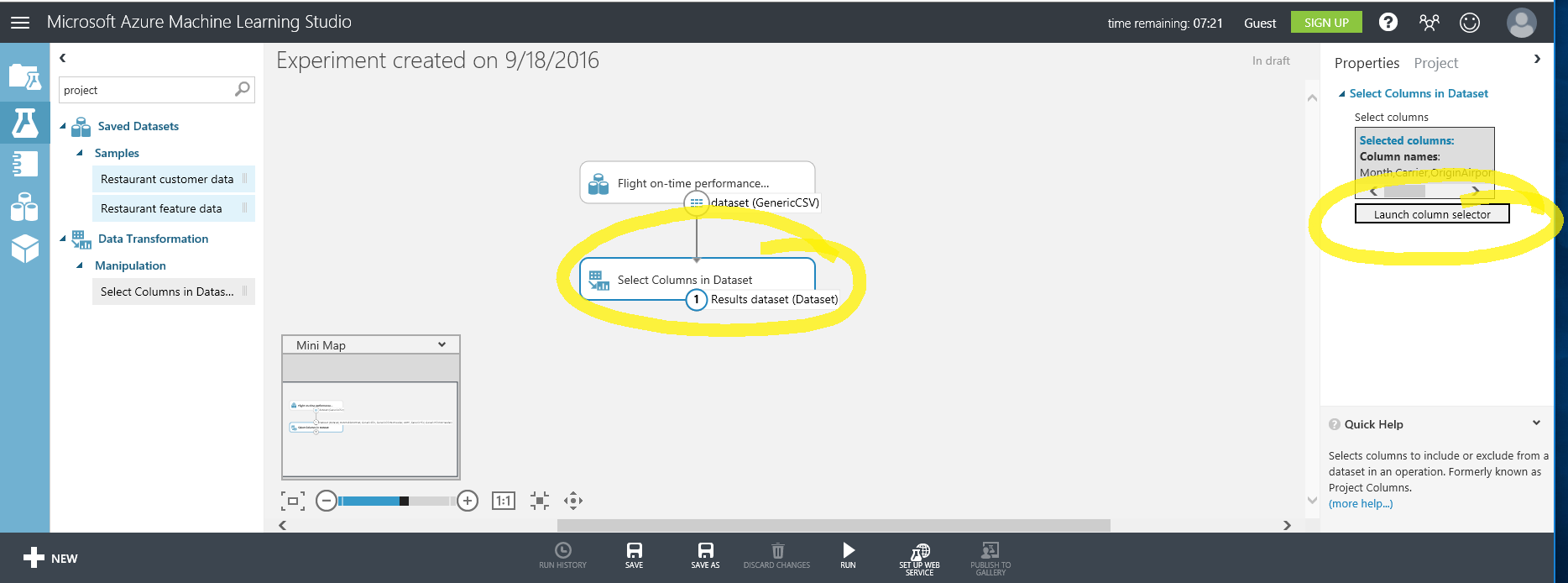
Connect the output of Flight on-time performance dataset to the input of the Select Columns in Dataset by clicking on the lower center dot and dragging to the input, top center dot, of the Select Columns in Dataset task.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/4b.png)



### C. Launch Column Selector

Click on the Select Columns in Dataset module, then on the far right, select **Launch column selector**.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/4c.png)



### D. Select Columns

Select the columns you think affect whether or not a flight is delayed as well as the column we want to predict ArrDel15. In the following screenshot, I selected Month, Carrier (airline), OriginAirportID, DestAirportID, DayofWeek, ArrDel15, CRSDepTime, Cancelled, Diverted, DepDelay, DayofMonth. You might select more or less columns. You can also add or remove columns later. The columns used will affect the prediction.

### 



### E. Complete Column Selection

Select the checkbox in the lower right of the **Select columns** window.

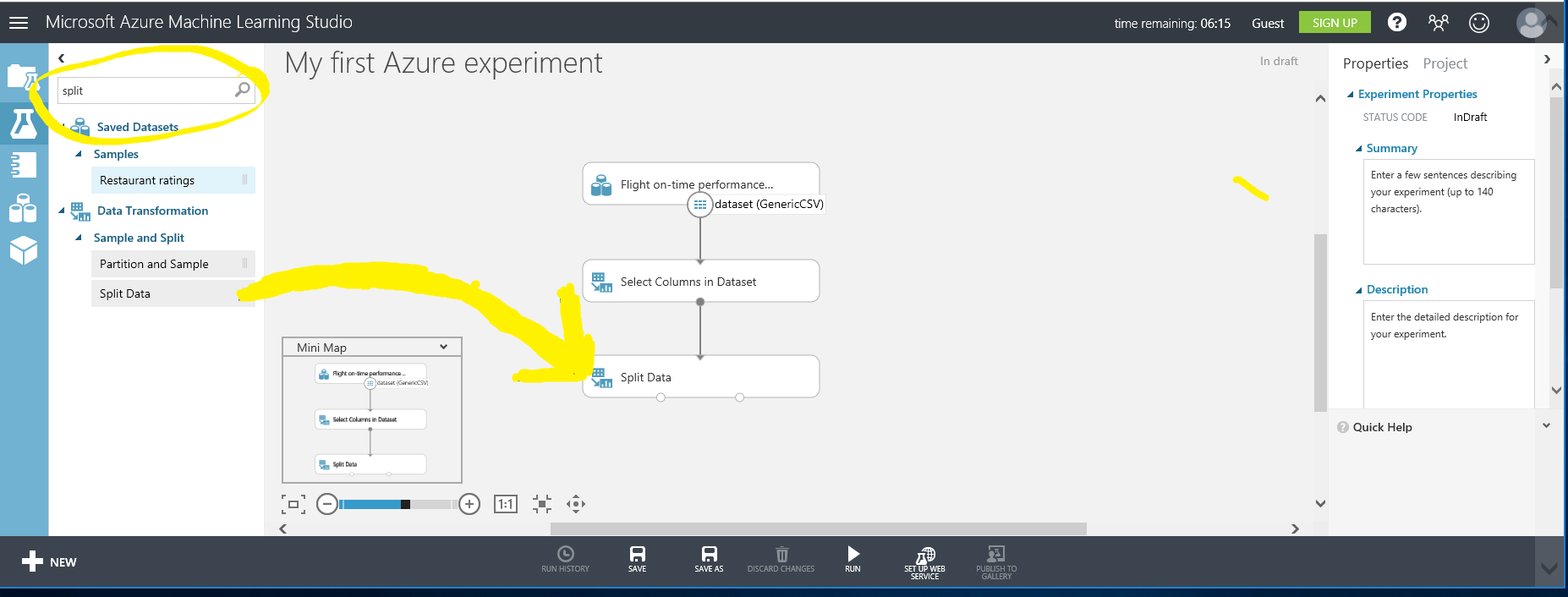
## 2.5 Split The Data Into A Training And Test Set

The Split Data task allows us to divide up our data, we need some of the data to try and find patterns so we can make predictions. We need to save some of the data to test if the model we create successfully makes predictions.

Traditionally, you will split the data 80/20 or 70/30. For today’s challenge everyone will use an 80/20 split. That means 80% of the data will be used to train the model and 20% will be used to test the accuracy of the model we develop.

### A. Split Data Task

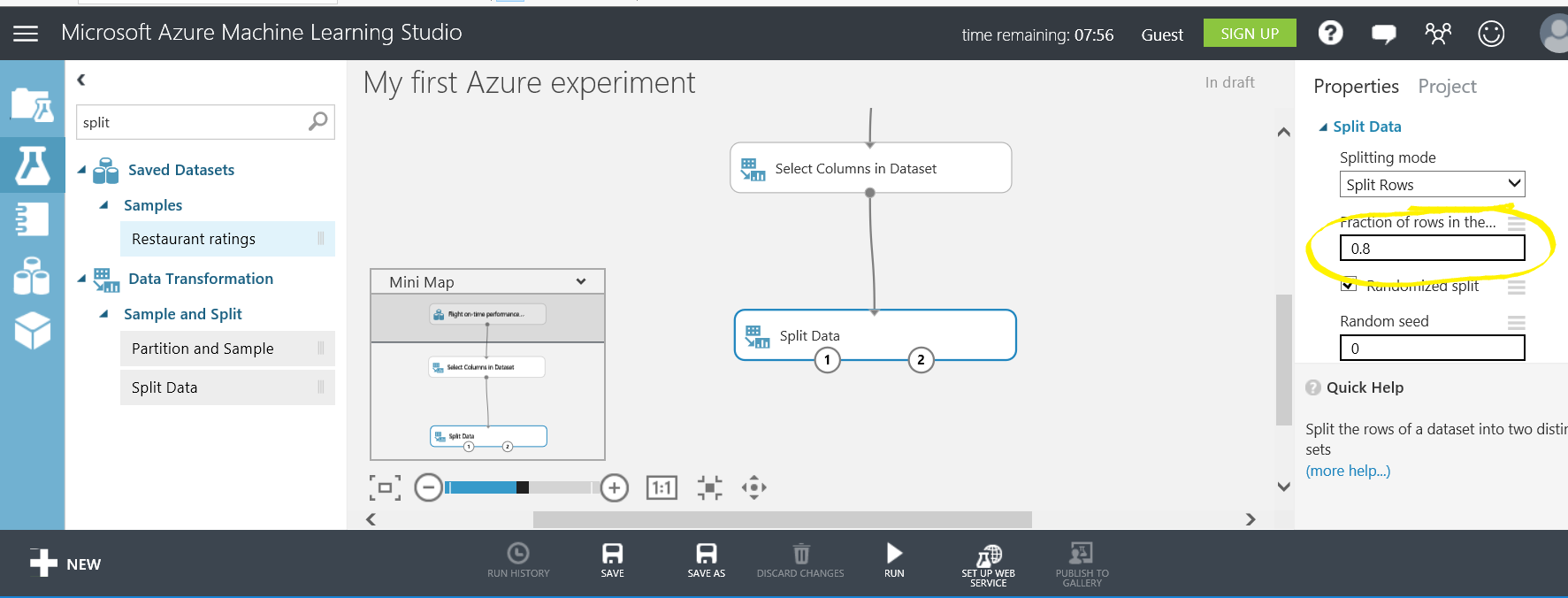
Type “**split**” into the search bar and drag the Split Data task to the workspace. Connect the output of Select Columns in Dataset task to the input of the Split Data task (same way we connected the Flight Data to the Select Columns modules).

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/5a.png)



### B. Split our input data

Click on the Split Data task to bring up the Properties Pane and specify .8 as the Fraction of rows.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/5b.png)



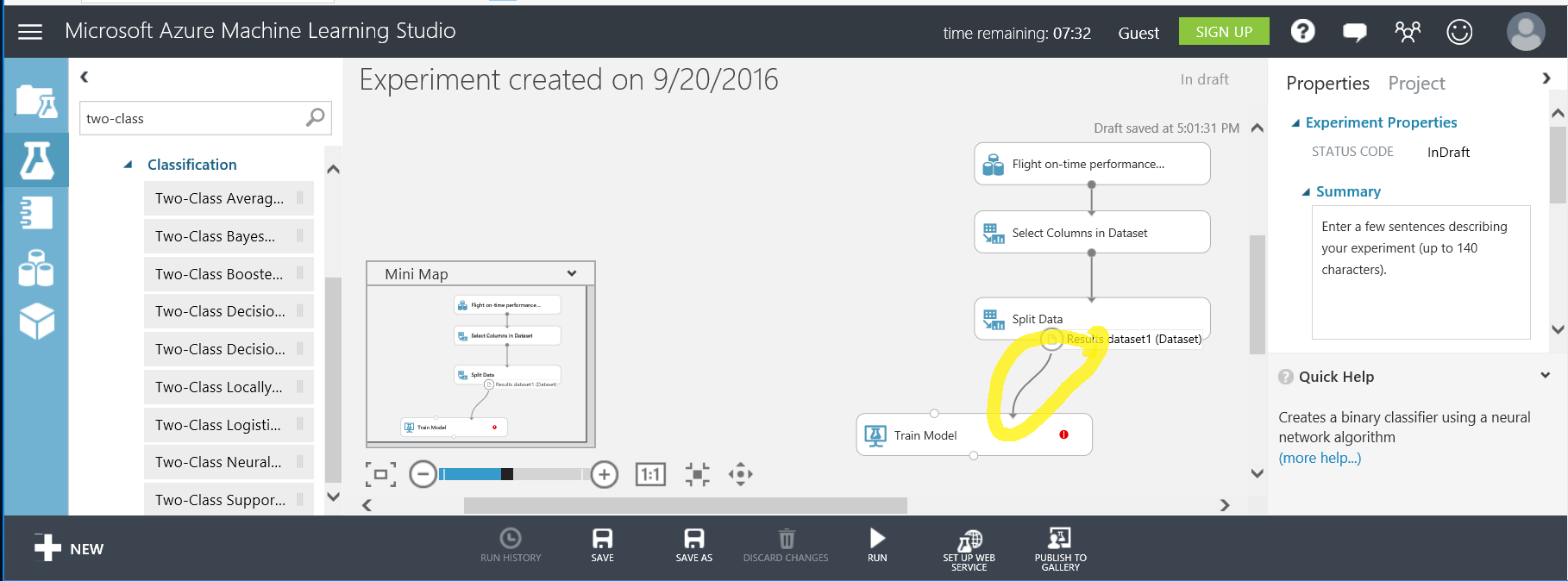
## 2.6 Train The Model

Next, we identify which data is to be predicted. In our case, we are predicting the value of the column ArrDel15 which indicates if a flight arrival time was delayed by more than 15 minutes.

### A. Connect Data

Type “**train model**” into the search bar. Drag the Train Model task to the workspace.

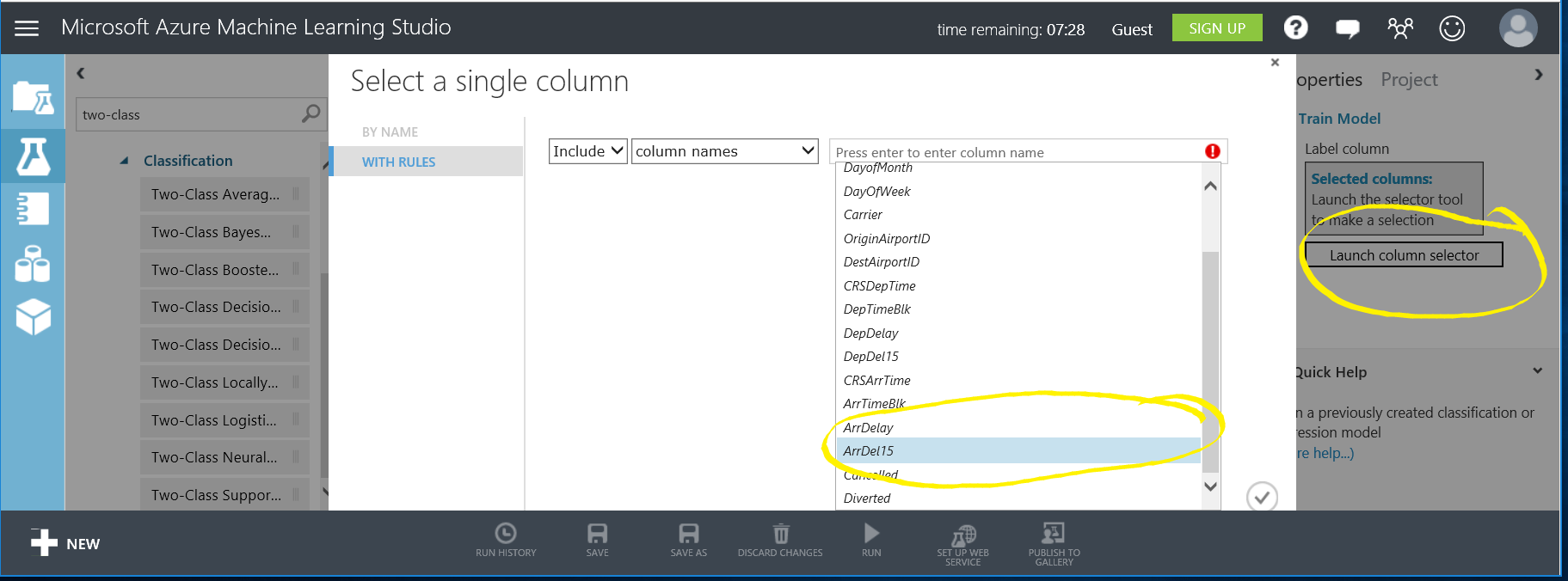
Hovering over the input and output dots will reveal what each input/output represents. Connect the first output, Results Dataset1, (the circle on the left) of the Split Data task to the **rightmost** input of the Train Model task. This will take 80 % of our data and use it to train/teach our model to make predictions.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/6a.png)



### B. Identify Predicted Value

Click on the Train Model task. In the **Properties** window, select Launch Column Selector. Select the column ArrDel15 by typing "arrdel15" in to the text box (a smart filter of columns will appear). Click the checkbox in the lower right corner to complete the operation.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/6b.png)



## 2.7 Select Algorithm

If you are a data scientist who creates their own algorithms, you could now import your own R code to analyze the patterns. But, Azure ML provides a number of standard algorithms which are available for use.

Selecting an algorithm can be overwhelming, to help narrow the process a [Azure ML Cheat Sheet](https://aka.ms/azuremlcheatsheet) has been created. By narrowing the type of problem you are solving can find the algorithms that will be most likely to generate a good model.

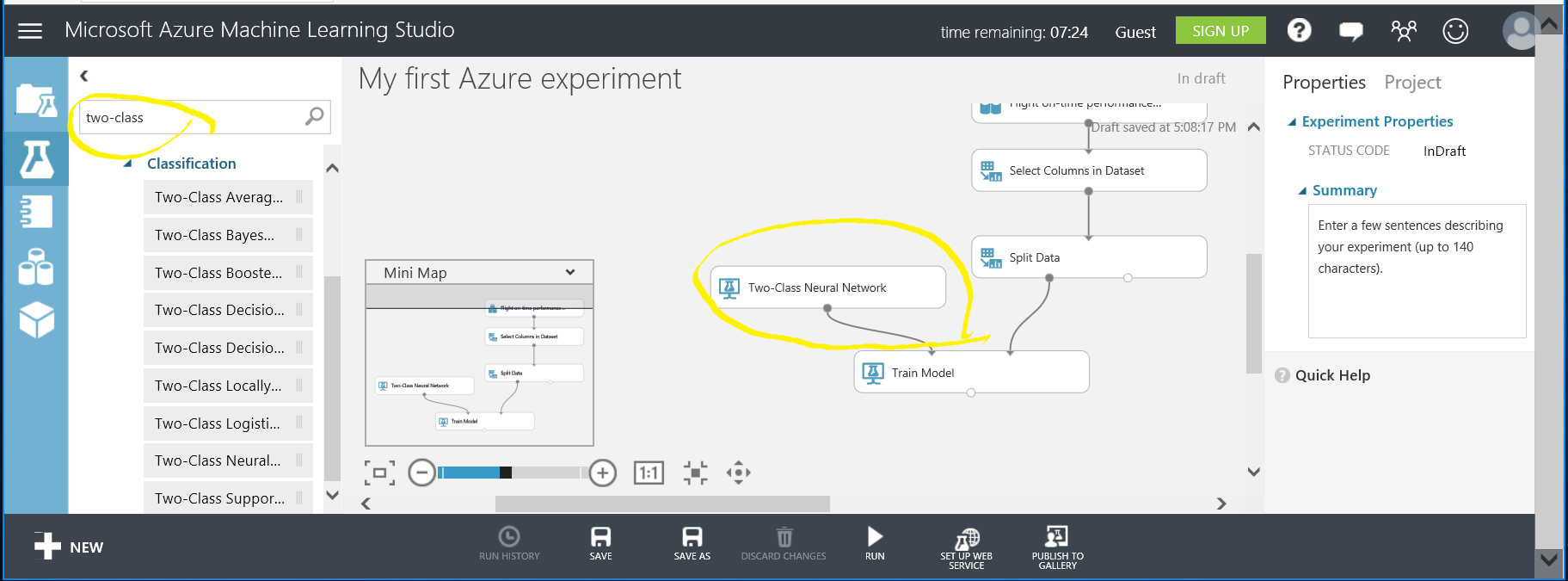
Today we are doing binary classification also known as Two-Class Classification. Using the cheat sheet we can narrow our selection to a standard algorithm called Two-Class Neural Network.

As you can see there are many Two-Class algorithms that we can choose from so we may want to try different ones out as we refine our model. Swapping out or even comparing two algorithms is made easy with Azure Machine learning as you will see.

### A. Connect algorithm

Type “**two-class**” into the search bar. You will see a number of different classification algorithms listed and each has its own advantages and disadvantages. Each of the two-class algorithms is designed to predict a binary outcome.

Select Two-Class Neural Network and drag it to the workspace. Connect the output of the Two-Class Neural Network task to the **leftmost** input of the Train Model task.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/7a.png)



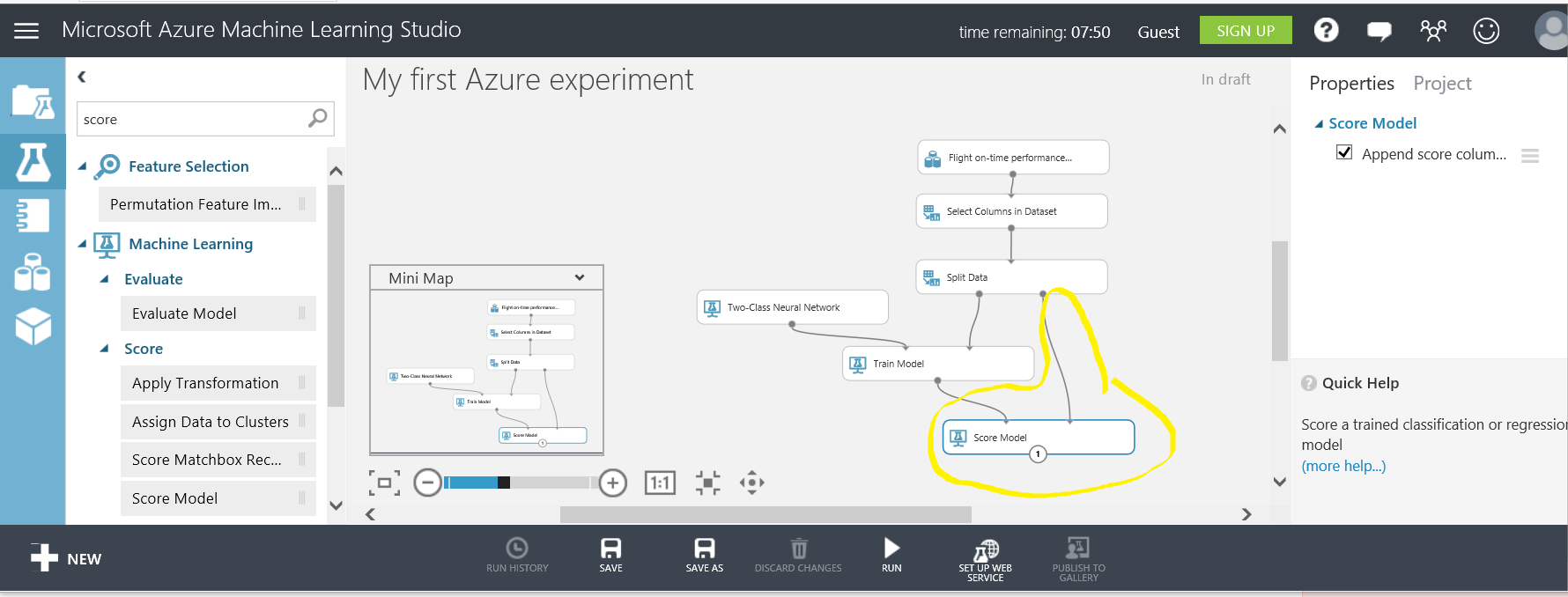
## 2.8 Score the Model

After the model is trained, it is evaluated to determine how well it predicts delayed flights, so the model is scored by testing it against the Test Data which is the remaining 20% of the data we split to the second output of the Split Data task.

### A. Connect test data

Type “**score**” into the search bar and drag the Score Model task to the workspace.

Connect the output of Train Model to the **left input** of the Score Model task. Connect the Test Data, the **right output** of the Split Data task to the **right input** of the Score Model task as shown in the following screenshot. The output of this task is a scored dataset.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/8a.png)



## 2.9 Evaluate Model

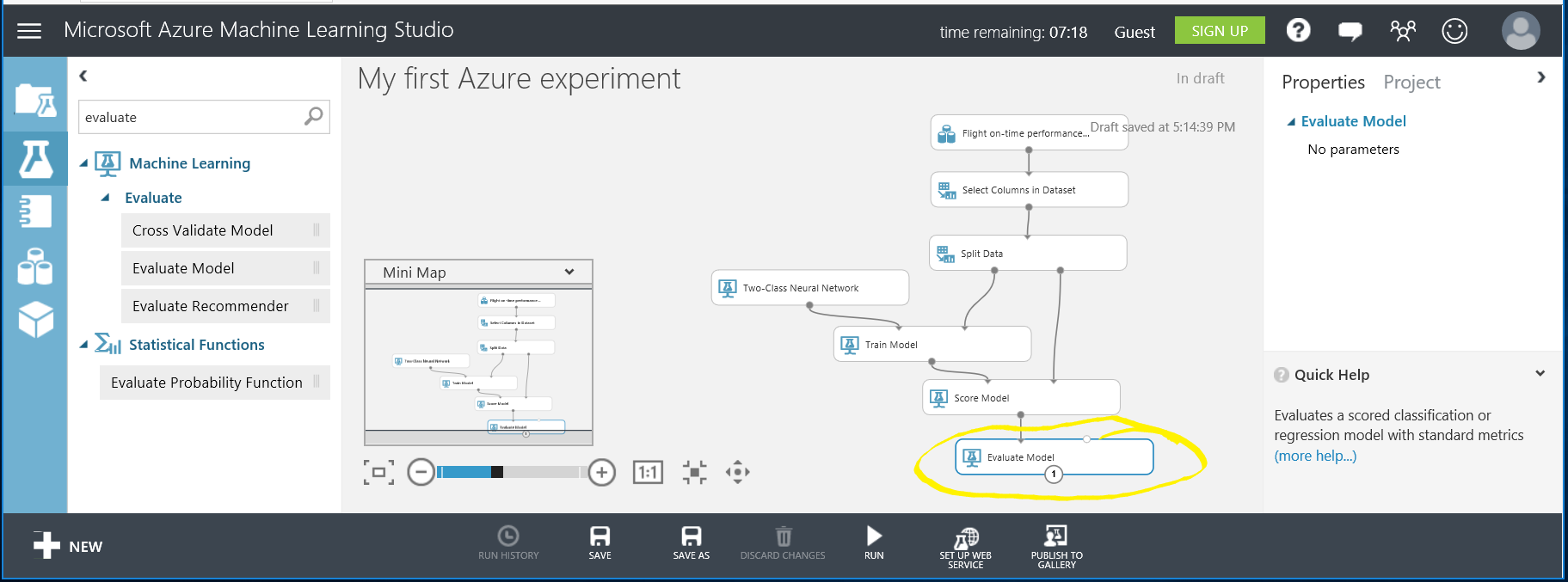
Next, the model is evaluated to determine its accuracy. This is done by evaluating the trained model by using the test data.

### A. Determine accuracy of model

Type “**evaluate**” into the search bar and drag the Evaluate Model task to the bottom of the workspace.

Connect the output of the Score model task to the **left input** of the Evaluate Model task. The other input and output of the Evaluate Model task are not connected at this time.

You are now ready to run your experiment!

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/9a.png)

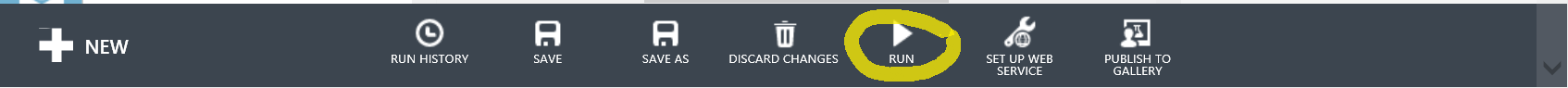


## 2.10 Run Experiment

### A. Select Run

Select Run on the bottom toolbar. You will see green check marks appear on each task as it completes. The data is flowing through your Machine Learning Workflow, starting with data selection, being trained against the model, and finally being evaluated.

This process can take several minutes. When there is a green check mark on the Evaluate Model task the process is complete.

[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/10a.png)

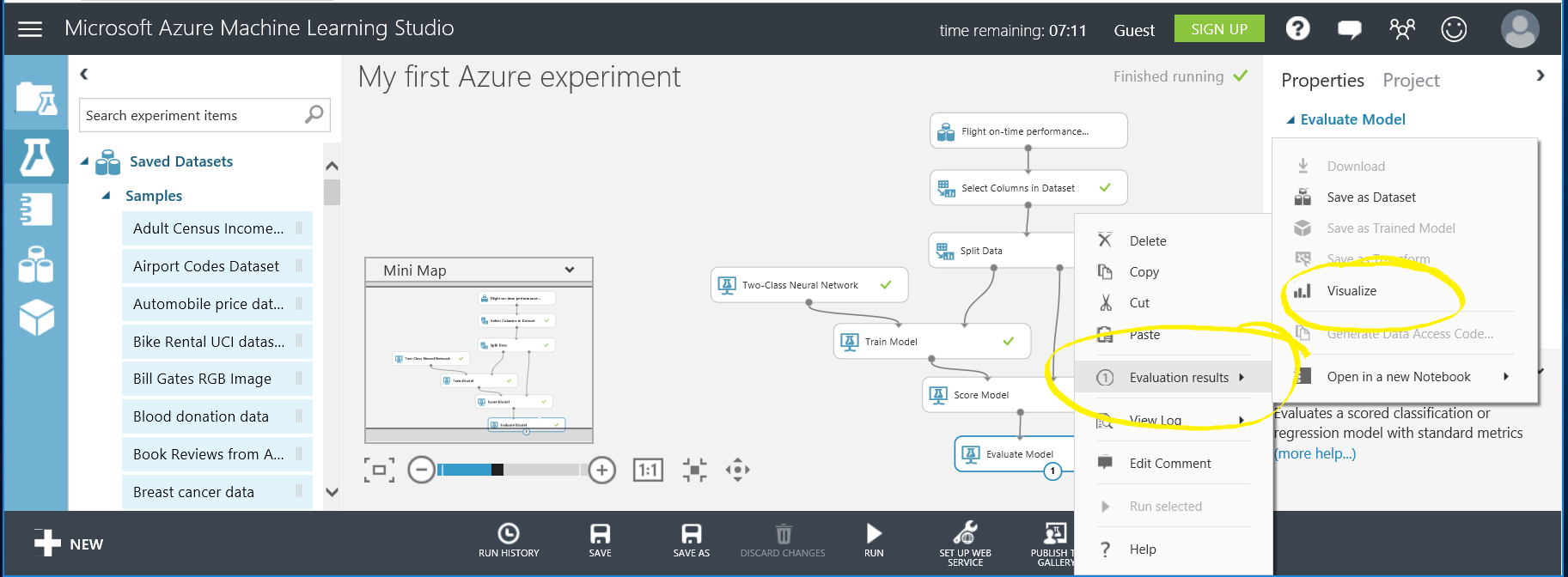


## 2.11 Post Run: Evaluate Model

It is usually necessary to evaluate the model, improve it, re-run it and repeat.

### A. Evaluate The Model

When the entire experiment is completed, right click on the Evaluate Model task and select “**Evaluation results | Visualize**” to see how well the model predicted delayed flights.

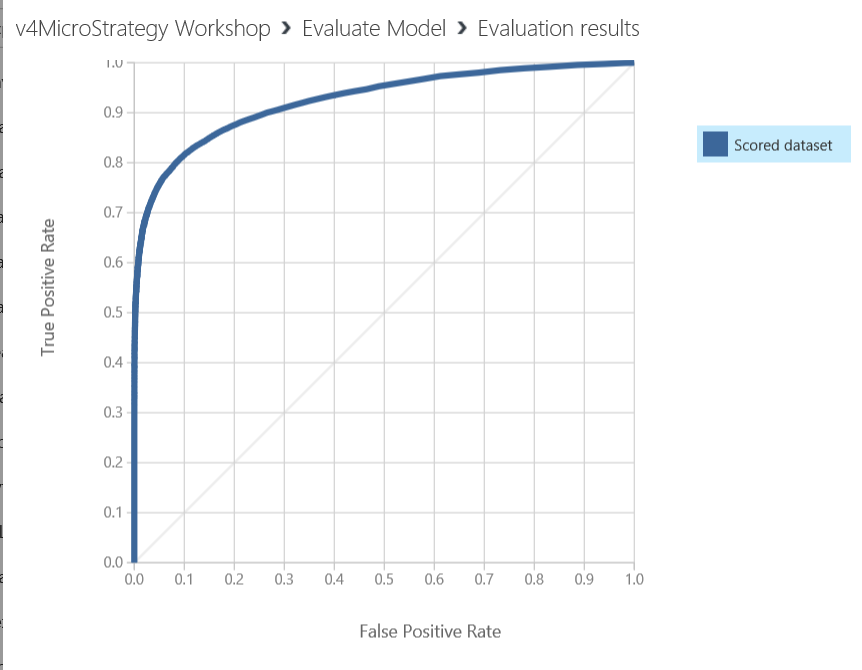
[](https://github.com/bethz/AzureML-FlightPrediction/blob/master/images/11a.png)



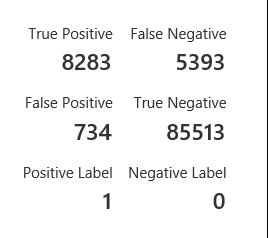
## 2.12 Interpreting Results

The first run of a model is a baseline and is considered a first step.

One useful piece of the evaluation results is the first graph the **True Positive Rate versus False Positive Rate**. This graph is a representation of the Area Under the Curve. A 45 degree flat line on this chart indicates guessing randomly. A more accurate model than random guessing looks like the image below, our current model.



Model Accuracy



If you scroll down you can see the accuracy – Higher accuracy is good! You can also see the number of false and true positive and negative predictions. - **True positives** are how often your model correctly predicted a flight would be late - **False positives** are how often your model predicted a flight would be late, when the flight was actually on time (your model predicted incorrectly) - **True negatives** indicate how often your model correctly predicted a flight would be on time (arrDel15 is false) - **False negatives** indicate how often your model predicted a flight would be on time, when in fact it was delayed (your model predicted incorrectly)

You want higher values for True positives and True negatives, you want low values for False Positives and False negatives.

From the model, there were only a few False Positives which is good. There are a few False Negatives which can be considered for future work. There were a good number of True Positive and True Negative which indicates the model predicted those correctly and this is a very solid attempt at prediction. If planned for production, we would iterate by changing the algorithm or choosing additional data inputs or cleaning up more of the data.

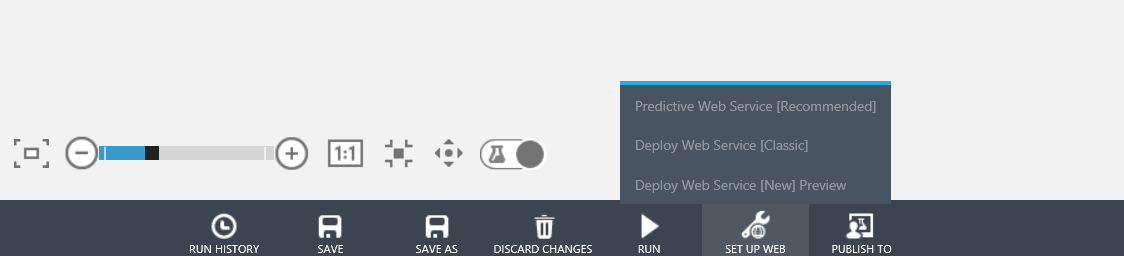
Congratulations! You have created a training experiment!

Publish Model as a Web Service

Once the model is created, we have 2 additional steps to publish it on the web:

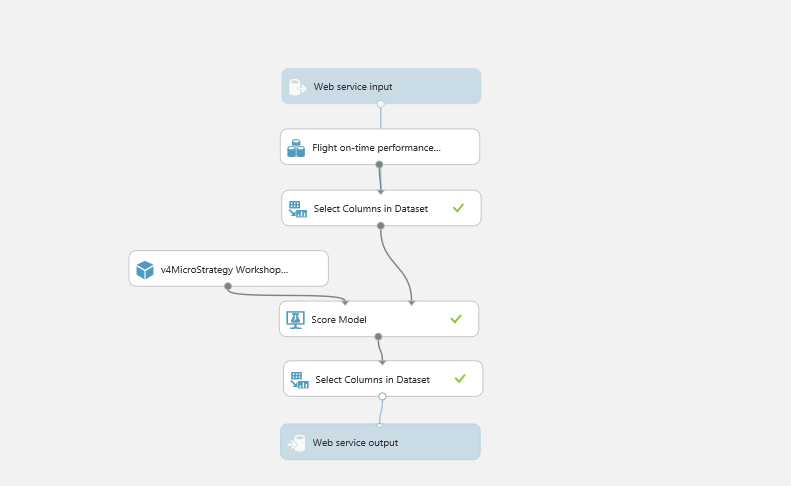
1. Convert training experiment to a predictive experiment
2. Deploy it as a Web Service
3. Create a predictive experiment

Create a predictive experiment by clicking Predictive Web Service.



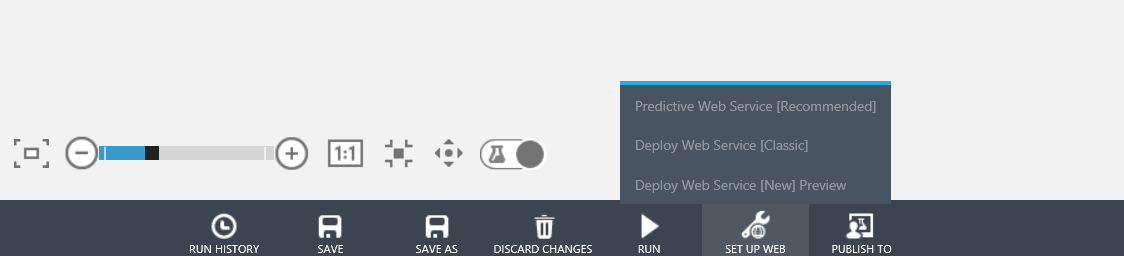


Your model will be automatically modified to add web inputs/outputs and the trained model as shown below.



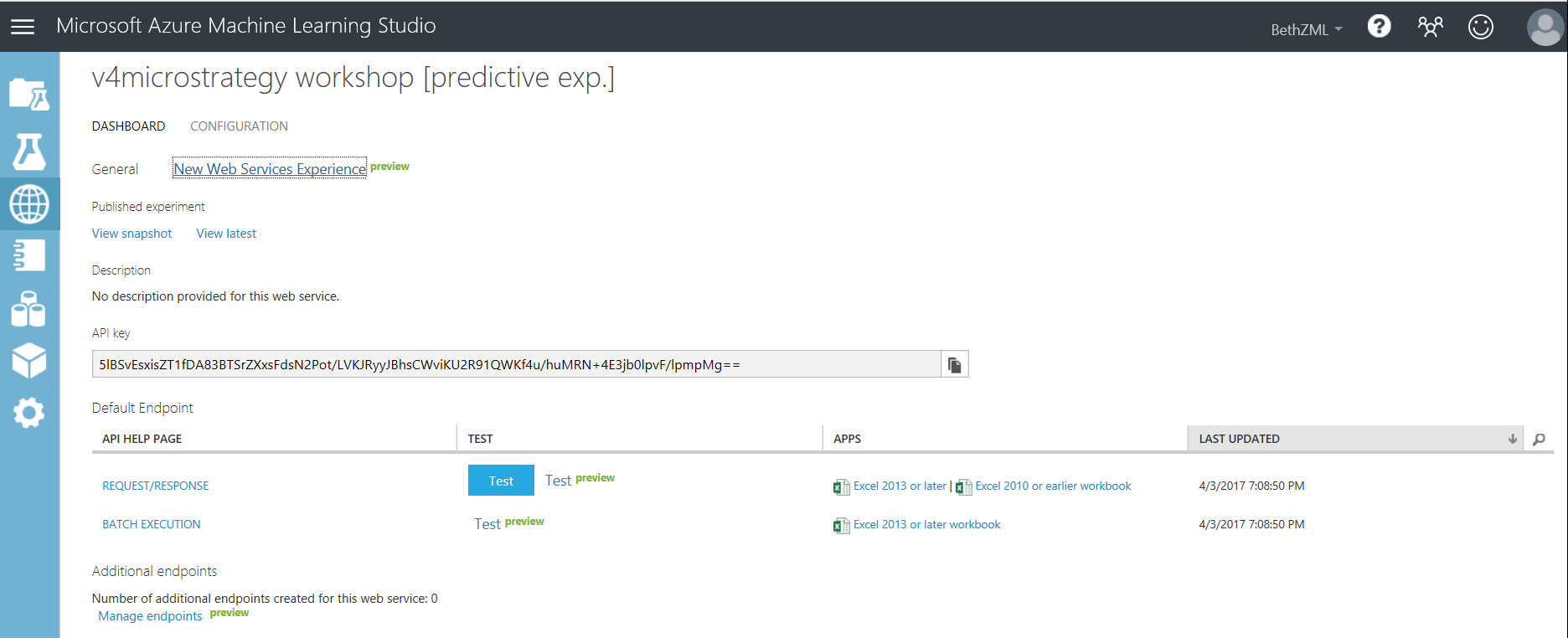
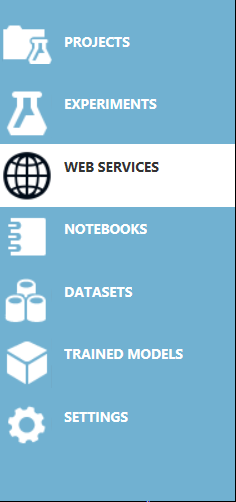
1. Publish a Web Service

Next, click Deploy Web Service [Classic] and you have an active Web Service ready for consumption.

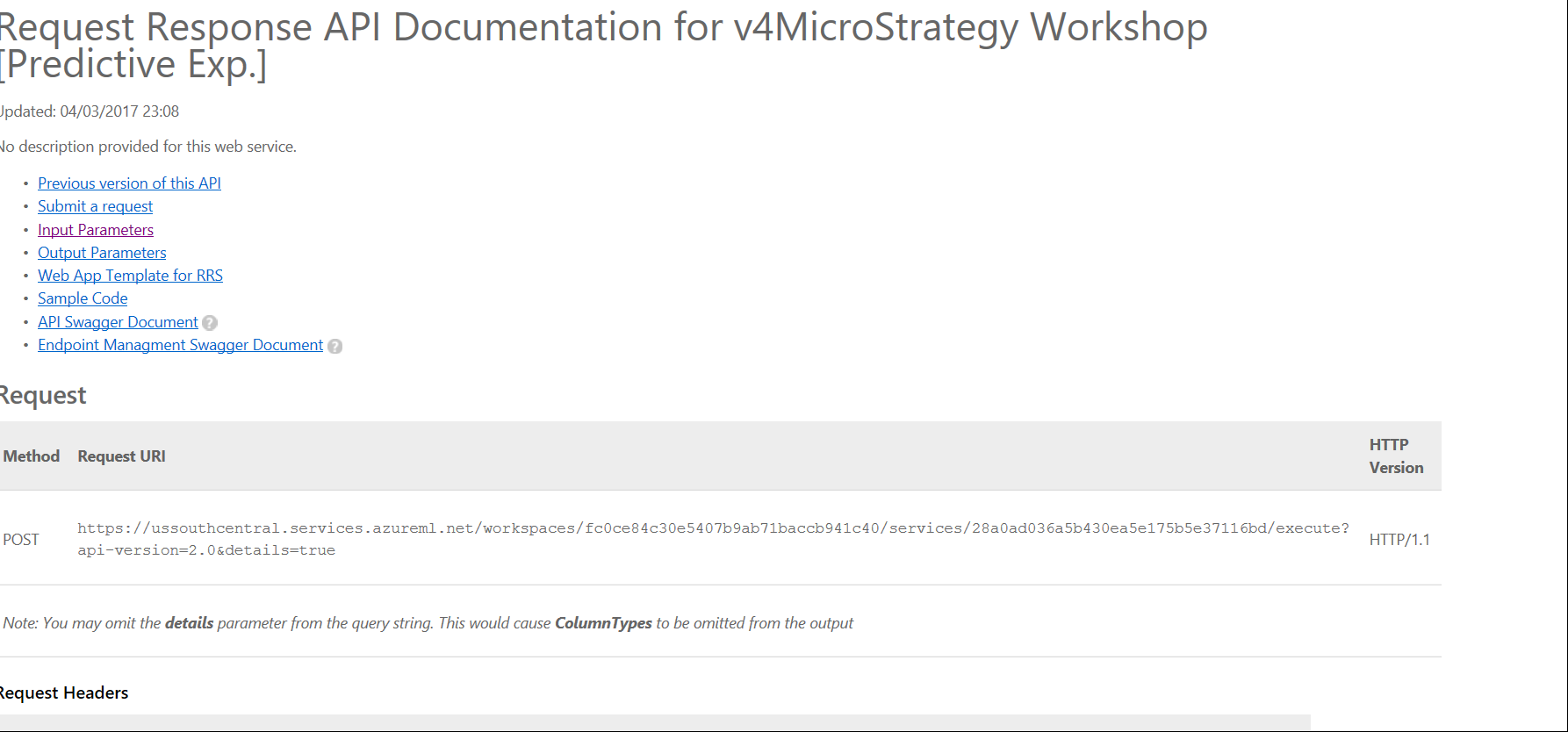




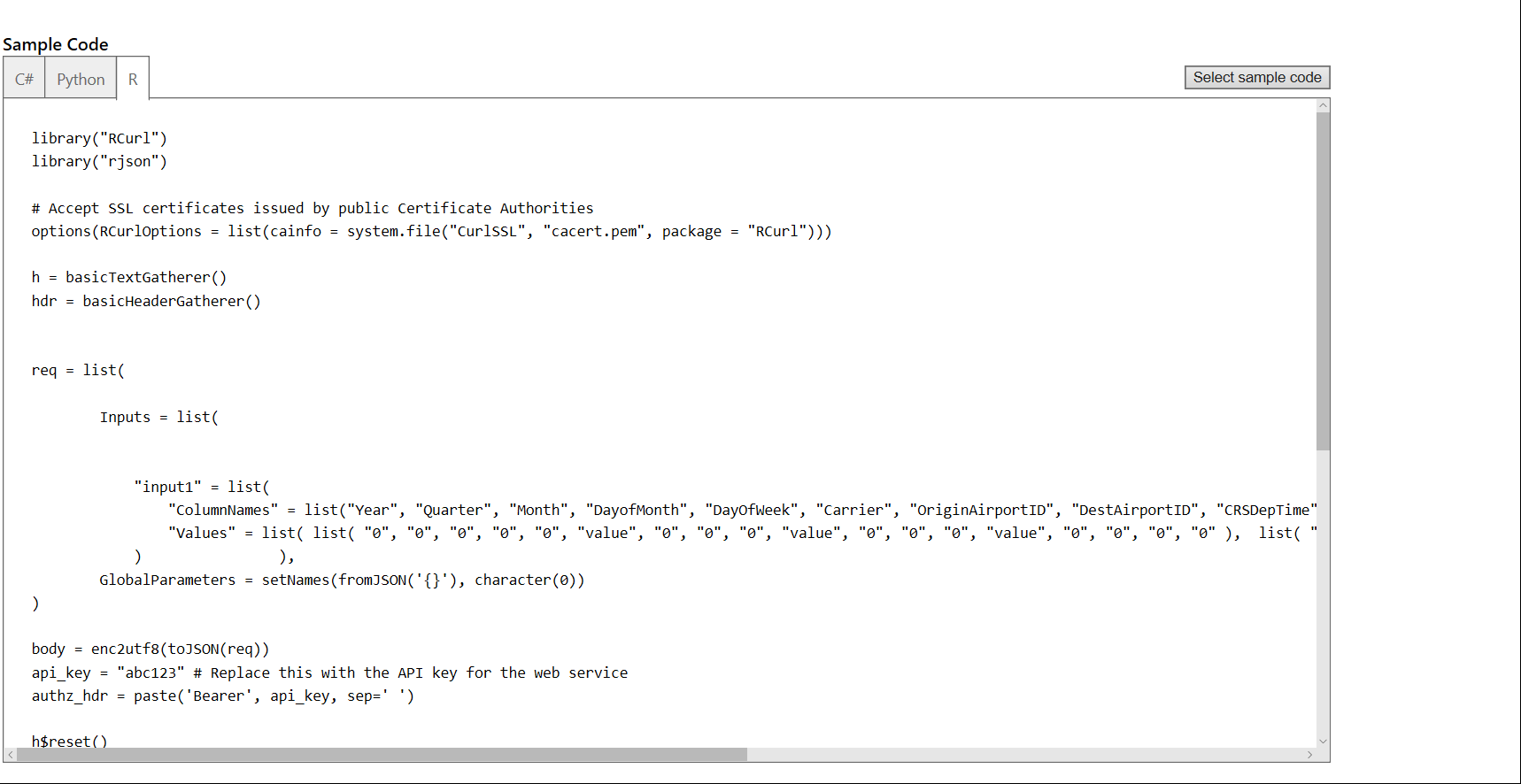
Here is how to obtain the API Key and URI as input to MicroStrategy.

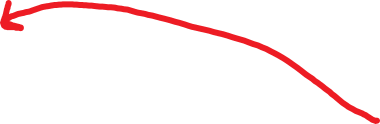












1. Step-by-step MicroStrategy Workstation

Downloading R script file and data file

1. Ask user to download the R script file and excel data file, and put them under C: drive.

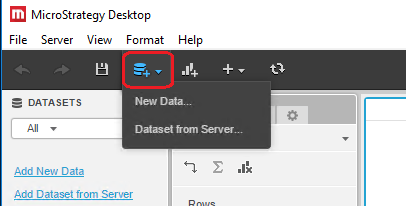
Getting started

1. Launch MicroStrategy Desktop.



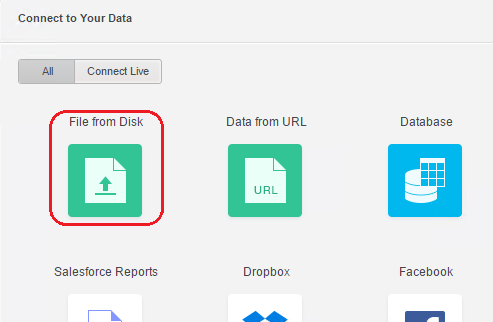
1. If this is the first time you are accessing MicroStrategy Desktop, the Quick Tips guided product tour will be displayed. Click Hide Quick Tips to remove the notes.

Connect to your data

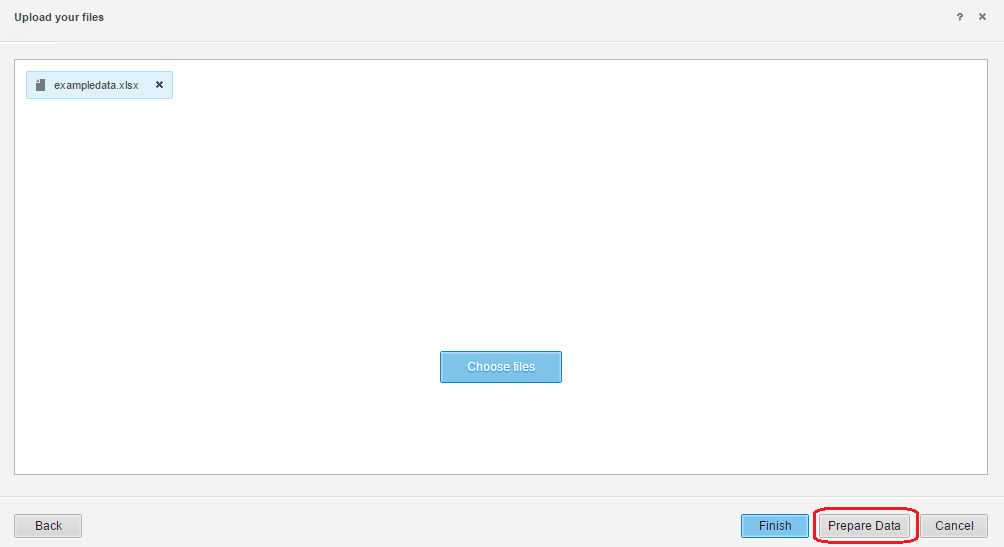
1. From the Dashboard Datasets panel, click on the **Add Data** button
2. Click **New Data,** The Connect to Your Data wizard comes up.

TIP: MicroStrategy Desktop allows you quickly and intuitively connect to almost any data source, from a tabular data file like Excel spreadsheets, to Hadoop big data sources, cloud- based applications, and relational databases.

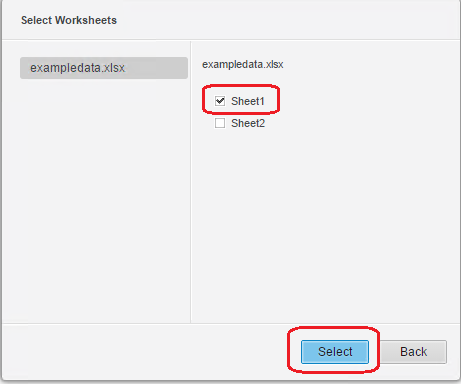
1. You want to access data stored in an Excel file, so choose the **File from Disk** option from the available data sources.



1. On the next screen, click **Choose Files** to find and import a spreadsheet from your local machine.
2. In the dialog that appears, find the excel file you have downloaded under C: drive, select it, and click Open.
3. Select the **Prepare Data** button in the dialog box to refine the data you are importing.

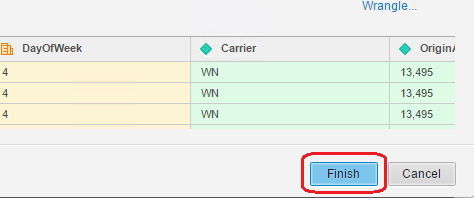


1. Only check the checkbox for “Sheet1”. Click **Select** to move on to the next step.



On the next screen you will be presented with a preview of the data. Here you can review a sample of the data in the dataset, the mapping of the columns as attributes or metrics, and the data type of each column.

1. After have a general idea of your data, and then click Finish button to finish data import.

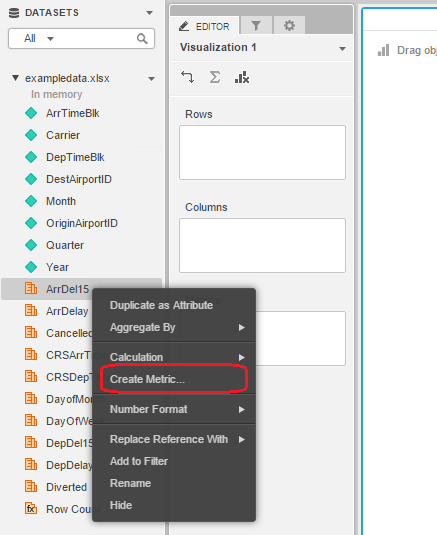


TIP: MicroStrategy automatically maps the columns as attributes (your business dimensions) and metrics (your performance indicators or KPIs) that are available for analysis. This mapping is based on the column data types, and content. If any adjustments are needed, they can be performed here.

Make Predictions on new data

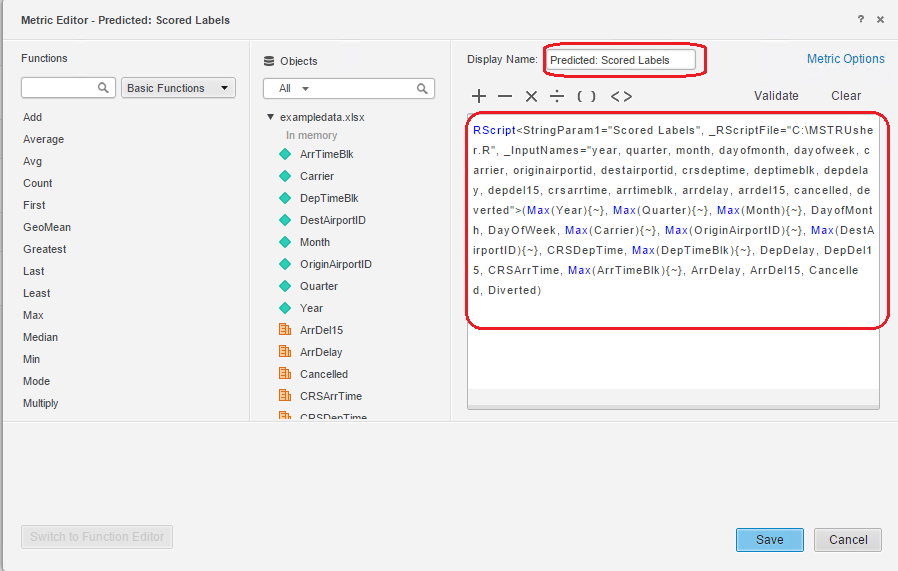
Now that you have your new data imported and prepared, you are ready to make predictions to them based on the machine learning model you have on Azure. Before you start visualizing your data, take a moment to save the dashboard.

1. Click the **Save** button. 
2. In the dialog box, navigate to the Desktop and save your dashboard as **Micorsoft Azure Workshop.** It is now saved as a shareable MicroStrategy file.
3. In the DATASET panel, right click on any metric (e.g. ArrDel15), and select “Create Metric” from the menu.



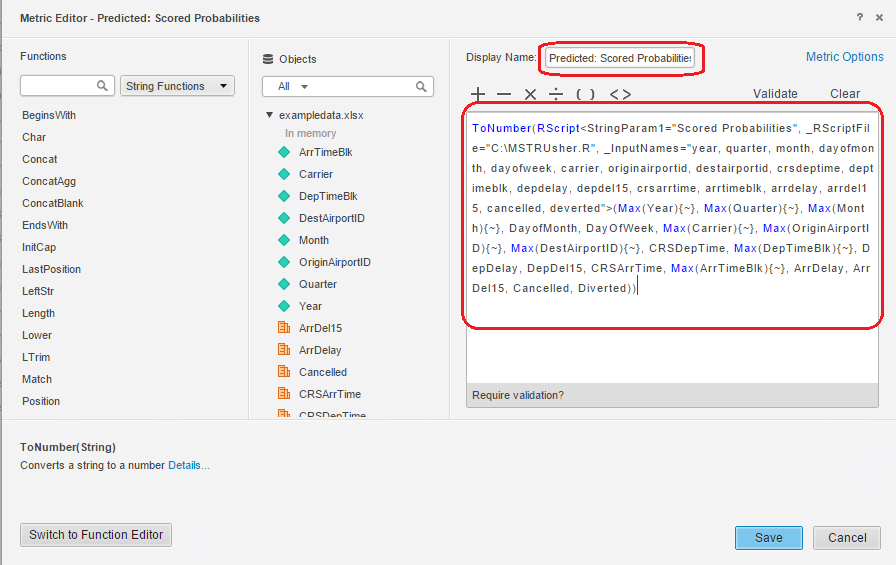
1. In the Metric Editor, copy and paste the metric expression below, and rename the metric as “Predicted: Scored Labels”.

RScript<StringParam1="Scored Labels", \_RScriptFile="C:\MSTRUsher.R", \_InputNames="year, quarter, month, dayofmonth, dayofweek, carrier, originairportid, destairportid, crsdeptime, deptimeblk, depdelay, depdel15, crsarrtime, arrtimeblk, arrdelay, arrdel15, cancelled, deverted">(Max(Year){~}, Max(Quarter){~}, Max(Month){~}, DayofMonth, DayOfWeek, Max(Carrier){~}, Max(OriginAirportID){~}, Max(DestAirportID){~}, CRSDepTime, Max(DepTimeBlk){~}, DepDelay, DepDel15, CRSArrTime, Max(ArrTimeBlk){~}, ArrDelay, ArrDel15, Cancelled, Diverted)

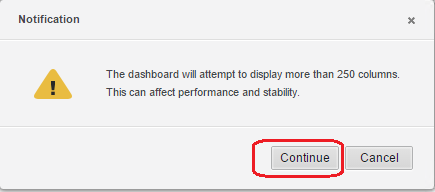


1. Validate the metric formula, and click Save.
2. In the DATASET panel, right click on any metric (e.g. ArrDel15), and select “Create Metric” from the menu to create another metric.
3. In the Metric Editor, copy and paste the metric expression below, and rename the metric as “Predicted: Scored Probabilities”.

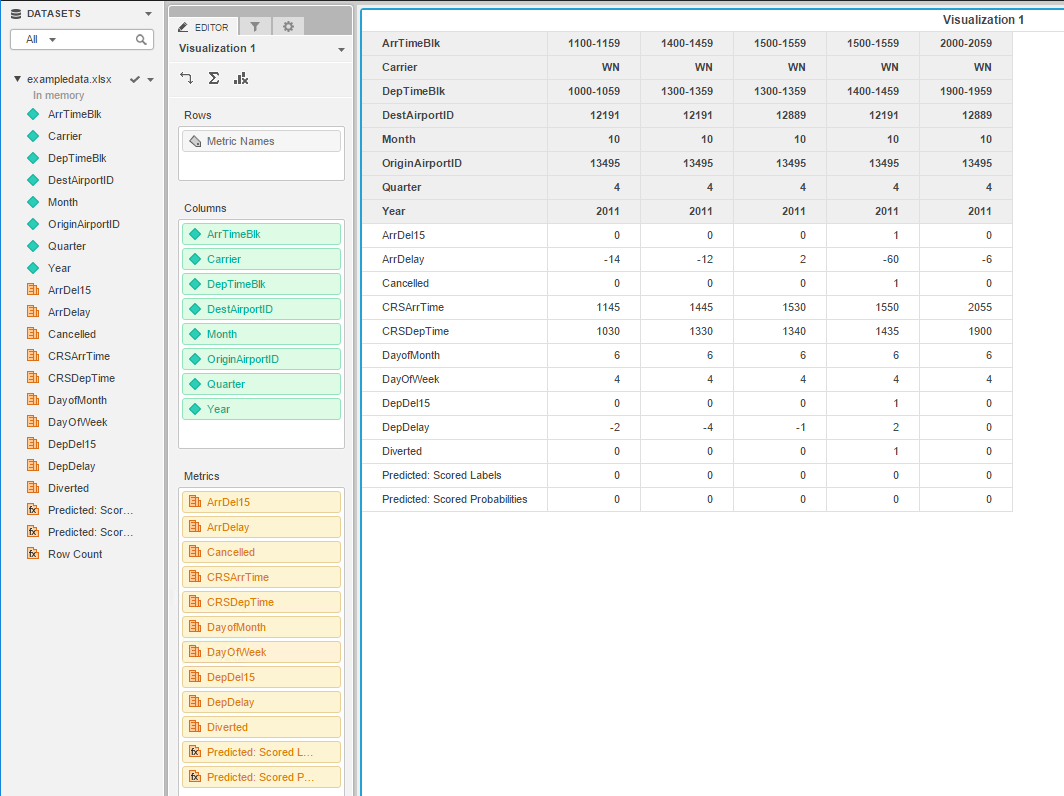
ToNumber(RScript<StringParam1="Scored Probabilities", \_RScriptFile="C:\MSTRUsher.R", \_InputNames="year, quarter, month, dayofmonth, dayofweek, carrier, originairportid, destairportid, crsdeptime, deptimeblk, depdelay, depdel15, crsarrtime, arrtimeblk, arrdelay, arrdel15, cancelled, deverted">(Max(Year){~}, Max(Quarter){~}, Max(Month){~}, DayofMonth, DayOfWeek, Max(Carrier){~}, Max(OriginAirportID){~}, Max(DestAirportID){~}, CRSDepTime, Max(DepTimeBlk){~}, DepDelay, DepDel15, CRSArrTime, Max(ArrTimeBlk){~}, ArrDelay, ArrDel15, Cancelled, Diverted))



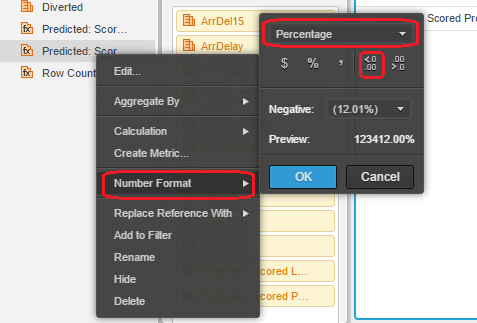
1. Validate the metric formula, and click Save.
2. Drag and drop all the attributes to the “Columns” drop zone (You can click on the first attribute, press Shift key and click on the last attribute to select all of them).
3. Click Continue button when there is a notification.



1. Drag and drop all the metrics except Row Count to the “Metrics” drop zone (You can click on the first metric, press Shift key and click on the last metric to select all of them).
2. Drag and drop “Metric Names” which is currently in the “Columns” drop zone, to the “Rows” drop zone. Now you dashboard looks like following:



1. Change data format for the metric “Predicted: Scored Probabilities”. Right click on this metric in the DATASETS panel, select “Number Format”, and in the number format dialog, choose Percentage, and increase decimals to 2.



1. Click OK.
2. Now your dashboard looks like following. The metric “Predicted: Scored Labels” shows the classification results for your new data, and metric “Predicted: Scored Probabilities” shows the prediction probabilities.

