Microsoft Azure Machine learning tutorial

Topic: Predicting Ratings/stars of business given in yelp dataset.

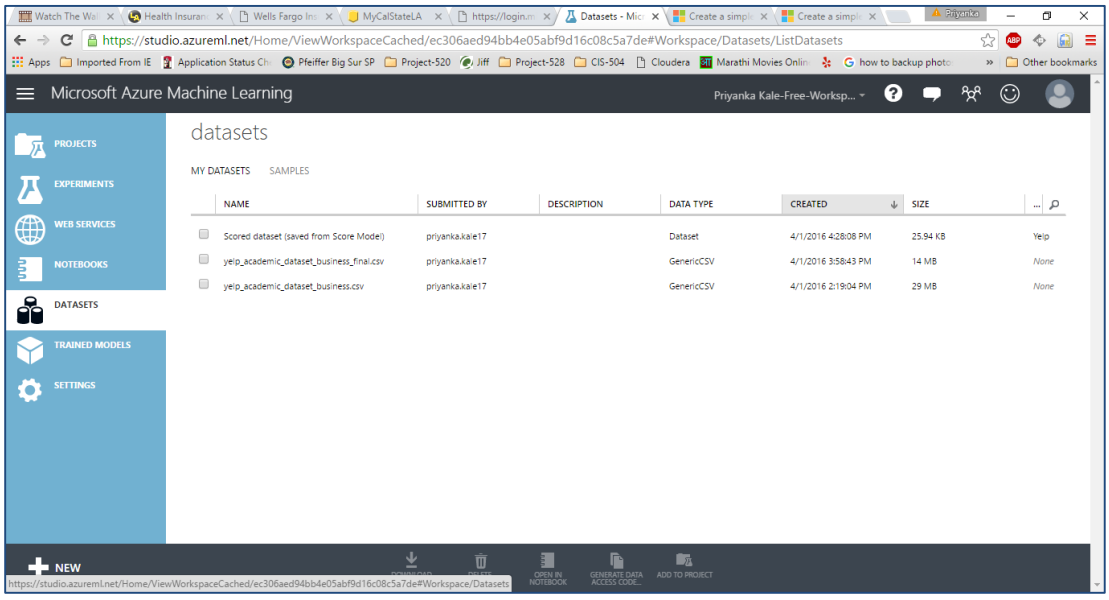
Here I have created a multiclass logistic regression model that predicts the ratings of a business based on different variables such as review count, name and business ID. To do this, I have used Azure Machine Learning Studio to develop and iterate on a simple predictive analytics experiment.

Four steps to create an experiment using ML studio:

1. Create a model
    - Step 1: Upload the data
    - Step 2: Preprocess and clean data
2. Train the model
    - Step 3: Choose and apply a learning algorithm from the set.
3. Score and test the model
    - Step 4: Predict ratings

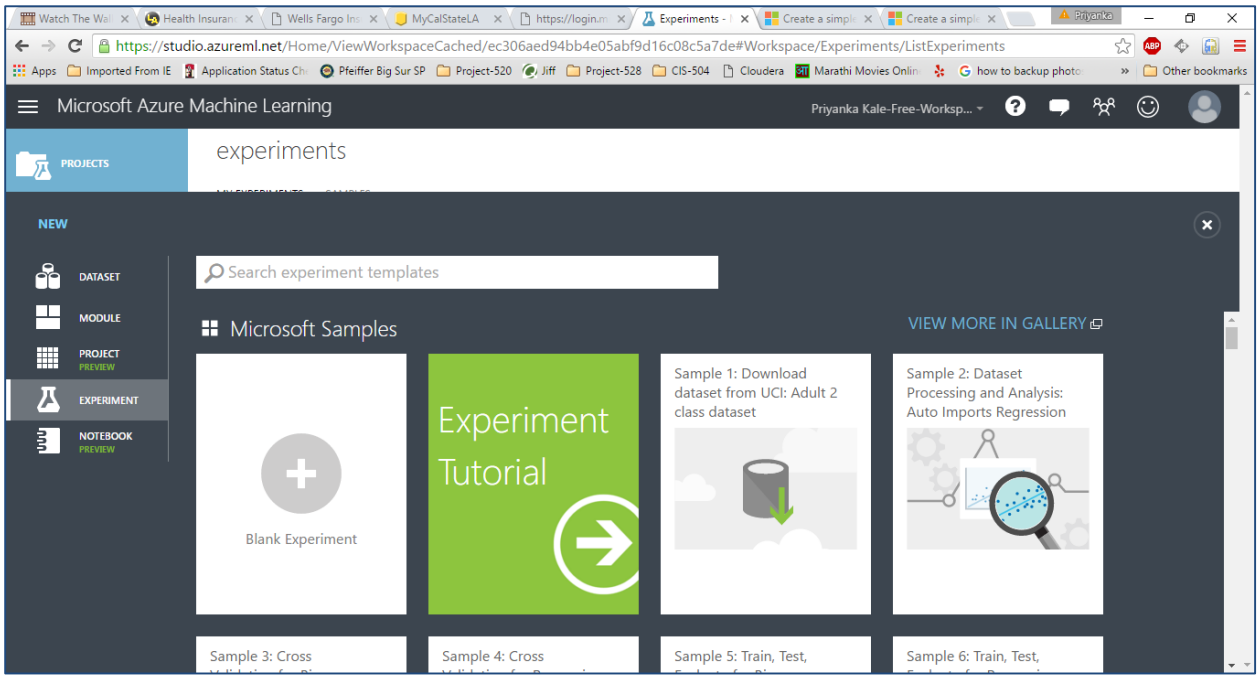
Step 1: Upload the data

For this experiment I am using dataset downloaded from Yelp dataset source given at https://www.yelp.com/dataset_challenge . This dataset includes entries like business ID, Name, Location, Ratings, Review Count, Category of the business etc. Go to Dataset tab on left hand side, click on **+ New** sign and upload the dataset file. When you click on my datasets then the dataset you have uploaded will appear.



1. Start a new experiment by clicking +NEW at the bottom of the Machine Learning Studio window, select EXPERIMENT, and then select Blank Experiment. Select the default

experiment name at the top of the canvas and rename it to something meaningful, for example, **Business Ratings prediction** or **Yelp dataset analysis** etc.



2. To the left of the experiment canvas is a palette of datasets and modules. Search for the dataset you want to use for the experiment.
3. Drag the dataset to the experiment canvas. In this case, upload Yelp dataset which we have uploaded earlier in my datasets.
4. To see what this data looks like, click the output port at the bottom of the Yelp dataset, and then select **Visualize**.

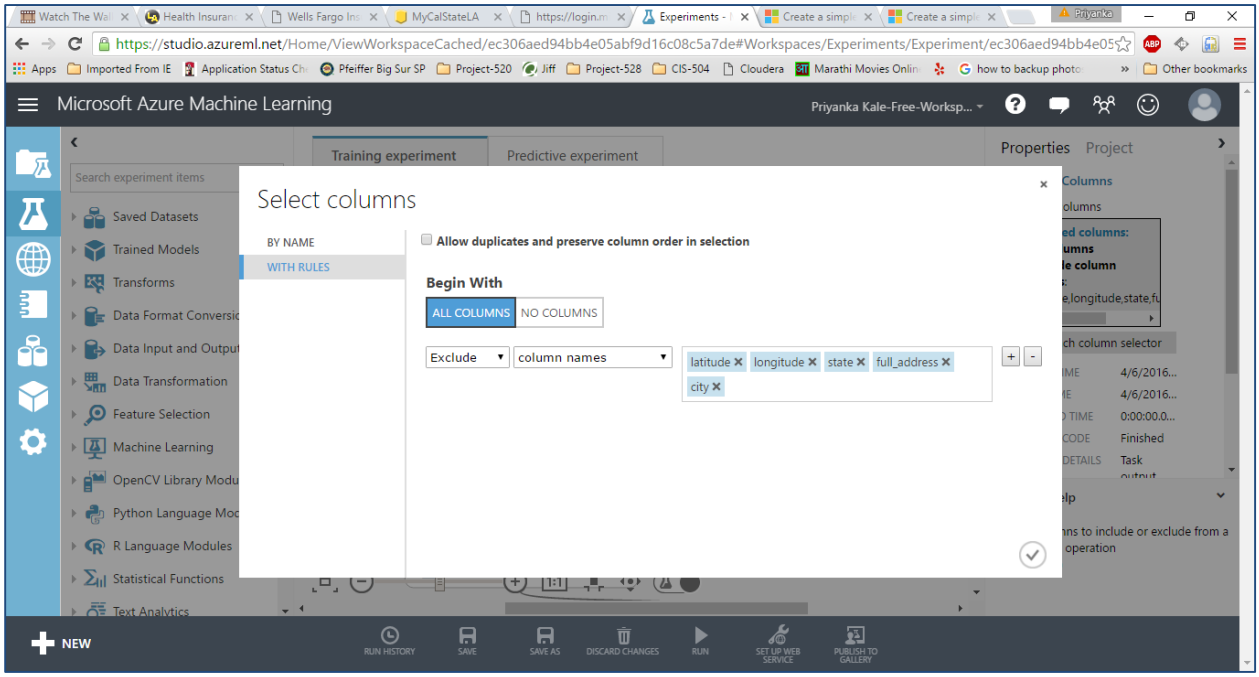
Step 2: Preprocess raw data

A dataset usually requires some preprocessing before it can be analyzed. You might have noticed the missing values present in the columns of various rows. These missing values need to be cleaned so the model can analyze the data correctly. In our case, we'll remove any rows that have missing values.
First we'll remove the normalized-losses column, and then we'll remove any row that has missing data.

1. Search for project columns in the Search box at the top of the module palette to find the Project Columns module, then drag it to the experiment canvas and connect it to the output port of the Yelp (Raw) dataset. This module allows us to select which columns of data we want to include or exclude in the model.

2. Select the Project Columns module and click Launch column selector in the Properties pane.

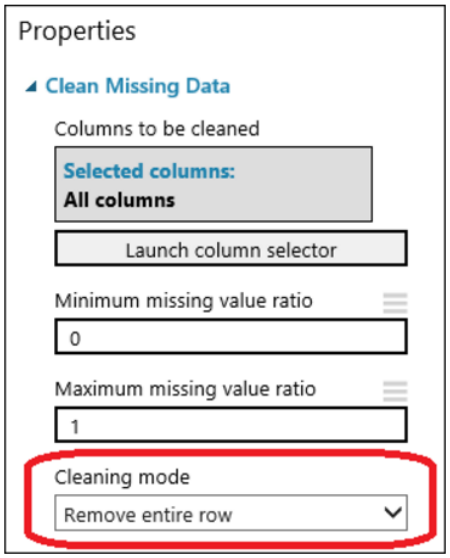
3. Make sure all columns is selected in the filter drop-down list, Begin With. This directs Project Columns to pass through all the columns (except those we're about to exclude).



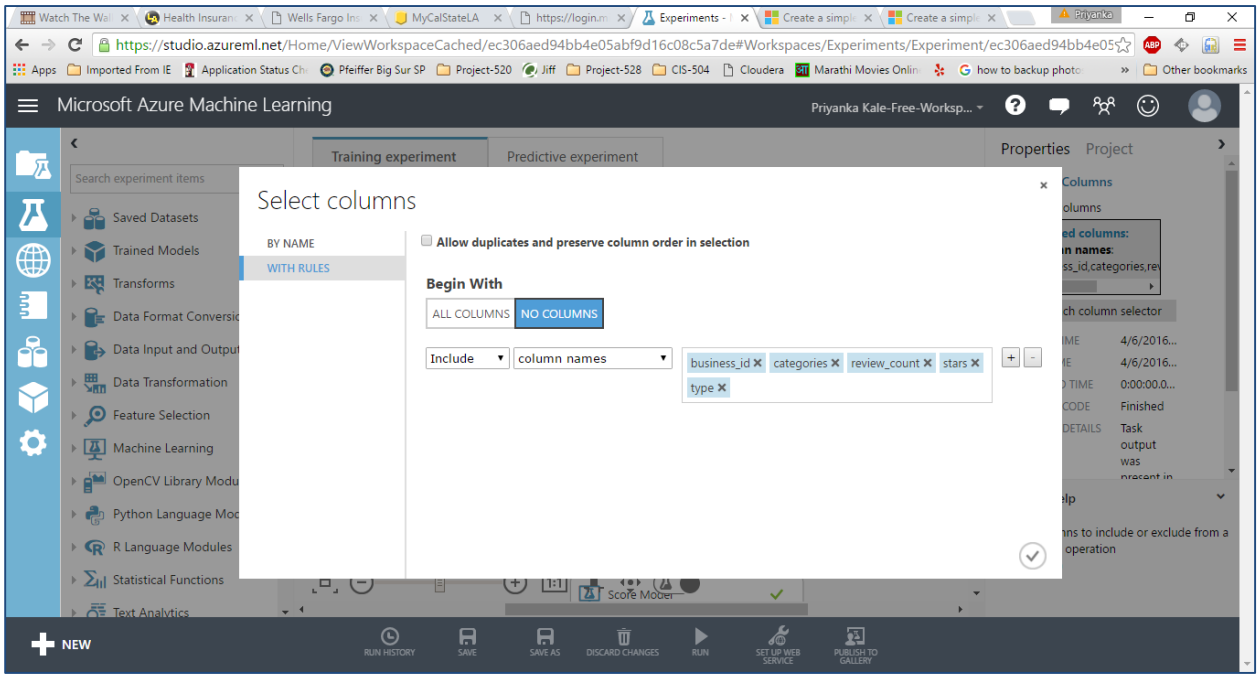
4. In the next row, select Exclude and column names, and then click inside the text box. A list of columns is displayed. Select normalized-losses, and it will be added to the text box.

5. Click the check mark (OK) button to close the column selector. The properties pane for Project Columns shows that it will pass through all columns from the dataset except normalized-losses.

6. Now search for the module named clean missing data. Drag that to the experiment. Set the Properties pane. Fill minimum missing value as 0 and maximum missing value as 1.



7. Again take one more Project column. Add only those columns this time which needs to get processed by the algorithm. We can add only those columns which are necessary for processing rest all can be omitted. Here I am taking data only for Arizona and Nevada State. Here I am keeping Businees ID, Name, Review count, Stars, Type of the Business.



Now that the data is clean, we're ready to specify what features we're going to use in the predictive model.

Step 3: Choose and apply a machine learning algorithm:

Now that the data is ready, constructing a predictive model consists of training and testing. I am using data to train the model and then test the model to see how close it's able to predict ratings.

Classification and ***Regression*** are 2 types of supervised machine learning techniques. We want to predict the rating of the business, which can be any value starting 1 to 5, so we'll use a regression model.

Our data for both training and testing by splitting it into separate training and testing sets. Select and drag the Split Data module to the experiment canvas and connect it to the output of the last Project Columns module. Set **Fraction of rows in the first output dataset** to 0.75. This way, we'll use 75 percent of the data to train the model, and hold back 25 percent for testing.

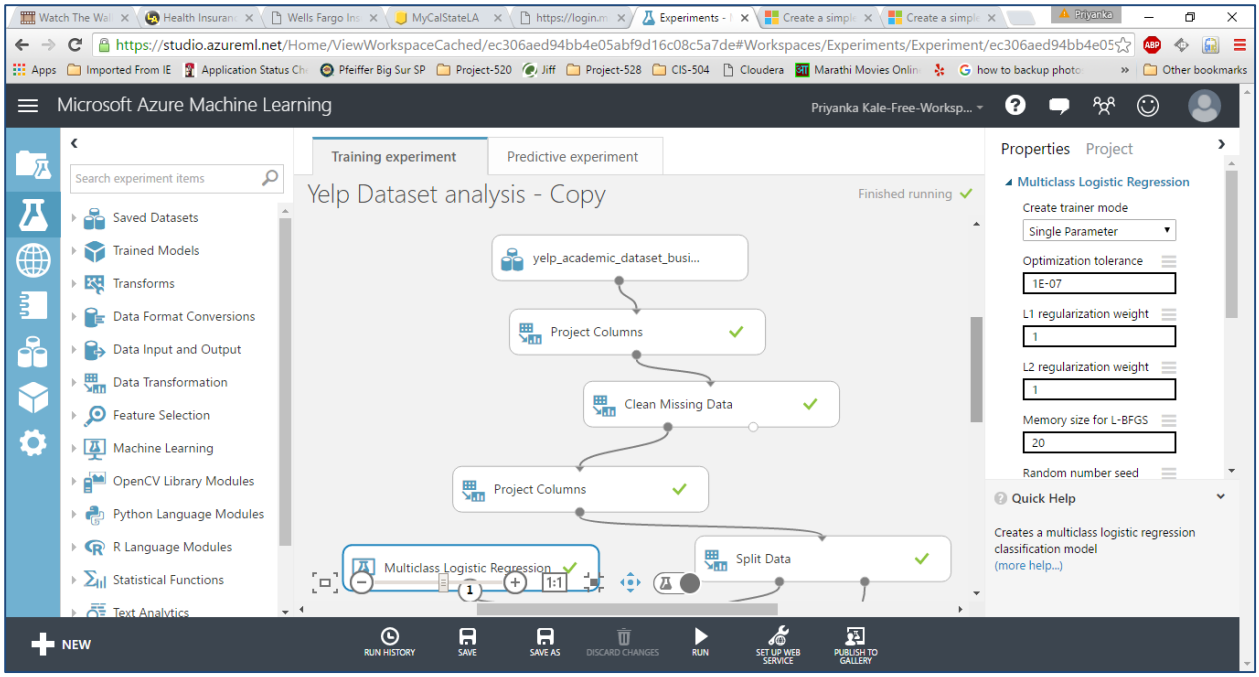
1. Run the experiment. This allows the Project Columns and Split Data modules to pass column definitions to the modules we'll be adding next.

2. To select the learning algorithm, expand the Machine Learning category in the module palette to the left of the canvas, and then expand Initialize Model. This displays several categories of modules that can be used to initialize machine learning algorithms. For this experiment, select the multiclass Logistic Regression module under the Regression category and drag it to the experiment canvas.

3. Find and drag the Train Model module to the experiment canvas. Connect the left input port to the output of the Multiclass Logistic Regression module. Connect the right input port to the training data output (left port) of the Split Data module.

4. Select the Train Model module, click Launch column selector in the Properties pane, and then select the Stars column. This is the value that our model is going to predict.

5. Run the experiment.



Step 4: Predict new business ratings:

6. Now that we've trained the model using 75 percent of our data, we can use it to score the other 25 percent of the data to see how well our model functions.

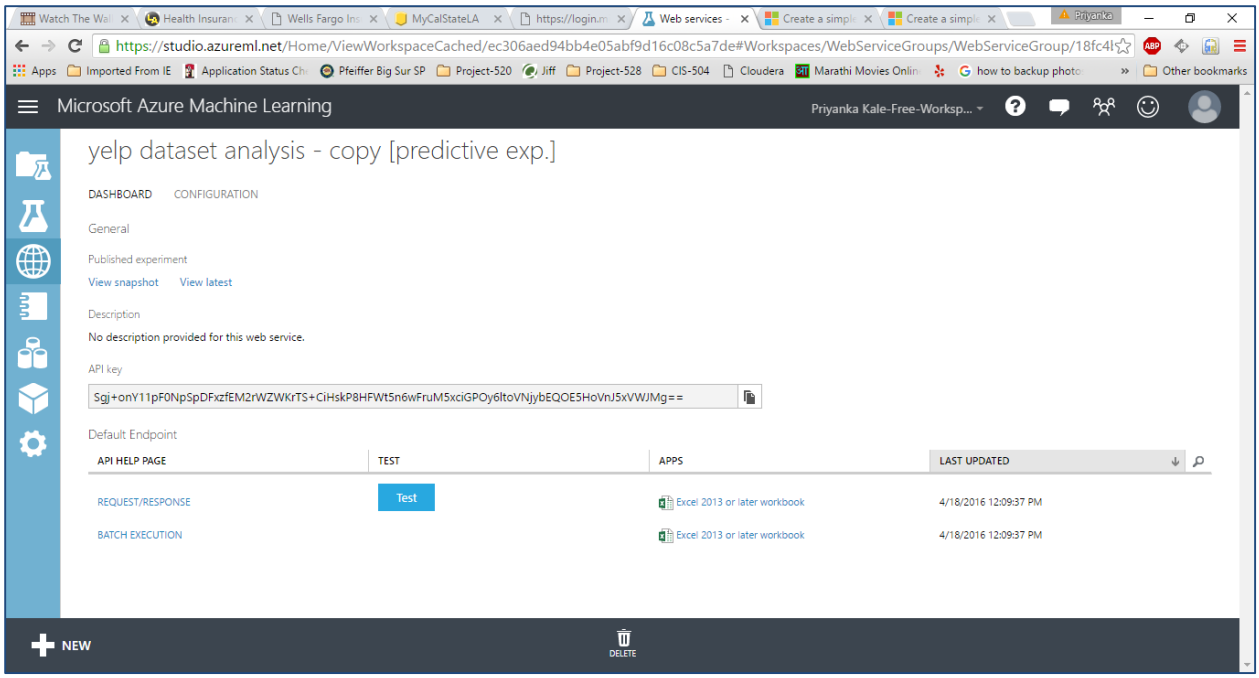
7. Find and drag the **Score Model** module to the experiment canvas and connect the left input port to the output of the Train Model module. Connect the right input port to the test data output (right port) of the Split Data module.

8. To run the experiment and view the output from the Score Model module, click the output port, and then select Visualize. The output shows the predicted values for ratings and the known values from the test data.

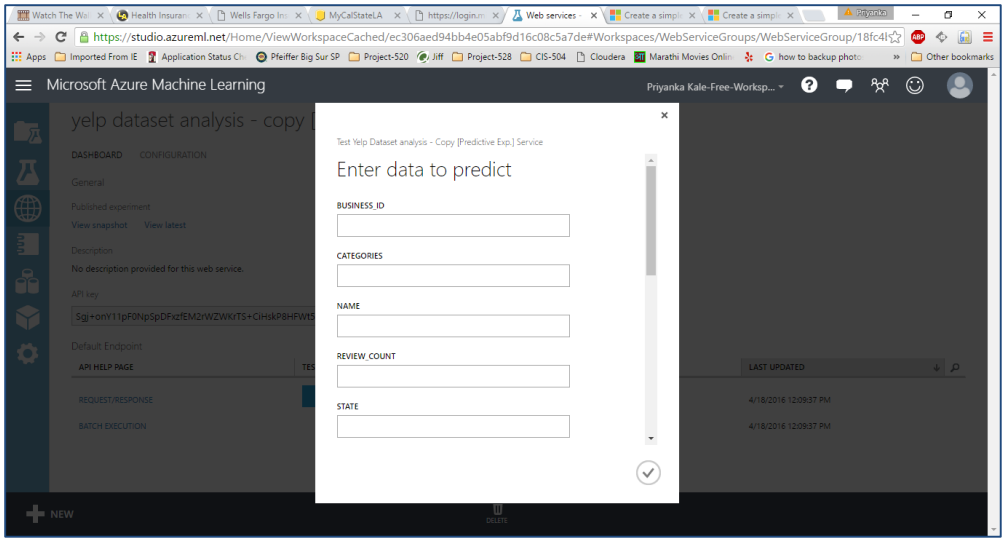
9. Finally, to test the quality of the results, select and drag the Evaluate Model module to the experiment canvas, and connect the left input port to the output of the Score Model module.

10. Run the experiment.

11. Once the experiment is run successfully, click on the set up web service and select 1st option. This will create predictive Experiment. Now you can run the experiment. Once finished successfully, click on the deploy web service and test the results.

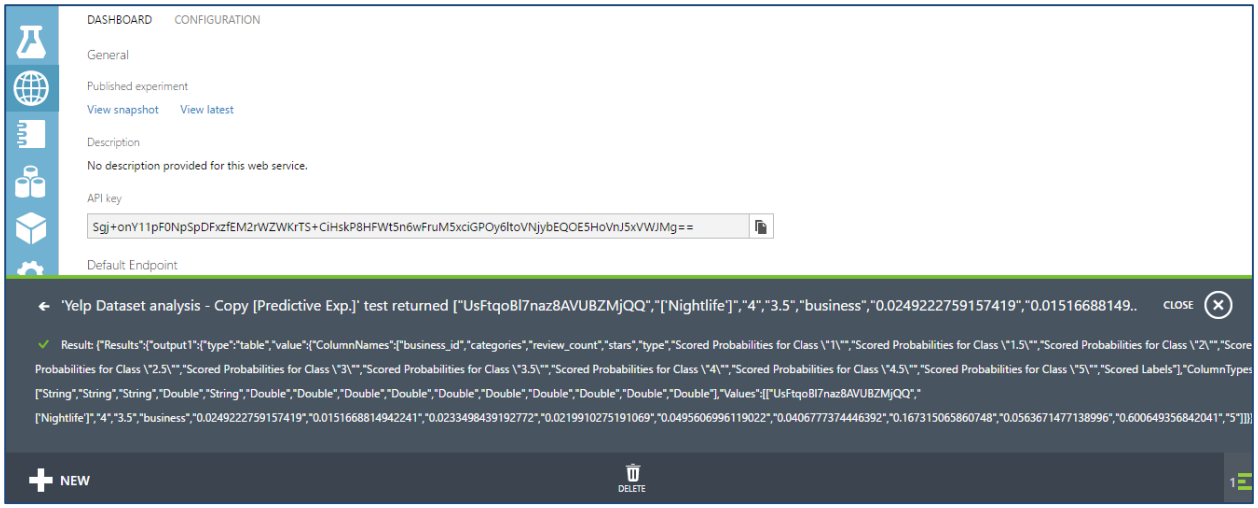


Click on Test. Fill the values in the pop up window and click ok.



Results:

The results will be in the form of probabilities. The highest probabilities will be considered as the closest one and the prediction is done.



It will give probabilities for all the ratings starting 1 to 5 and then nearest probability will be taken into consideration.

Conclusion:

Thus Multiclass Logistic regression is used to predict the value of ratings/Stars a business will receive depending on the parameters given in the training model. In multiclass logistic regression, the classifier can be used to predict multiple outcomes. Here in our case it gives you probabilities for stars/ratings.