**Assignment 2**

**Advance in Data Sciences and Architecture**

**INFO 7390 - SPRING 2017**

PROFESSOR:

**SRIKANTH KRISHNAMURTHY**

TEAM MEMBERS:

**HARSHAD PARDESHI (**[**www.linkedin.com/in/harshad-pardeshi/**](file:///C:\Data\Important\Study\ADS\Presentations\Kubernete\kubernets_datascience-master\kubernets_datascience-master\www.linkedin.com\in\aashritandon\)**)**

**PRITHVI KAMATH (**[**www.linkedin.com/in/**](http://www.linkedin.com/in/)**prithvipkamath/)**

**TEJESH RATHI (www.linkedin.com/in/tejesh-rathi/)**

# **Problem Statement**

You are working at a bank and you are considering investing in Lending club. Since there are no standard models, you are expected to build prediction models that will help you predict the interest rates based on various parameters users would input.

Your first challenge is to programmatically download the data from https://www.lendingclub.com/info/download-data.action

Your goal is to download the data programmatically from the website and create one dataset for the entire database and perform the following tasks:

* Data download: How will you download all loan data and create one dataset
* Missing data analysis: How will you handle missing data?
* Feature engineering: What variables do you need to predict interest rates? Ensure users would be able to give you that information to help you predict rates
* Pipeline: Using Luigi/Pinball/Airflow automate the above 3 steps.
* You need to create one more pipeline to do this for the “Declined Loan data”. Repeat above steps

You should dockerize the whole project (excluding the Power BI dashboard) and write clear instructions on how to run the docker image. Research how you would schedule this on Amazon/Azure/BlueMix so that this image would run and execute the Luigi/Pinball/Airflow pipeline. After running this pipeline, the clean pre-processed data should be stored on S3/Object Storage/Blob storage(AWS/BlueMix/Azure)

* Write a Jupyter notebook using R/Python to graphically represent different summaries of data. Summarize your findings in this notebook.
* Summarize your key insights about different user profiles, states, loan amounts etc.
* Create a Data scientist view of Power BI dashboards to illustrate your key insights

Your next goal is to build a model to predict interest rates. You will get leads from people with different profiles and you must decide if you will give loans or not and if you will give a loan, how much interest you would charge for those loans.

**Classification**

Use the “Loan Data” and the “Declined Loan Data” datasets to build classification models that will generate a flag whether to give a loan or not.

Start with logistic regression using Jupyter and Python/R

Compute ROC curve and Confusion matrices for training and testing datasets and comment on the results.

Repeat this using Random Forest, Neural Network models and SVN algorithms.

Choose one model you will deploy and implement this model on the Microsoft azure machine learning studio and create a REST API

You should be able to a new record (You can define what features you will use) and the result will be a flag whether you would give a loan or not.

**Clustering**

Once you have decided to give a loan, you should build models to decide what interest rate to give. You are debating whether to create one model for all customer prospects or segment data into clusters and then build prediction models specific to each cluster. You think of creating segments or clusters and build models one for each cluster. Your brainstorm with your team and come up with 3 possibilities

1. Segment data into clusters (you define how many) manually using categorical or numerical features. For example, you can segment by state, by ownership of home, by average dti or a combination of features.

2. You use a clustering algorithm (that can factor both numerical and categorical variables) and segment data into k clusters. You will then build prediction models for each cluster.

3. No clusters; Just use data as is

Once you do the clustering use t-sne to visualize your clusters for some sample test data. See http://distill.pub/2016/misread-tsne/ for guidance on using t-sne

**Prediction**

Write a prediction script in a Jupyter notebook in R/Python that builds a Regression model for the interest rate using data from the 3 clustering methodologies you worked.

Try variable selection and build the best model for each segment/cluster (Note: You may have many segments and each model may have different coefficients based on the clusters used to train. You should automate it. Try parallel computing libraries to make things go faster)

o Compute MAE, RMS, MAPE for training and testing datasets

o Repeat this using Random Forest, Neural Network models and KNN algorithms.

o Choose the best model amongst the 4 types of algorithms.

o Deploy the best algorithm/algorithms on Azure ML studio

o You will have a bunch of Rest APIs you should be able to choose from based on the cluster the record belongs to

**Deployment**

Design the following workflow:

Given a record, use a pre-trained clustering model to cluster the record to a segment.

You will have 3 cluster assignments (1-manual, 2-based on your clustering algorithm, 3-default 1- cluster for all data)

For each cluster, there should be a RestAPI which is linked to a chosen prediction model. Look up that API and use it to predict the 3 distinct interest rates.

Select the highest interest rate and return it as your prescribed interest rate.

**Part 2: Building and Evaluating Models**

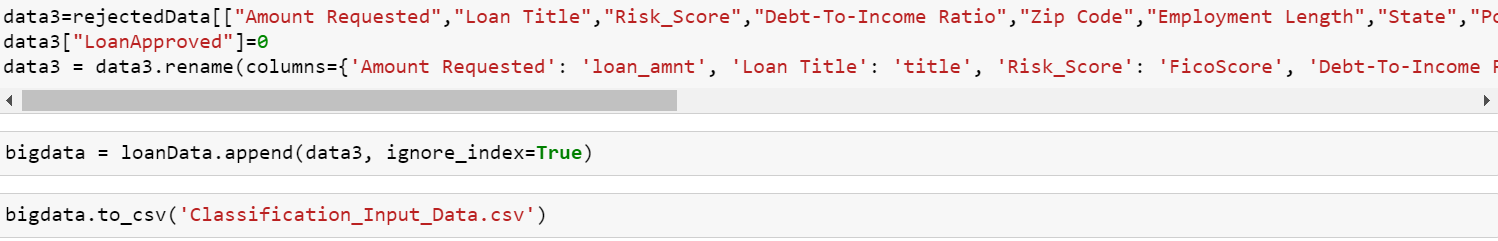
**Classification**

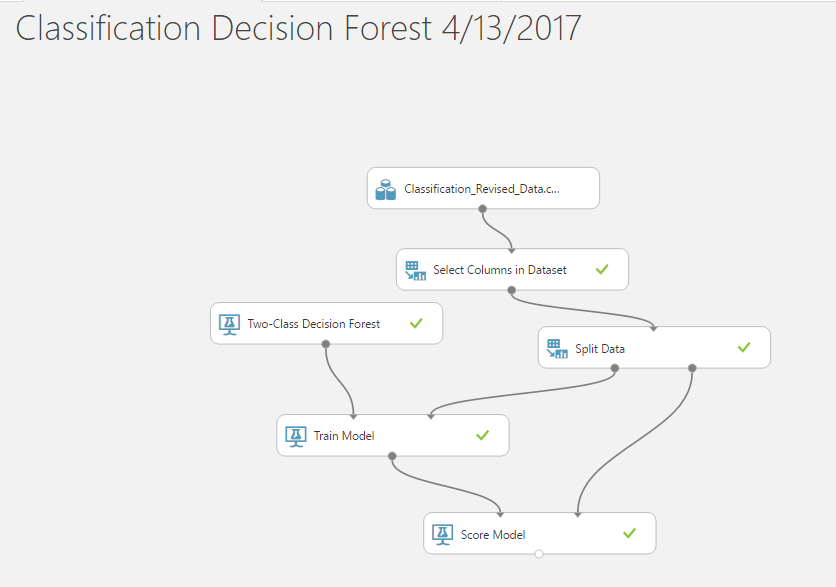
Our first step in classification is to obtain a dataset which has accepted and rejected observations.

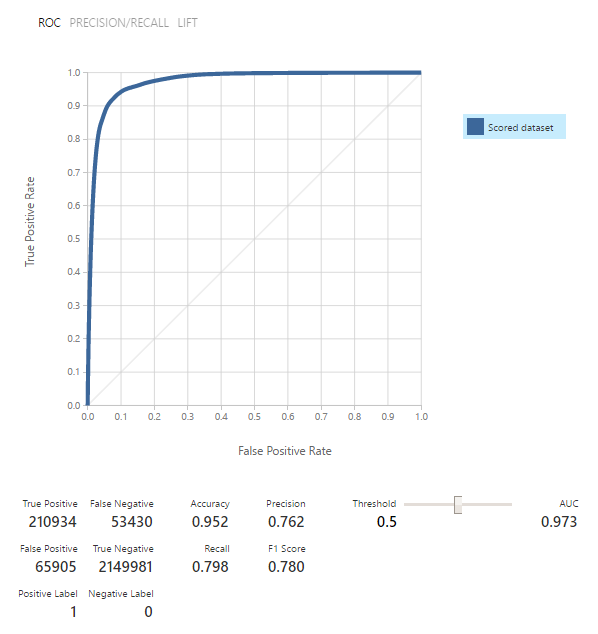
To get this combined data we are concatenating loan accepted data with the rejected data based on common columns in both the files. The selected columns are as follows:

|  |  |
| --- | --- |
| **Accepted Loan Data** | **Rejected Loan Data** |
| Loan\_amnt | Amount Requested |
| title | Loan Title |
| FicoScore (derived attribute) | Risk\_Score |
| dti | Debt-To-Income Ratio |
| zip\_code | Zip Code |
| emp\_length | Employment Length |
| addr\_state | State |
| policy\_code | Policy Code |
| LoanApproved | LoanApproved (derived attribute) |

Below is the code snippet for the same.







Based on the combined file of rejected and accepted records, we trained our model based on few variables, whether to give loan to a person or not. We took the common columns between both the rejected and accepted files, combined it together and moved the data with policy code 2 into the rejected file, as they will not be getting the loan henceforth.

Among the 8 columns, we figured out few columns which are most relevant and gives us higher accuracy of output based on backward selection. We then trained those 8 columns and predicted whether a person should be given loan or not.

We implemented various classification algorithms like Logistic Regression, Random Forest, KNN, Neural Network using iPython Notebook.

Among these, we figured out the best model based on the Accuracy rate we obtained.

Random Forest – 95.2%

SVN – 92.8 %

Logistic Regression – 92.8%

Neural Network – 94.9%

As per the accuracy rates, Random Forest has the highest accuracy and hence that is our best model for Classification. We then implemented Random Forest on Microsoft Azure Machine Learning and the output is as shown in the above image.

**Clustering**

In this, we finalized 13 variables to train the data. The 13 features of our model are as below:

sub\_grade,

derived\_int\_rate,

annual\_inc,

Derived\_term,

FICO,

derived\_emp\_length,

addr\_state,

loan\_amnt,

verification\_status,

home\_ownership,

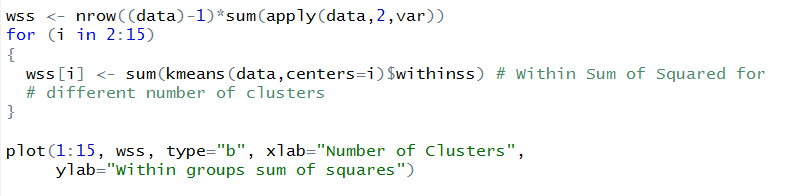
derived\_mths\_since\_last\_delinq,

issue\_year,

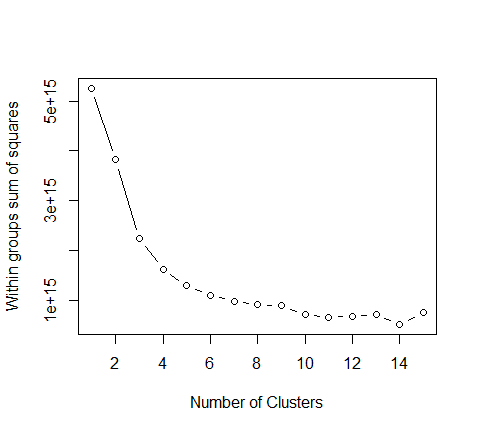
application\_type

**Algorithm Clustering**

We have implemented Bend Graph to get the number of clusters using the below code snippet.



Our bend curve is as follows:

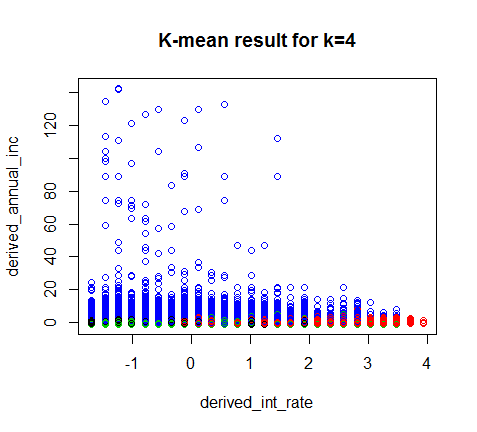


From this Bend Graph, we understand that the line seems to be flat after 4. Hence we chose 4 clusters to implement our clustering algorithms.

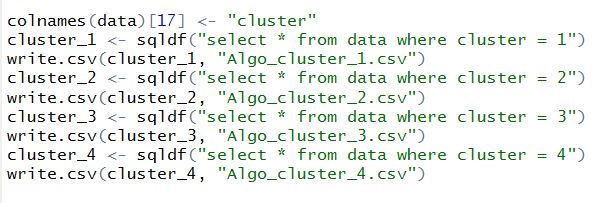
The Clustering algorithm we have chosen is KMeans. To implement KMeans algorithm we have converted the selected columns to integer format and scaled them.

Then we have plotted a scatter plot to show the distribution across each cluster.

The scatter plot is as follows:

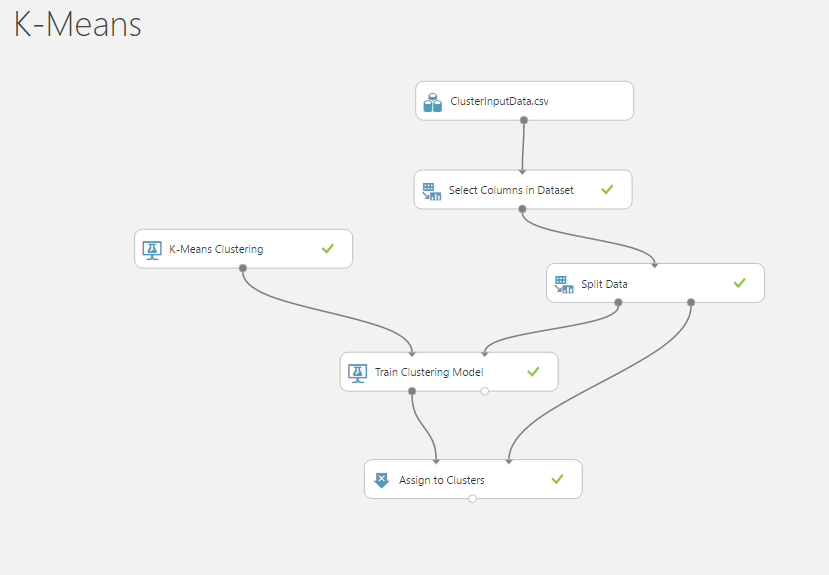


Once we get the scatter plot, we use the cluster information and append it for each observation in the input dataframe and create files for each cluster. These files are cluster specific and are used as input to the prediction algorithms that run on each cluster.



The following prediction algorithms are implemented on each cluster:

* Linear Regression
* Random Forest Regression
* Neural Network



K-Means is a clustering method where in the data is divided into k number of clusters, in our case it is 4 clusters.

The above model shows the implementation of k-means in Microsoft Azure Machine Learning. This divides the data into 4 clusters, based on the user input in the Web UI created.

We have selected Neural Network Regression to be our best prediction model.

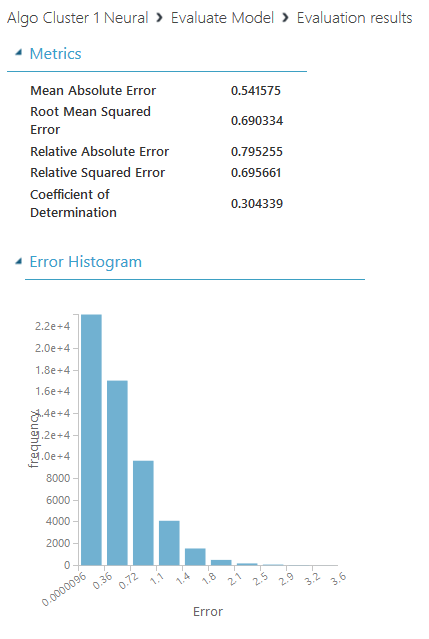
****

Figure : Neural Network results on Cluster 1.

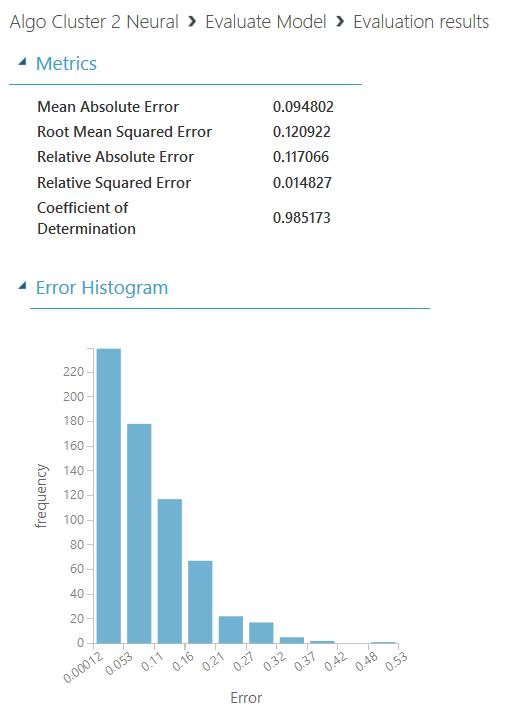


Figure : Neural Network result on cluster 2

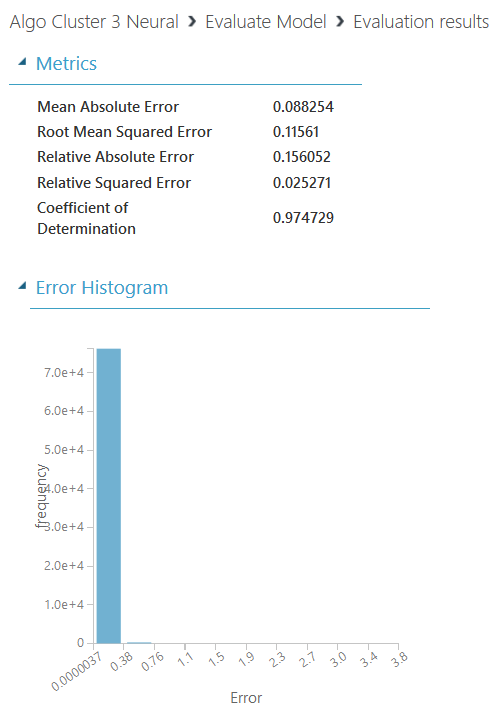


Figure : Neural Network results on cluster 3

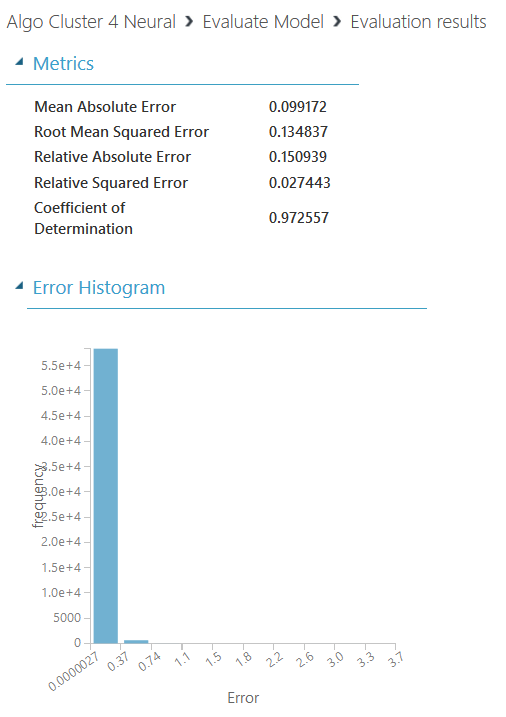


Figure : Neural Network result on cluster 4.

**Manual Clustering**

We are dividing the entire dataset into 3 clusters while implementing manual clustering based on the below factors:

* State
* FICO score
* Employment Length
* Term

For each of the above factors, we have grouped each observation into 3 groups and each group has been assigned a specific weight. Each observation is later assigned a value based on the range of these weights.

Observations with weights 0-5 form the first cluster.

Observations with weights 6-9 form the second cluster.

Observations with weights 10-12 form the third cluster.

Below is the explanation on the distribution of weights for each observation.

**State Distribution:**

States have been divided into 3 groups based on cash solvency distribution.



|  |  |
| --- | --- |
| **State** | **Weights** |
| State1 | 1 |
| State2 | 2 |
| State3 | 3 |

**FICO score:**

Observations with FICO score in the range 300-449, 450-649 and 650-850 have been divided into 3 groups

|  |  |
| --- | --- |
| **FICO** | **Weights** |
| 300-449 | 1 |
| 450-649 | 2 |
| 650-850 | 3 |

**Employment Length**:

Observations with Employment length 0-3, 4-7 and 8-10 have been grouped together into 3 groups.

|  |  |
| --- | --- |
| **Employment Length** | **Weights** |
| 0-3 | 1 |
| 4-7 | 2 |
| 8-10 | 3 |

**Loan Term:**

We only have 2 values for loan term, but we have divided them into 3 groups.

Observations with loan term as 36 months are clubbed together.

Observation with 60 months are further divided into 2 groups based on FICO score.

|  |  |
| --- | --- |
| **Loan Term** | **Weights** |
| 36 months | 1 |
| 60 months and FICO score 450-649 | 2 |
| 60 months and FICO score 650-850 | 3 |

Below prediction algorithms are implemented on each of these 3 clusters.

* Linear Regression
* Random Forest Regression
* Neural Network

We have selected Neural Network Regression to be our best prediction model based on RMSE value.

Below are the screen shots for the same.

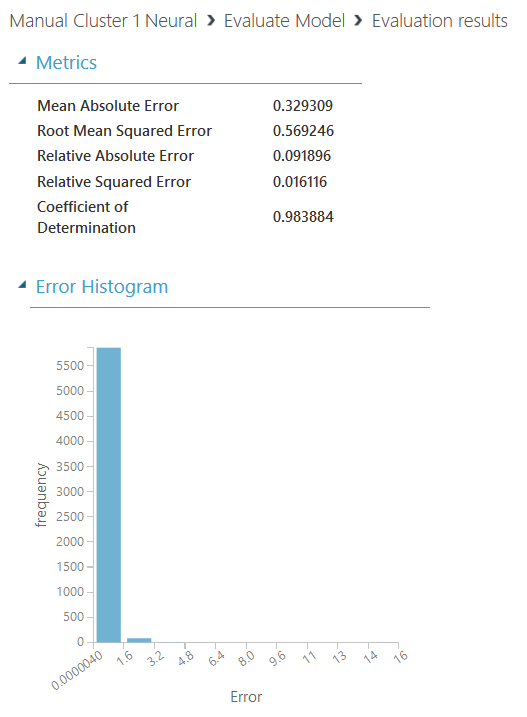


Figure : Neural Network result on Manual cluster 1

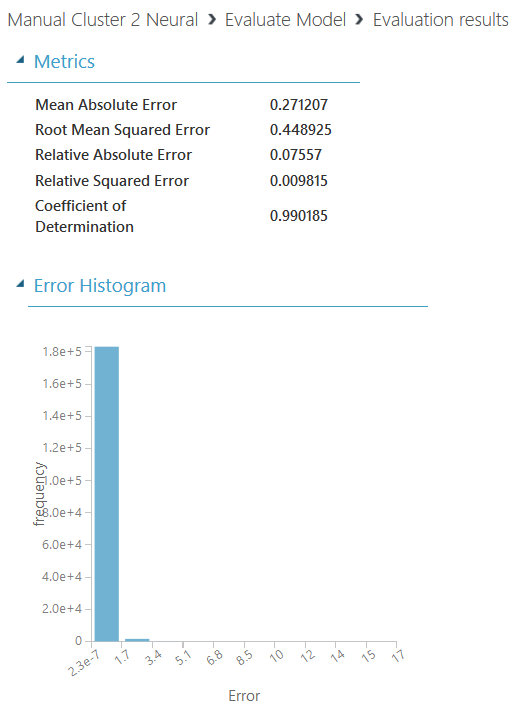


Figure : Neural Network results on Manual cluster 2

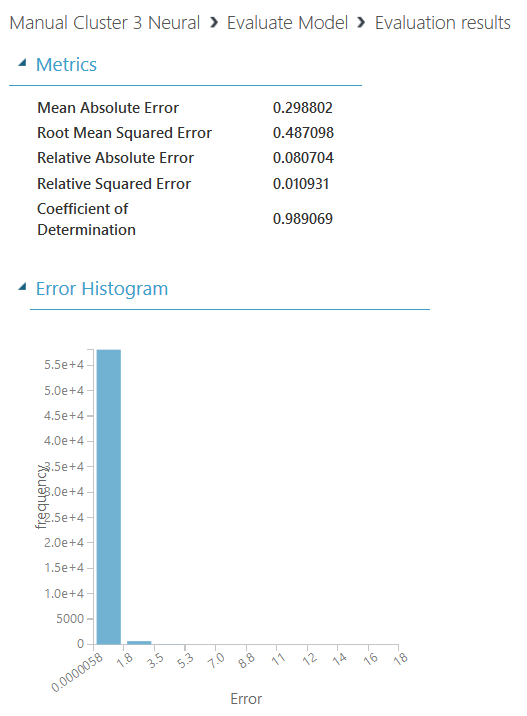
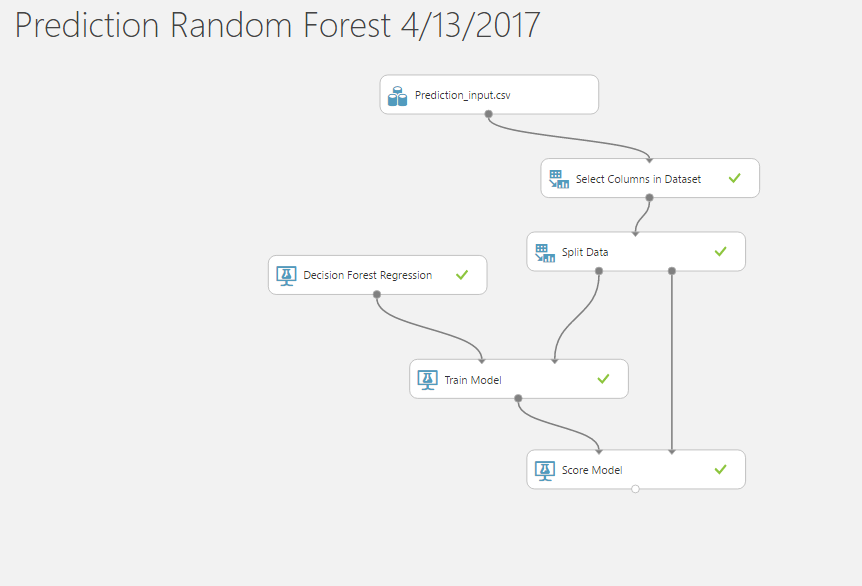
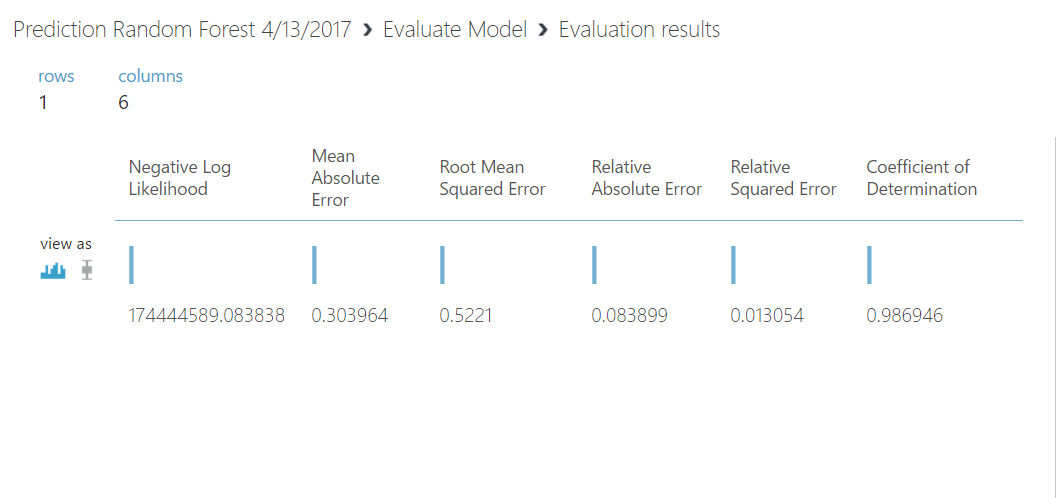


Figure : Neural Network results on Manual cluster 3.

**No Clustering**





The next aim was to predict the interest rate by training the data set using various Regression models. We started off this section by considering few shortlisted features from part 1. And then feature engineered those columns by using Hypothesis testing, generated the p-values for each variables of the data. We repeated the hypothesis testing and finally got a few variables to be considered for the best output. Later, we worked on Backward variable selection method to finalize the final set of columns which we need as an input from the user and based on those values, we will be calculating the interest rate.

As per the requirement, we were supposed to run Regression Algorithm like KNN, Random Forest, Neural Network and Linear Regression. We chose the best method among these 4 by considering the smallest Root Mean Squared Error (RMSE) value. We implemented the algorithms with the help of python code using ipynb files. There were various parameters we generated like MAPE, MAE, RMSE and R-Squared, but we considered RMSE as the main parameter on which we decide the accuracy and correctness of our Algorithms.

The value with the minimum Root Mean Squared Error is considered the best and based on all the values, we chose Random Forest to be the best model with RMSE of 0.52.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | RMSE | R-Squared | MAE |
| Linear Regression | 0.908 | 0.039 | 0.633 |
| Random Forest | 0.5221 | 0.013 | 0.303 |
| Neural Network | 0.5666 | 0.016 | 0.306 |

Based on the above RMSE value, we concluded that Random Forest is the best algorithm.

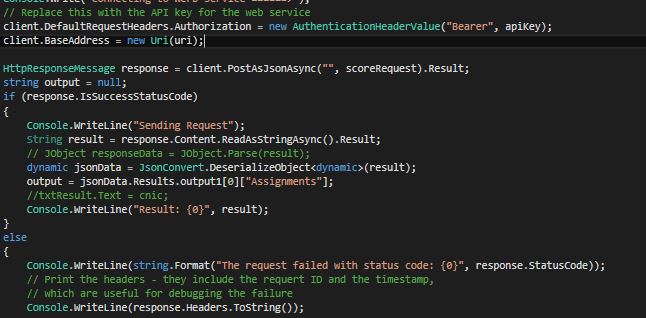
Deployment:

Once the best prediction model was selected for every cluster, we deployed them in Azure Machine Learning Studio by following the below steps:

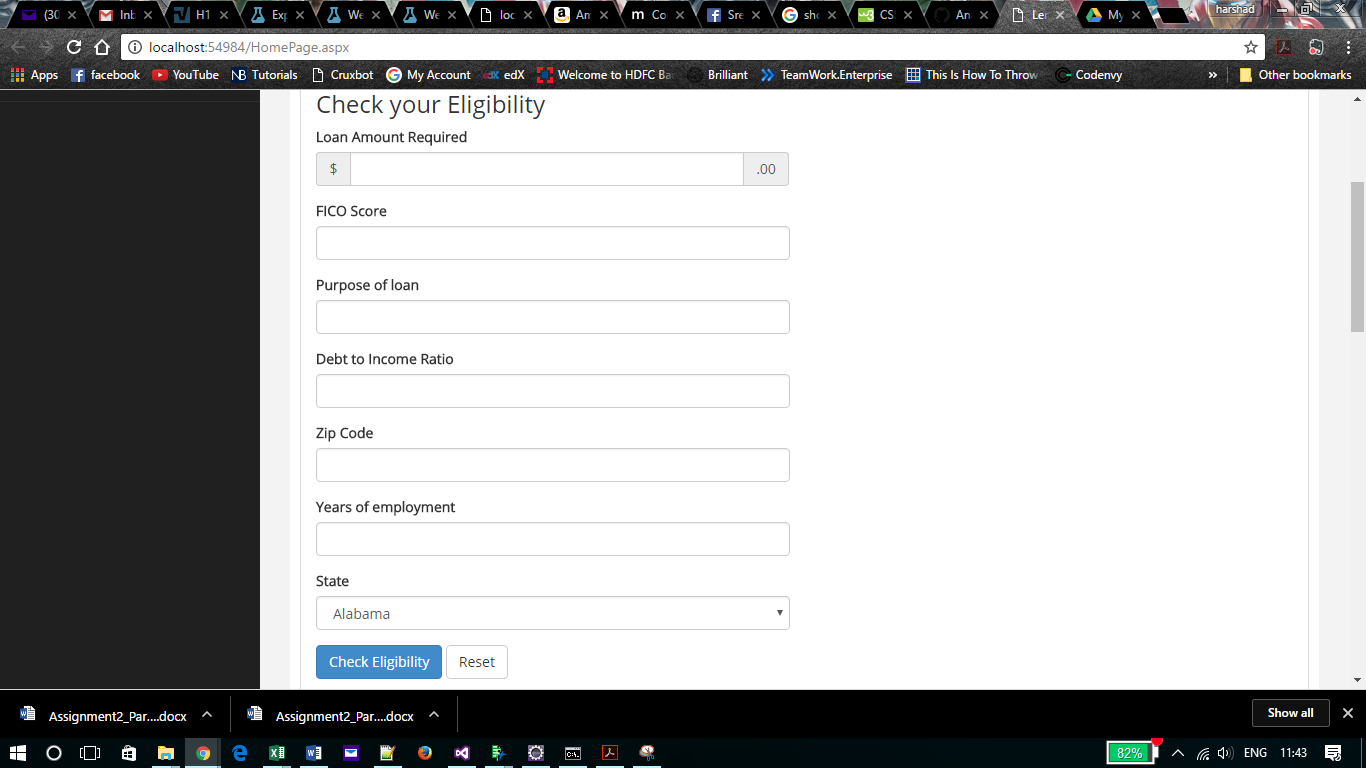
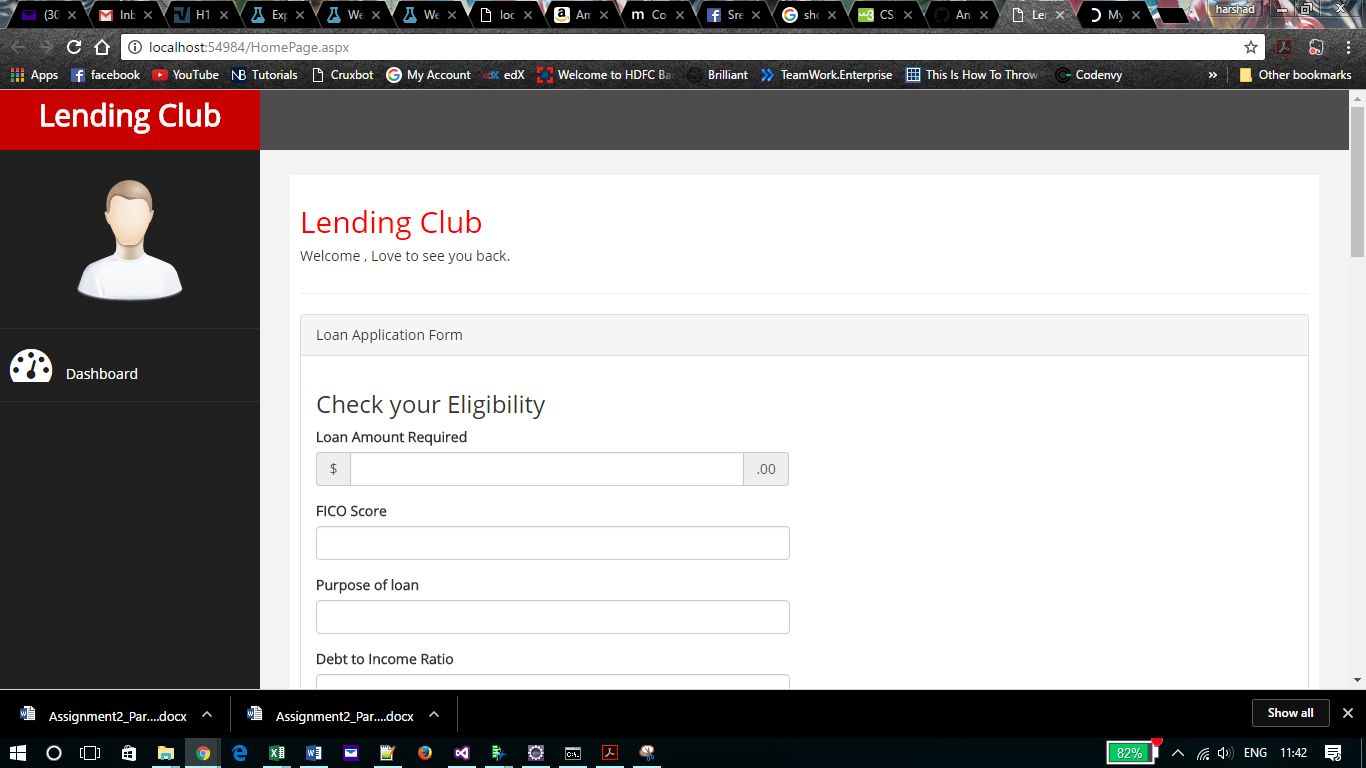
* Uploaded the datasets for each cluster
* Created a new experiment for each prediction model
* Imported the uploaded input file o the experiment
* Selected required columns from the file
* Split the input data into 80-20 format for training and testing the model
* Trained the best model.
* Tested the model by using the Score Model
* Evaluated the model using Evaluation Model
* Create a Predictive Web Service for the model
* Edit the Select Columns from Dataset Model to remove the column to be predicted
* Run the model again and deploy it to web service
* Test the model for an observation
* Consume this model using API key and Request-Response URL

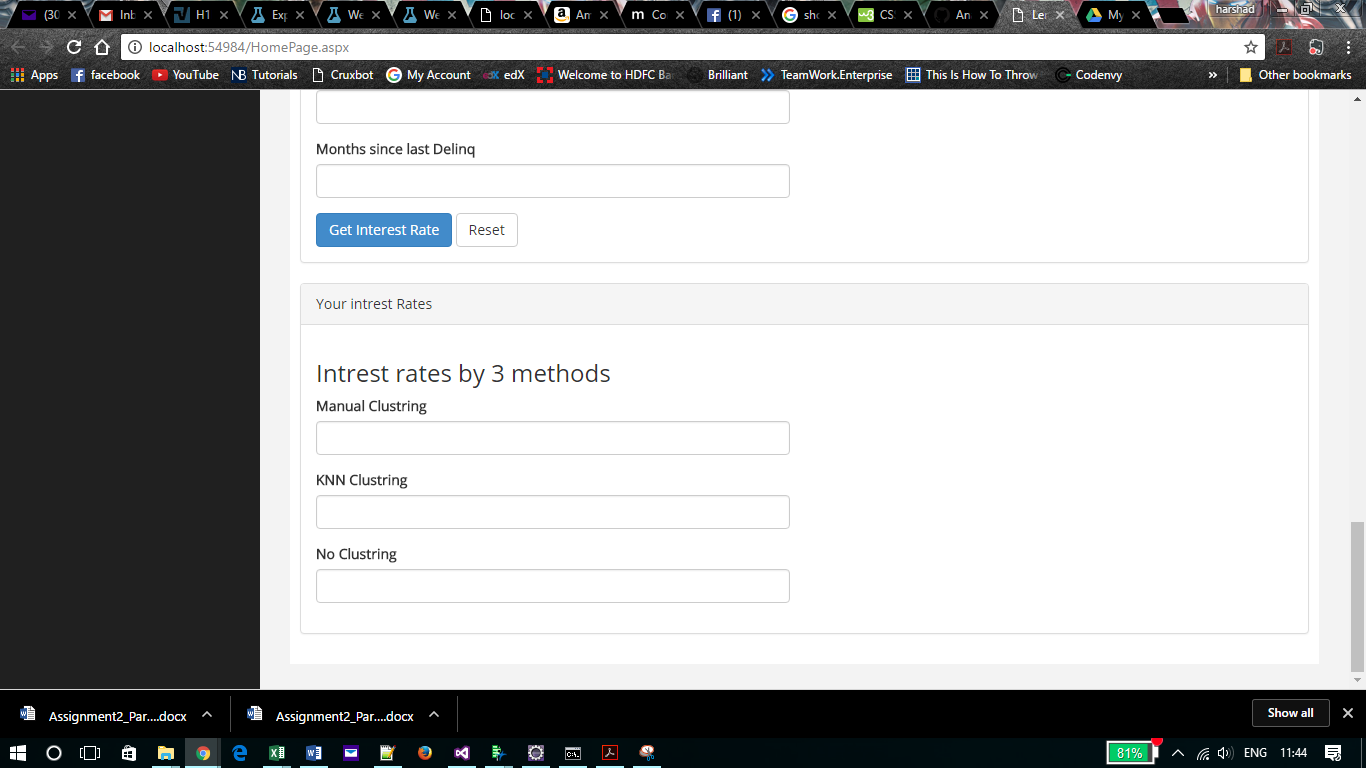
**Consuming the Web service using C#**

We have used C# and asp.net to form the request, call the API and read the response sent by the Rest API.



The input is given by a asp.net form which is designed to take all the inputs and show the output.





We had to implement this web service using some synchronous method so we have changed the request to Synchronous method.