**CIS 5560 TERM PROJECT TUTORIAL** 

Authors : Hemamalini,Lakshmi,Pallavi

Instructor : Jongwook Woo

Date : 05/18/2017

**LAB TUTORIAL**

Hemamalini Madhanguru([hmadhan@calstatela.edu](mailto:hmadhan@calstatela.edu))

Lakshmi Sundararajan([lsundar@calstatela.edu](mailto:lsundar@calstatela.edu))

Pallavi Attimakula([pattima@calstatela.edu](mailto:pattima@calstatela.edu))

**Machine Learning With Lending Loan Clubs**

**Objectives**

In this hands-on lab, you will learn how to:

* Get data manually using Microsoft Azure ML
* Create experiment using models
* Predict the lending of loans trend by splitting data to training and test data in Azure modules
* Visualization

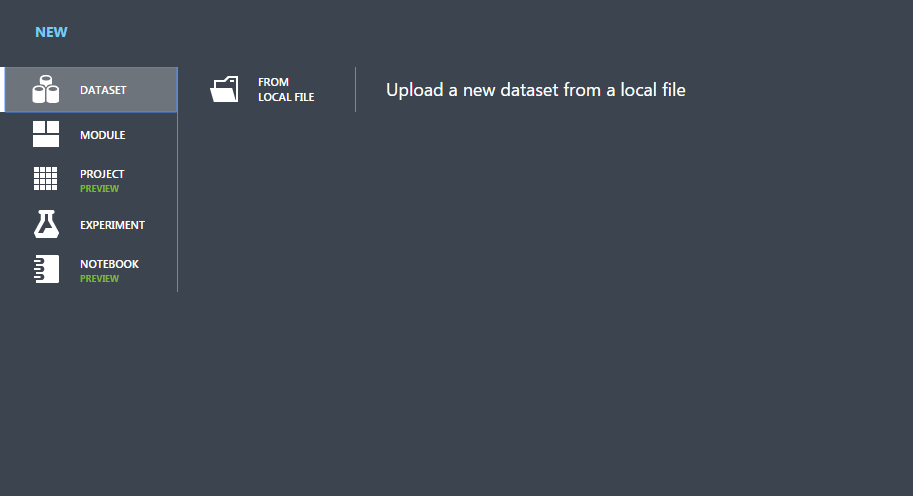
**Platform Specifications**

* IBM Bluemix BigInsights
* CPU Speed: 2.7Ghz
* Number of CPU cores: 4
* Number of nodes: 128
* Total Memory Size: 6GB

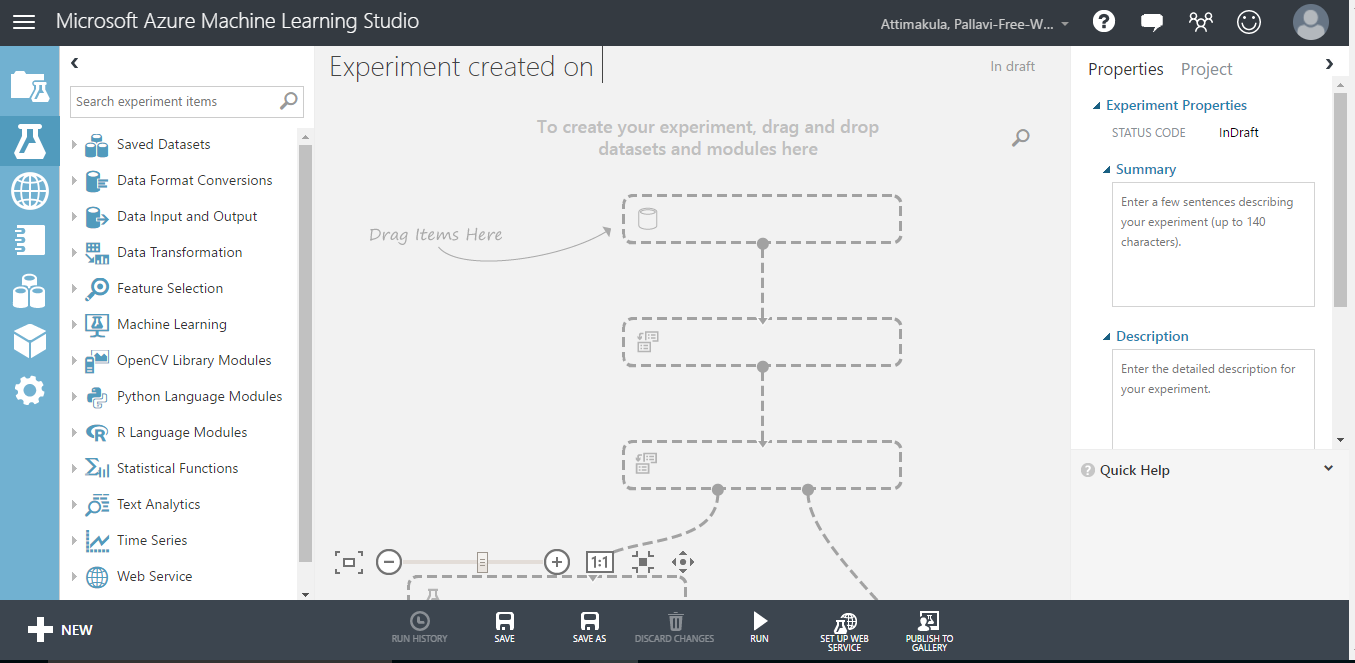
Step 1: Get data manually using Microsoft Azure ML

This step is to get data manually from our local file

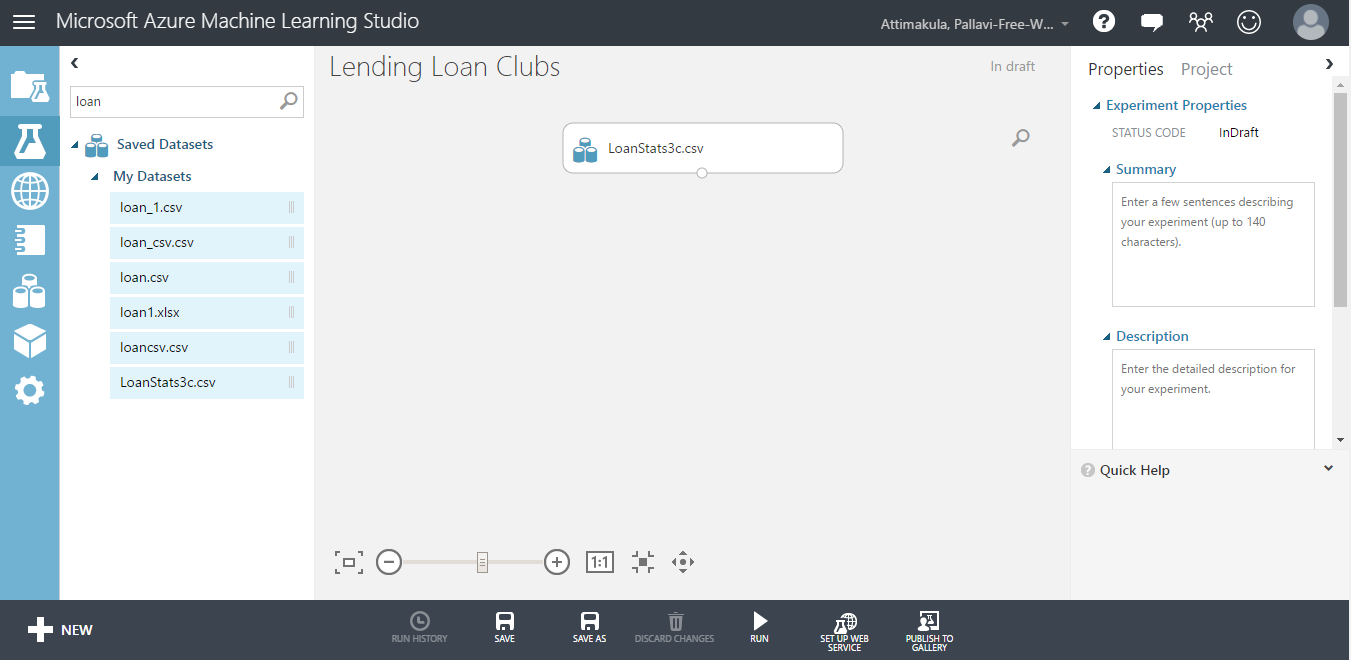
1. Open a browser and browse to <https://studio.azureml.net>
2. Click Sign In and sign in using the Microsoft account associated with your free Azure ML account.
3. Go to New option on the left and click on it
4. Select Dataset option and upload the file from your desktop



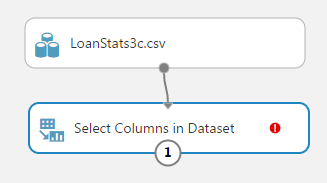
1. After uploading the file select the New option from the left corner and select experiment and blank experiment.
2. You should now be in Azure ML Studio with the Experiments page selected, which looks like the following image



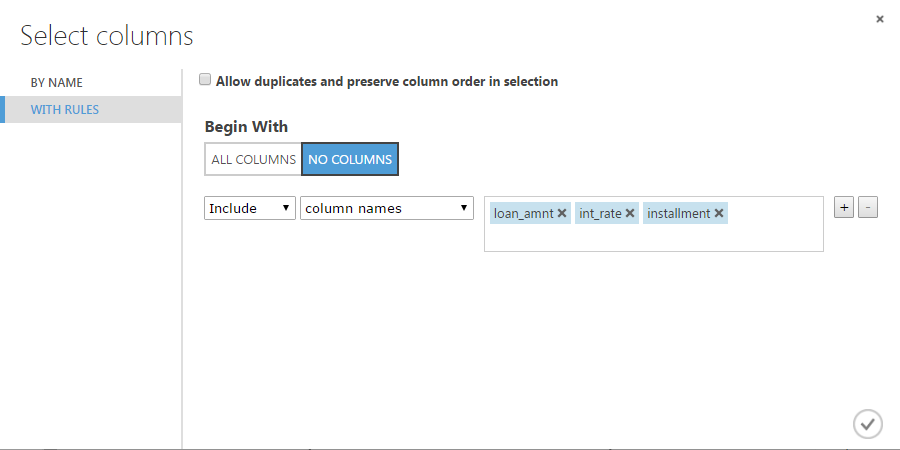
1. Change the title of your experiment from “Experiment created on today’s date” to “Lending Loan Clubs”
2. In the experiment items pane on the left, expand Saved Datasets, expand Samples, and drag LoanStats3c.csv dataset to the experiment canvas in the middle of the page, as shown in the following image.



1. Select the LoanStats3c.csv dataset on the canvas, and note that it has a single output port (indicated as a circle containing the value 1 at the bottom of the dataset icon). Right-click this output port and click Visualize to see the data that the dataset contains.
2. In the search box, type “Project Columns”. Then, in the filtered experiment items pane, under Data Transformation and Manipulation, drag the Project Columns – Select Columns in Data Set- module to the canvas and place it under the LoanStats3c.csv
3. Click the output port of the LoanStats3c.csv dataset, and drag it to the input port at the top of the Project Columns (Select Columns in Data Set module) to connect the items. Your experiment should now look like the following image.

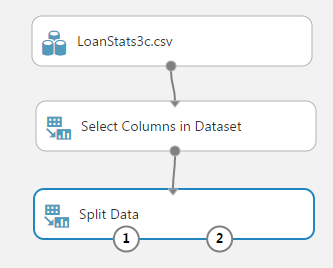


1. Select the Project Columns module, and in the Properties pane on the right, click Launch column selector. The column selector is a common user interface element in Azure ML modules, and enables you to select the columns you want to use in the module. In this case, the Project Columns module is used to filter out columns you don’t need, so that only the columns you want to use are passed into the data flow for the next module.
2. In the Select columns dialog box, select option With Rules to begin with no columns, and include the loan\_amnt, int\_rate and installment column names as shown in the image below. Then click the OK icon to close the column selector.



Step 2: Train NLP

1. In the experiment items pane, search for “Split”. Then drag the Split module (Split Data) that is found in the Data Transformation category to the canvas and place it under the Select Columns in Dataset module, and connect it to the end of the workflow as shown here:



You will use the Split module to split the data into two sets; one to train the model, and another to test it.

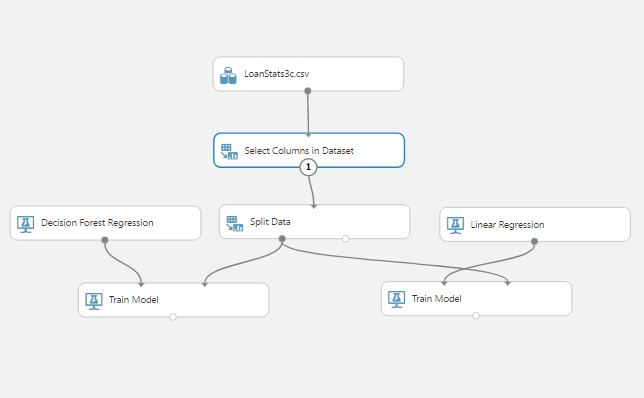
1. Select the Split module, and in the Properties pane, set the following properties:

Splitting mode: Split Rows

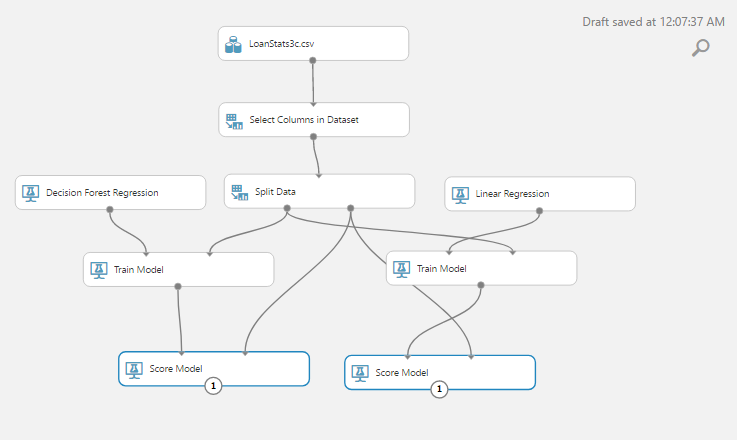
* + - Fraction of rows in the first output dataset: 0.7
    - Randomized: Selected
    - Random seed: 0
    - Stratified split: False

1. Search for the Linear Regression module, and drag it to the canvas beside the Split module. Then select the Linear Regression module and in the Properties pane, set the following properties: Solution method: Ordinary Least Squares
   * + L2 regularization weight: 0.001
     + Include intercept term: Selected
     + Random number seed: 345689
     + Allow unknown category levels: Selected

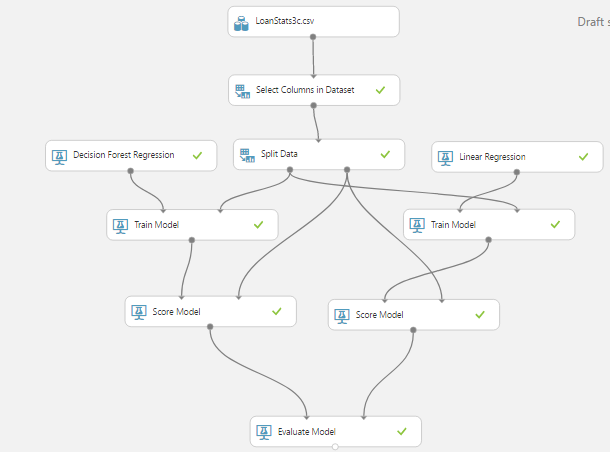
1. Search for the Decision Forest Regression module. Make sure you have selected the regression model version of this algorithm. Drag this module onto the canvas. Set the properties of this module as follows:
   * + Resampling method: Bagging
     + Create trainer mode: Single Parameter
     + Number of decision trees: 8
     + Maximum depth of the decision trees: 32
     + Number of random splits per node: 128
     + Minimum number of samples per leaf node: 1
     + Allow unknown values for categorized features: Checked
2. Search for the Train Model module, and drag it to the canvas beneath the Split and Linear Regression modules. Then connect the output from the Linear Regression model to the left input of the Train Model module, and connect the left output of the Split module (which represents the training data set) to the right input of the Train Model module.
3. Again search for the Train Model module. Drag this module onto the canvas beside the old Train Model
4. Connect the Untrained Model output port of the Decision Forest Regression module to the Untrained Model input port of the Train Model module.
5. Connect the Results dataset1 (left) output port of the Split module to the Dataset input port of the Train model module. It should look like the following image :



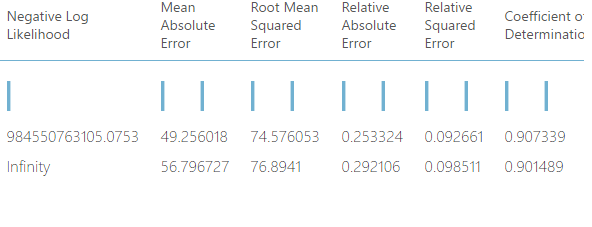
1. Search for the Score Model module, and drag it to the canvas below the first Train Model module. Then connect the output from first Train Model module to the left input of the Score Model module, and connect the right output from the Split module to the right input of the Score Model module .
2. Again search for the Score Model module, and drag it to the canvas below the second Train Model module. Then connect the output from second Train Model module to the left input of the Score Model module, and connect the right output from the Split module to the right input of the Score Model module; as shown in the following image:



1. In the experiment items pane, search for “Evaluate Model”, and drag the Evaluate Model module to the canvas below the Score Model module. Then connect the output from the first Score Model module to the first (left-most) input of the Evaluate Model module and connect the output from the second Score Model to the right input of the Evaluate Model as shown here:



1. Save and run the experiment. Then, when the experiment has finished, visualize the Evaluation Results dataset output of the Evaluate Model module, which should look like this:



The values shown are measurements of the accuracy of the model when comparing the label values that it predicts to the known values in the test dataset. For example, in the case of this regression model, the Relative Squared Error value indicates how well the model explains variance in the predicted label value and the known label value, with a lower number indicating a better predictive result. The specific measurements vary by model type, and you can connect two scored models to the Evaluate Model module to compare their effectiveness.

1. Close the evaluation results dataset.

References

1. Data Set URL - <https://www.kaggle.com/wendykan/lending-club-loan-data>

2. GitHub URL - <https://github.com/lsundar1203/PROJECT-5560>

3. References :

* Microsoft's DAT203x, Data Science and Machine Learning Essentials
* “Big Data Trend and Open Data”, Jongwook Woo, Aug 12 2016, UKC 2016, Dallas, Texas
* How to choose algorithms for Microsoft Azure Machine Learning, https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice