PRIYANJNA SHARMA

SUMIT DEO

GAUTAM PAWAR

ADS Final ProjectP roposal

TEAM 2

Table of Contents

[2 DATA PREPARATION 4](#_Toc479972235)

[2.1 Feature Engineering 4](#_Toc479972236)

[3 CLASSIFICATION 5](#_Toc479972238)

[3.1 Logistic Regression 6](#_Toc479972239)

[3.1.1 Confusion Matrix 6](#_Toc479972240)

[3.1.2 Accuracy Measures: 6](#_Toc479972241)

[3.1.3 Logistic Regression Report: 6](#_Toc479972242)

[3.1.4 ROC Curve: 7](#_Toc479972243)

[3.2 Random forest 7](#_Toc479972244)

[3.2.1 Confusion Matrix 7](#_Toc479972245)

[3.2.2 Random Forest Accuracy Measures: 8](#_Toc479972246)

[3.2.3 Random Forest Classification Report: 8](#_Toc479972247)

[3.2.4 ROC Curve: 8](#_Toc479972248)

[3.3 Neural Network 9](#_Toc479972249)

[3.3.1 Confusion Matrix 9](#_Toc479972250)

[3.3.2 Accuracy Measures: 9](#_Toc479972251)

[3.3.3 Neural Network Classification Report: 10](#_Toc479972252)

[3.3.4 ROC Curve: 11](#_Toc479972253)

[3.4 Best Classification Model (Random Forest) 11](#_Toc479972254)

[3.5 Azure ML Studio for building random forest model: 11](#_Toc479972255)

[4 CLUSTERING and PREDICTION 13](#_Toc479972256)

[4.1 Manual Clustering 13](#_Toc479972257)

[4.2 Prediction On Manual Clustering 14](#_Toc479972258)

[4.3 Clustering using algorithm 16](#_Toc479972259)

[4.5 Prediction Without Clustering 20](#_Toc479972262)

[5 Lending Loan Club Application (Webpage) 20](#_Toc479972263)

[5.1 CLASSIFICATION 22](#_Toc479972264)

[5.1.1 Inputs to REST API 23](#_Toc479972265)

[5.2 PREDICTION 23](#_Toc479972266)

[5.2.1 Inputs to REST API 23](#_Toc479972267)

[ Term 23](#_Toc479972268)

[ Grade 23](#_Toc479972269)

[ Purpose 23](#_Toc479972270)

[ Inquiries in Last 6 months 24](#_Toc479972271)

[ Public records 24](#_Toc479972272)

[ Loan Status: 24](#_Toc479972273)

[ Home Ownership: 24](#_Toc479972274)

[ Verification Status: 24](#_Toc479972275)

[ Application Type 24](#_Toc479972276)

[ Loan Amount 25](#_Toc479972277)

[ Annual Income 25](#_Toc479972278)

[5.3 Result 1: Manual Clustering 25](#_Toc479972279)

[5.4 Result 2: K-means Clusters 27](#_Toc479972280)

[5.5 Result 3: No Clusters 27](#_Toc479972281)

[6 Contribution 29](#_Toc479972283)

[7 REFERENCES 30](#_Toc479972285)

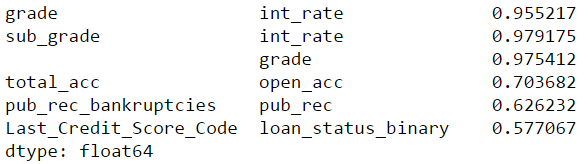
# DATA PREPARATION

## Feature Engineering

* Removed all the post loan related features like out\_prncp, total\_rec\_prncp, total\_rec\_late\_fee, recoveries etc that won’t be of any use in predication and classification.
* Performed different feature engineering algorithms like Recursive Feature Elimination (RFE), Lasso and Ridge regression etc. to evaluate relation between interest rate and other features.

## 

* Evaluated correlation between features to eliminate highly correlated features.



# CLASSIFICATION

1. Language used: Python

1. Added a new column **LoanApproval** in both files LoanData.csv and RejectLoanData.csv that will act as response variable stating whether the loan is approved or not and setting the value as 1 and 0 respectively
2. We have merged both the files to perform classification on combined dataset.
3. Selected following features for building classification models:

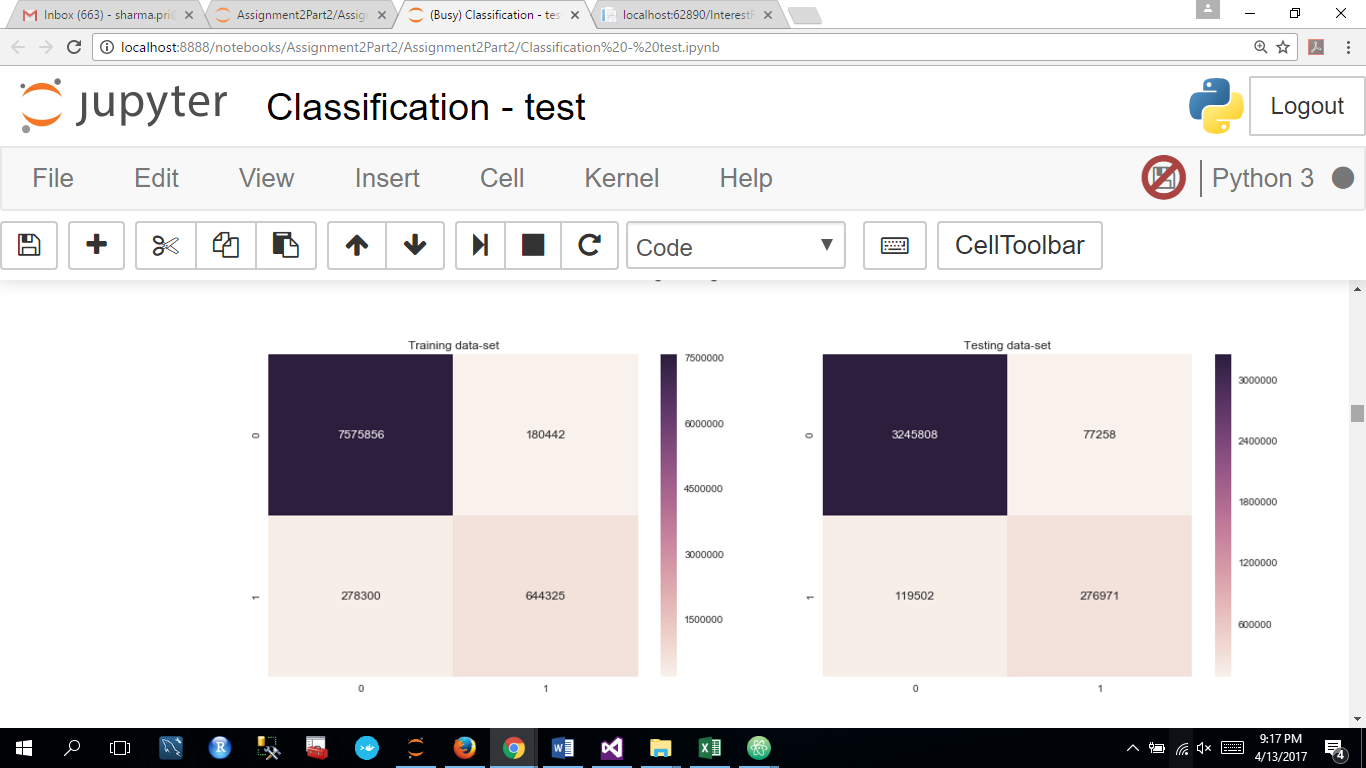


1. Used sklearn, matplotlib, seaborn packages to build and plot the models.

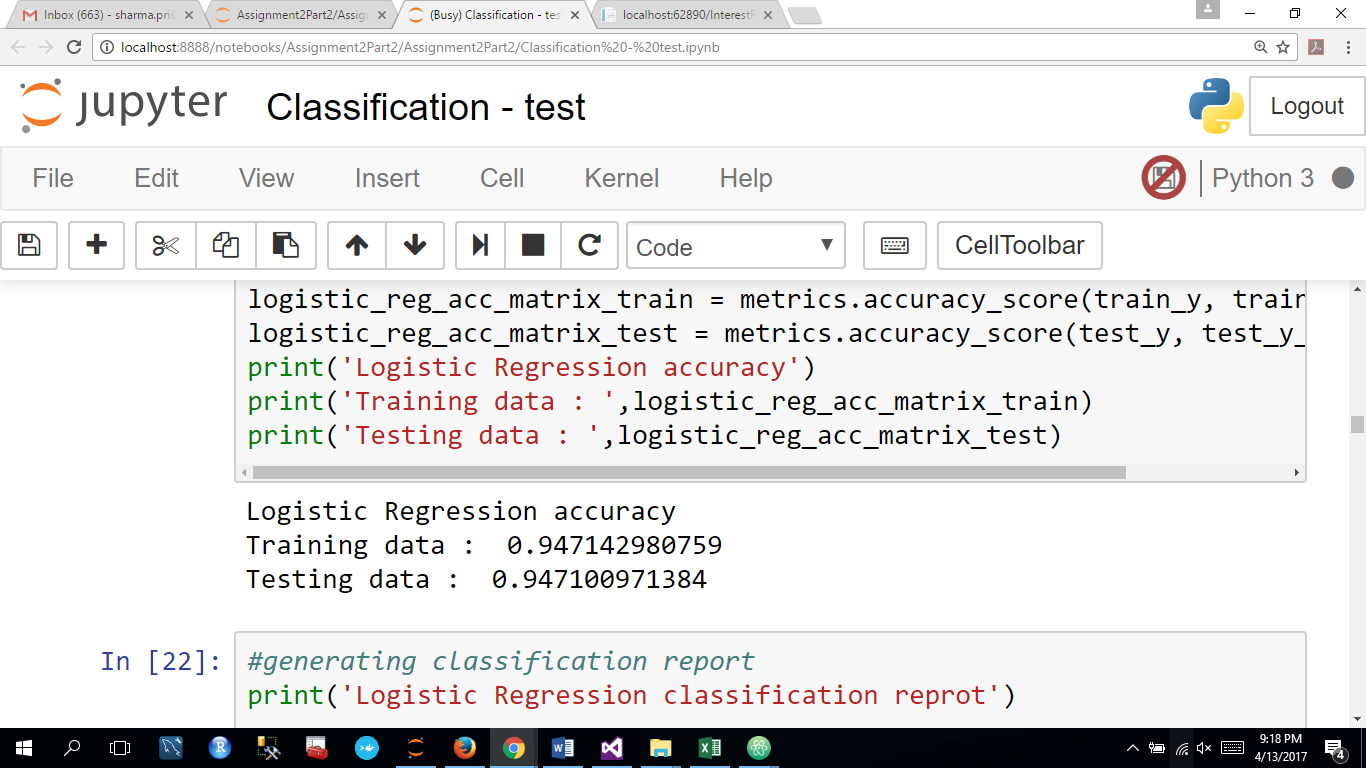
## Logistic Regression

Using Jupyter Notebook, divided the data into train and test. Set the LoanApproval feature as the target variable and perform the regression modeling.

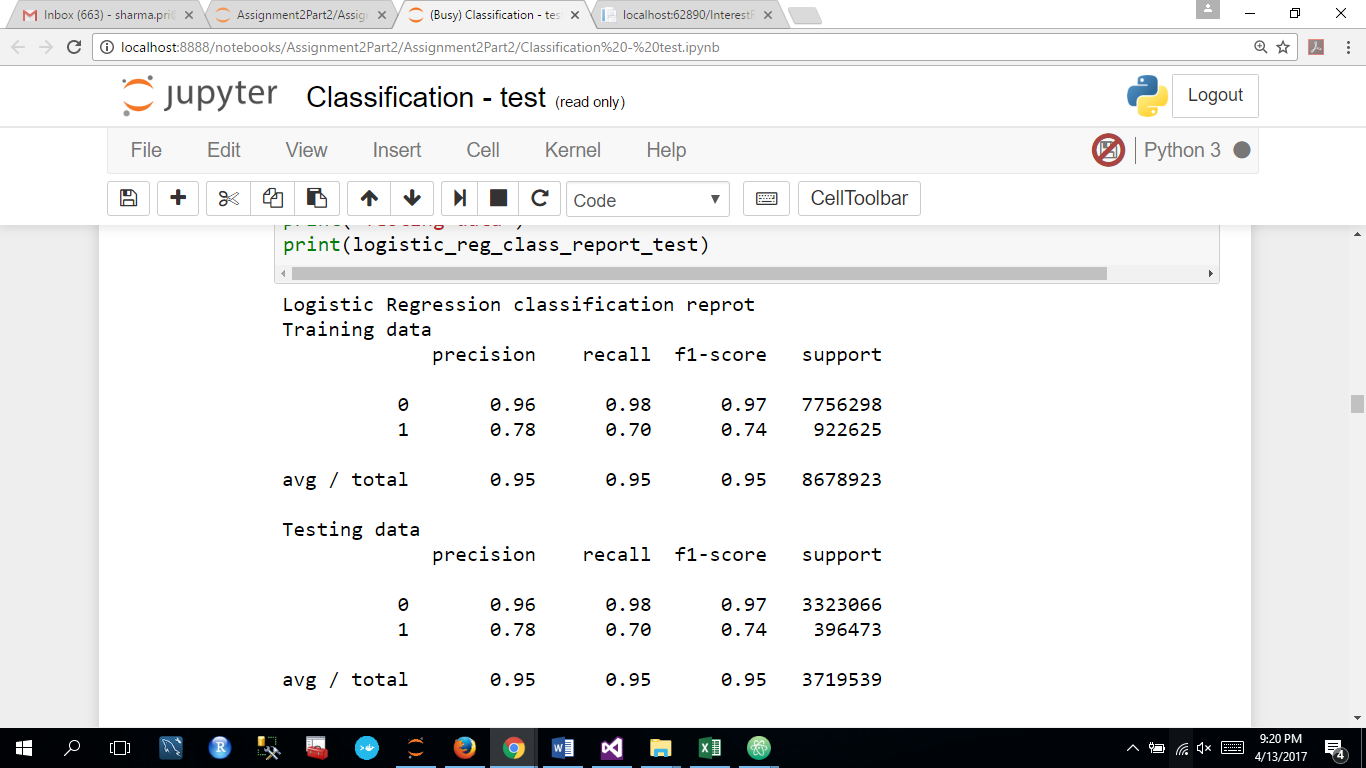
### Confusion Matrix



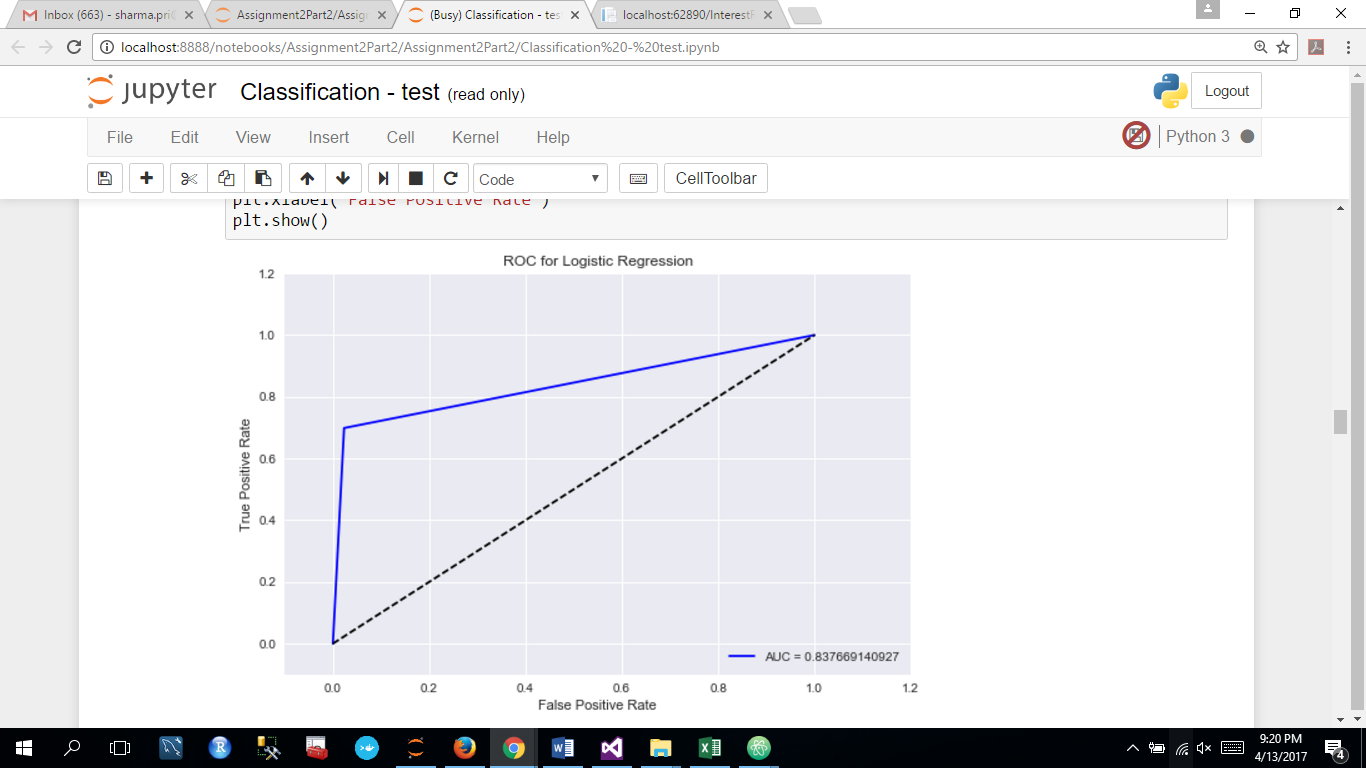
### Accuracy Measures:



### Logistic Regression Report:

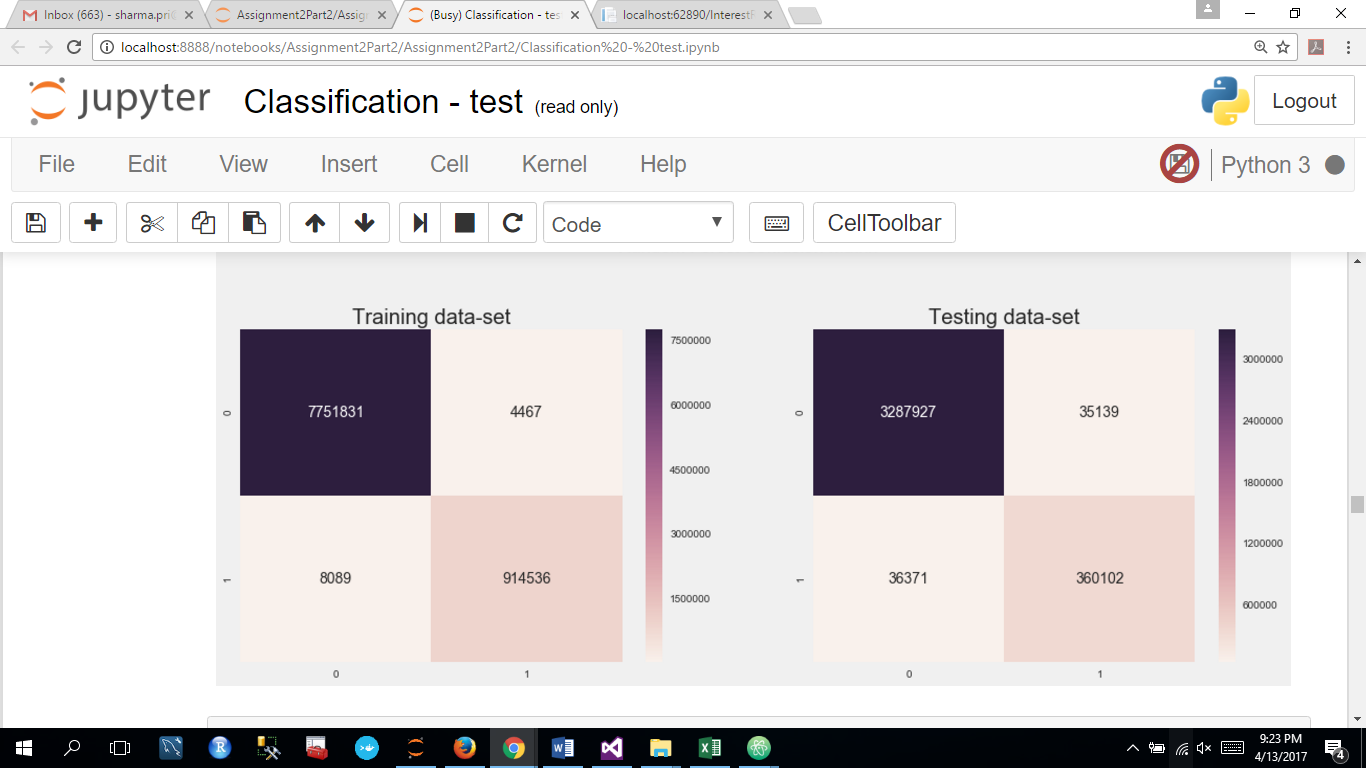


### ROC Curve:

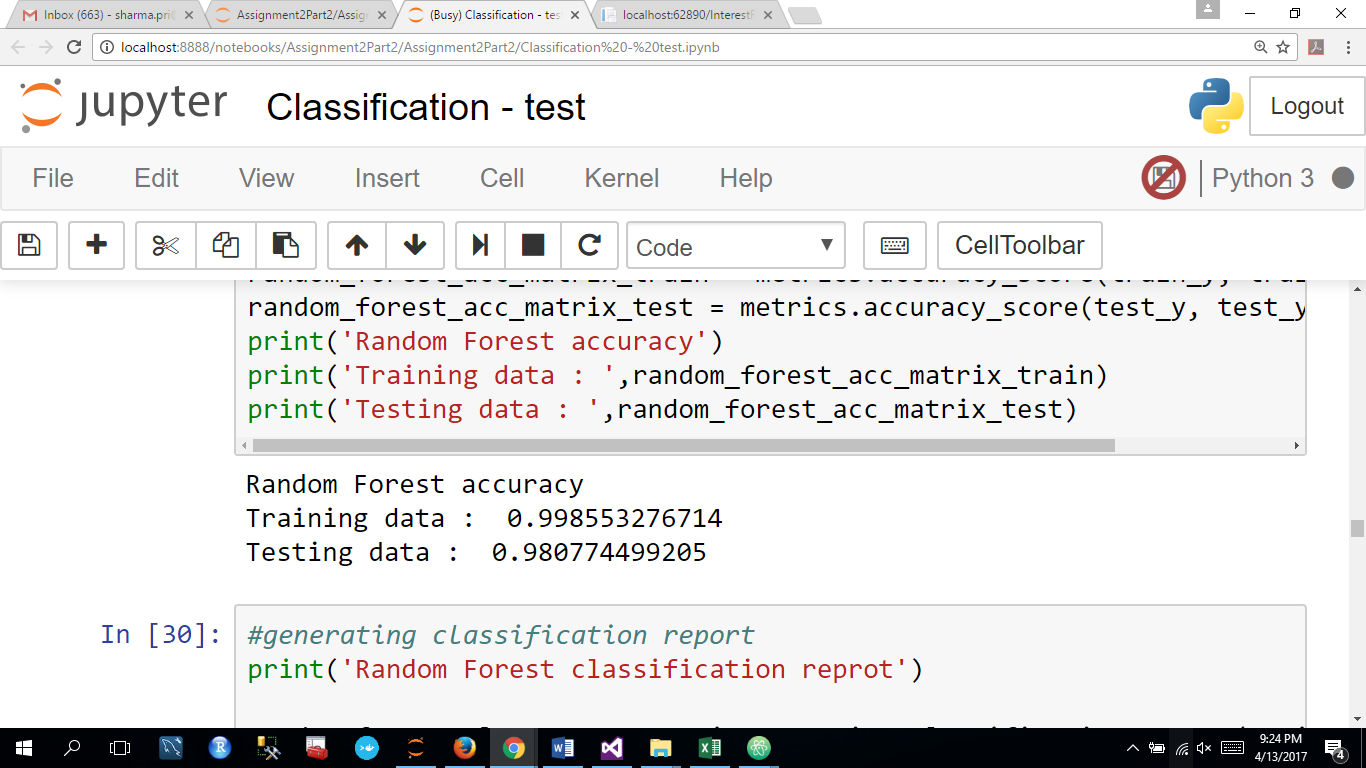


## Random forest

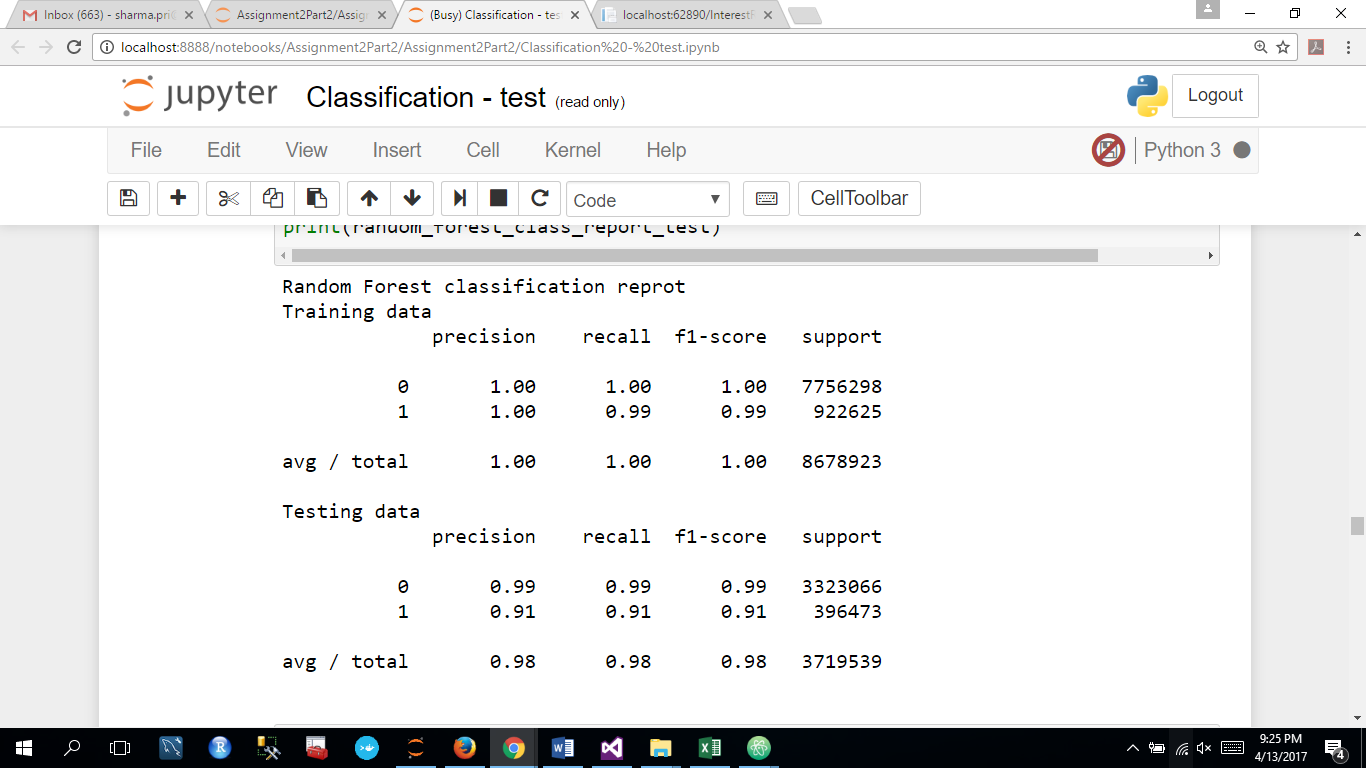
### Confusion Matrix



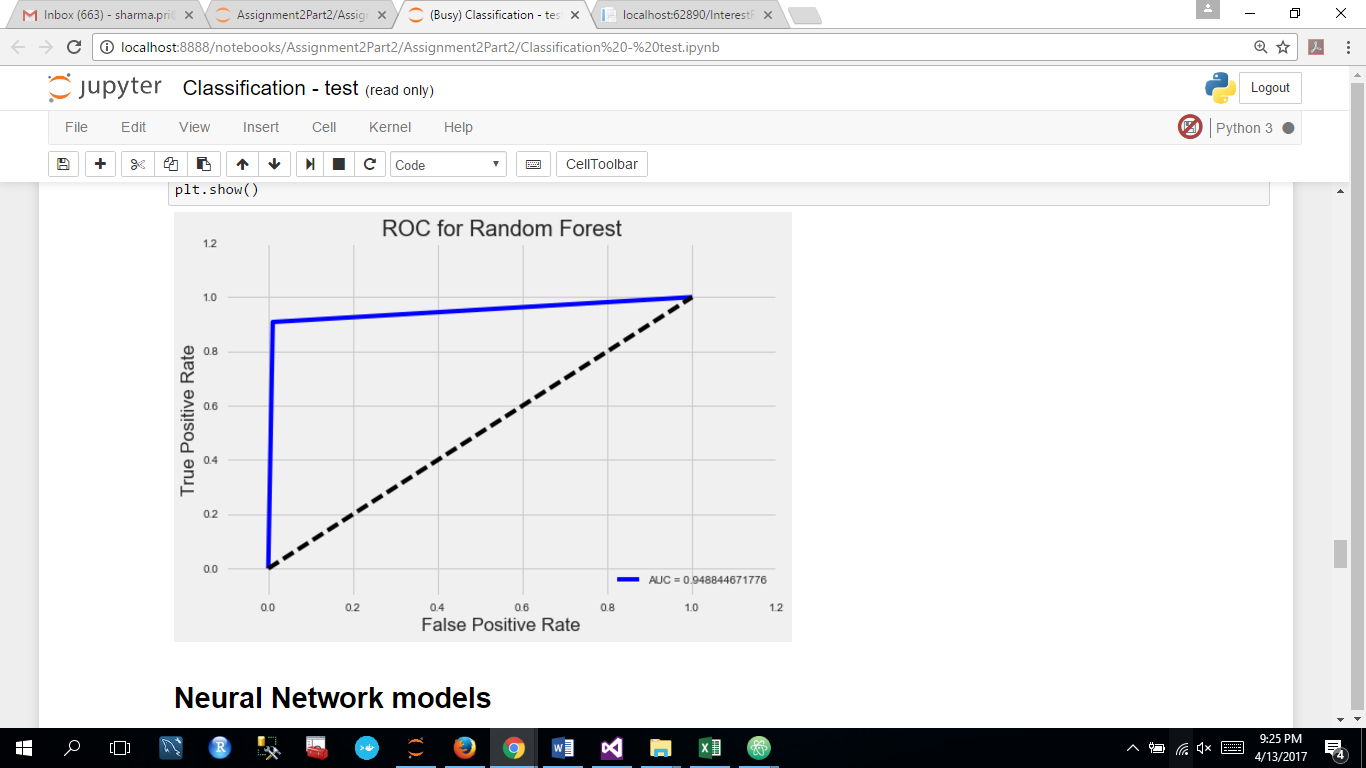
### Random Forest Accuracy Measures:



### Random Forest Classification Report:

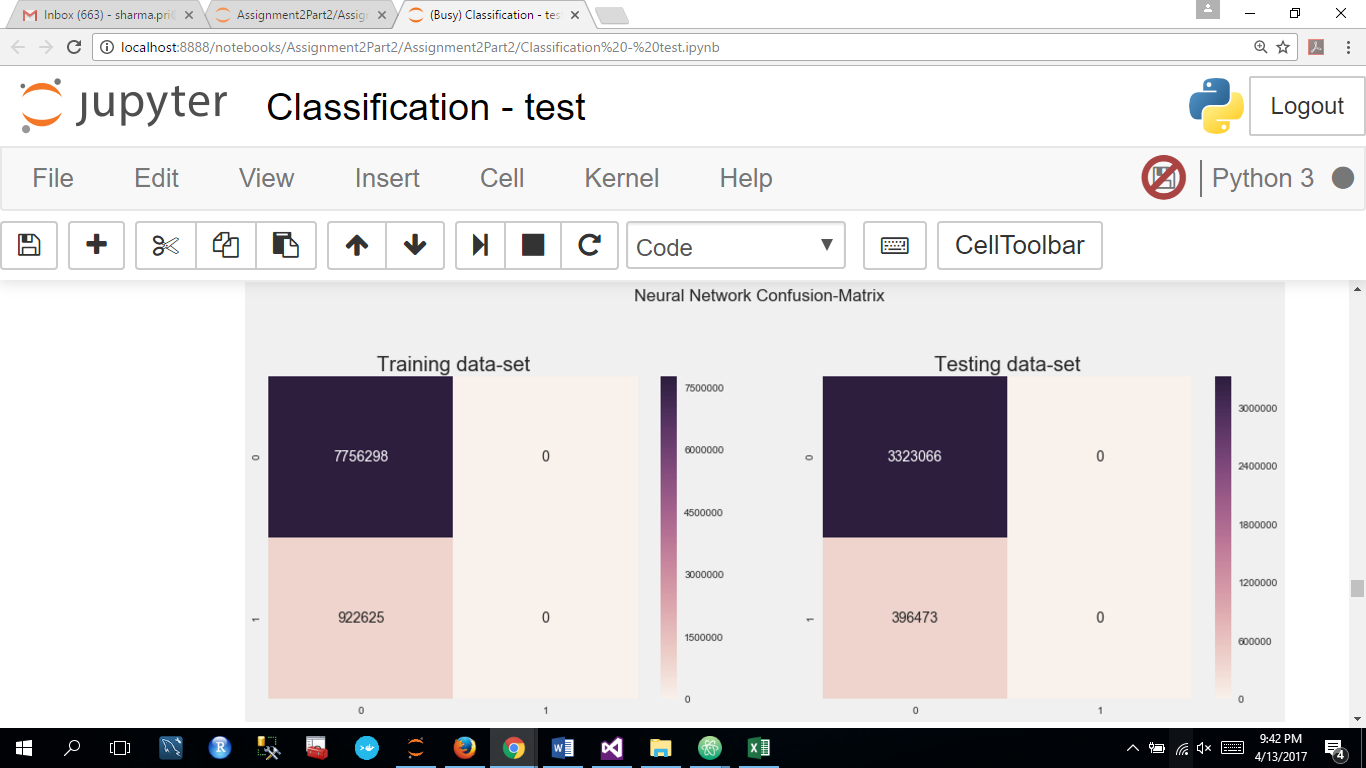


### ROC Curve:

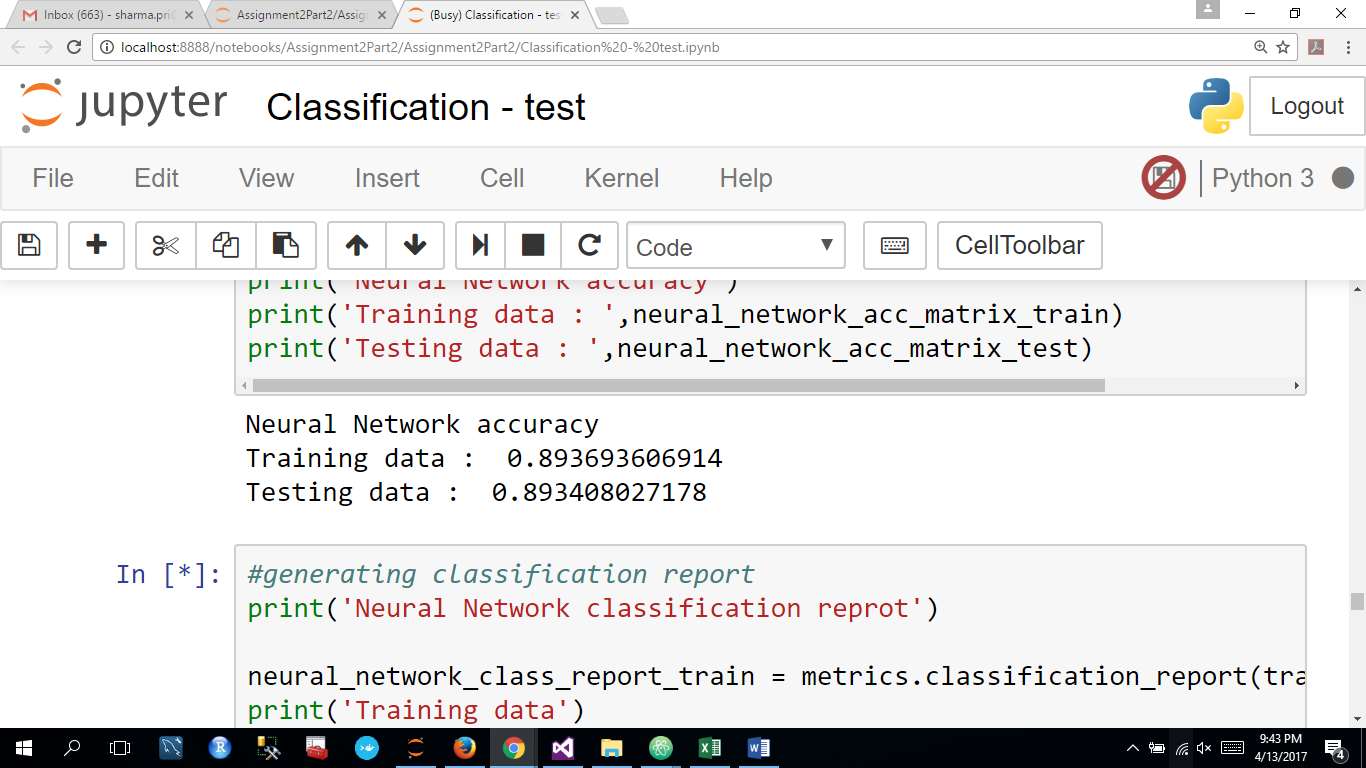


## Neural Network

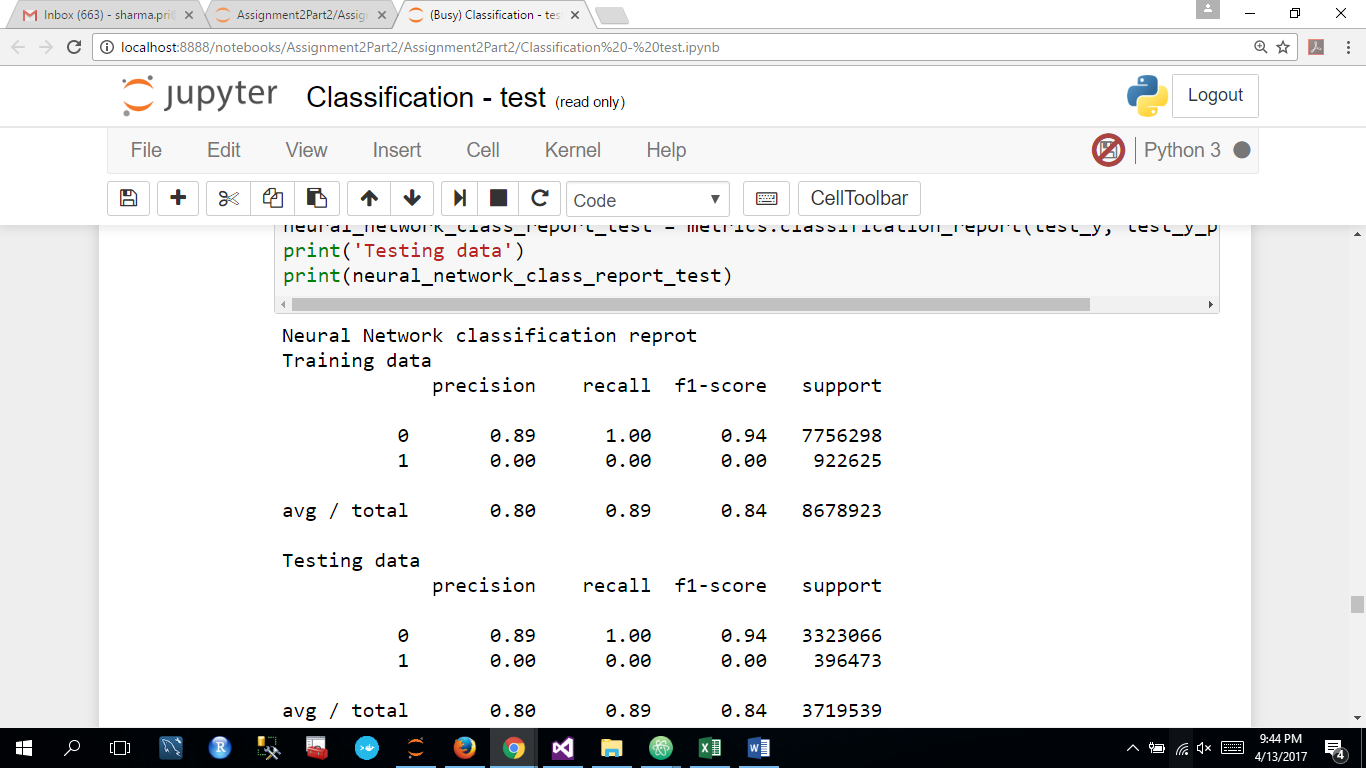
### Confusion Matrix



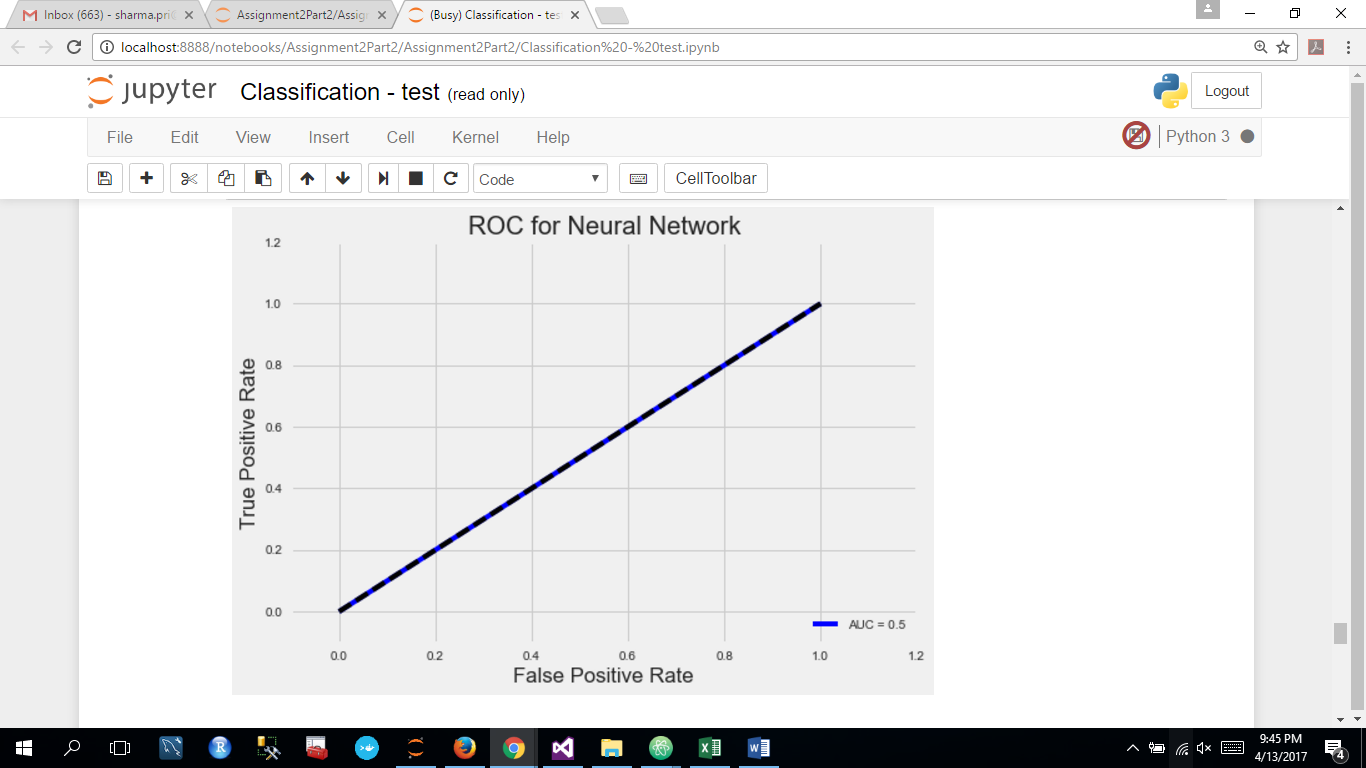
### Accuracy Measures:



### Neural Network Classification Report:



### ROC Curve:



## Best Classification Model (Random Forest)

We have selected random forest as our best model with 98.03% accuracy and implemented it on Microsoft Azure ML Studio.

|  |  |
| --- | --- |
|  | **Accuracy (Test Data) %** |
| Random Forest | 98.03 |
| Logistic Regression | 94.71 |
| Neural Network | 89.34 |

## Azure ML Studio for building random forest model:

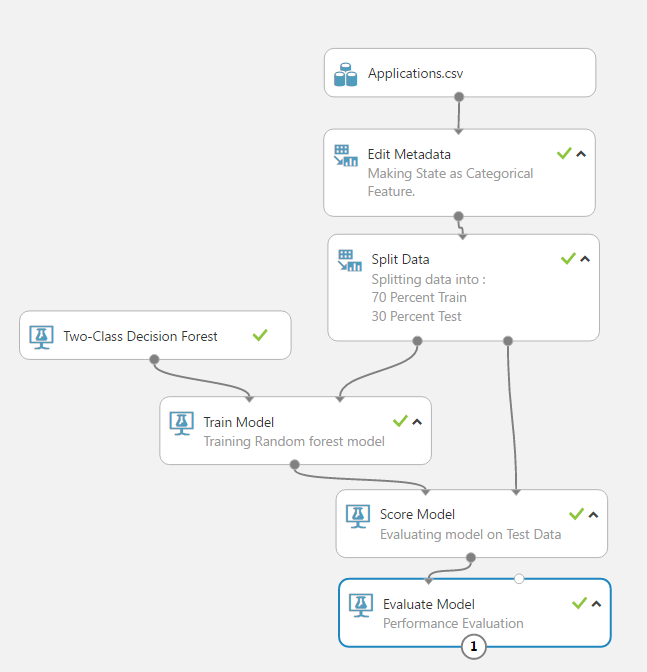
We have used Two-Class Decision Forest algorithm to perform classification modeling in the Azure studio.

1. Imported data into Azure ML storage.
2. Performed data manipulation in the initial phase.
3. Deploy.
4. Evaluate performance.
5. Build web service and generate API.

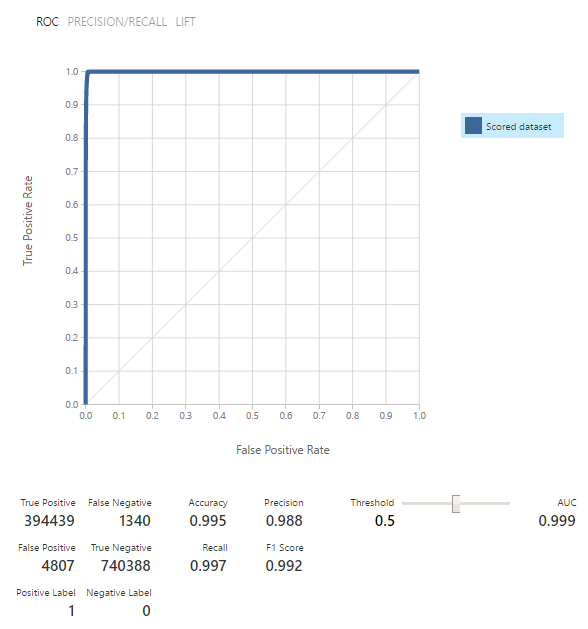
We have implemented features that we will use to define whether to give loan or not as:

Amount Requested, State, Employee Length, Credit Score, Application Year and DTI.

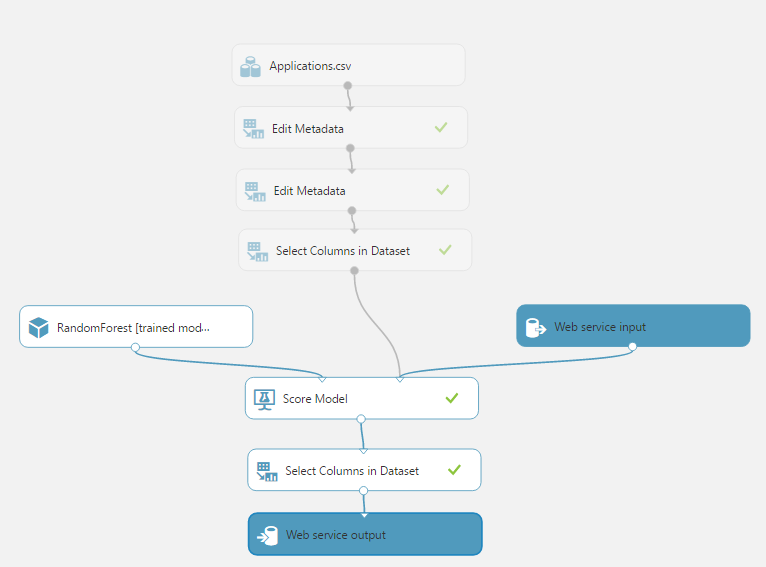
**Random Forest Model Classification**



**Performance Evaluation**



**Deploying Web Service**



# CLUSTERING and PREDICTION

## Manual Clustering

**Features Used:** Credit Score Code(Derived), Purpose

**Platform Used**: Jupyter notebook using python.

* **Step 1:**

We first divided out data set into 4 parts based on credit score code:

**Credit Score Code**: derived column

This column represents numeric category codes for credit score bins that we created in feature engineering.

* + **Part 1:**

This part of the dataset contains all the rows of the original dataset where credit score code <=10

* + **Part 2:**

This part of the dataset contains all rows of original dataset where credit score code is   
11

* + **Part 3:**

This part of the dataset contains all rows of original dataset where credit score code is   
12

* + **Part 4:**

This part of the dataset contains all rows of original dataset where credit score code is   
>=13

Next we further divided the above-mentioned parts into 2 groups each based on the purpose.

Group 1: where purpose = “debt-consolidation”

Group 2: for all other purposes

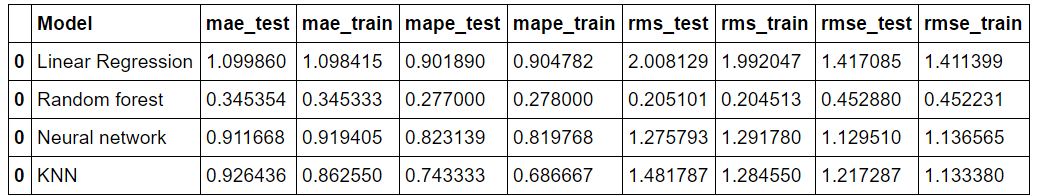
Applying the above steps, we have total of 8 clusters which we will be using further for prediction.



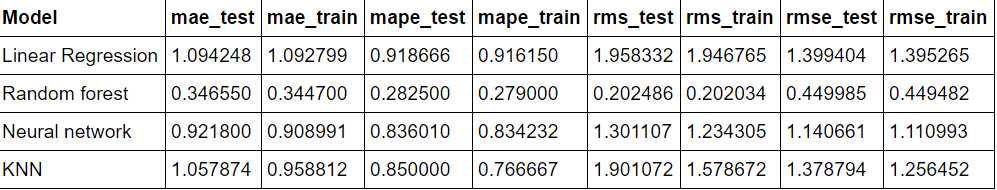
## Prediction On Manual Clustering

On each (8) clusters, we have performed prediction using Linear Regression, Random Forest, Neural Network and KNN algorithms. Analysis are below mentioned:

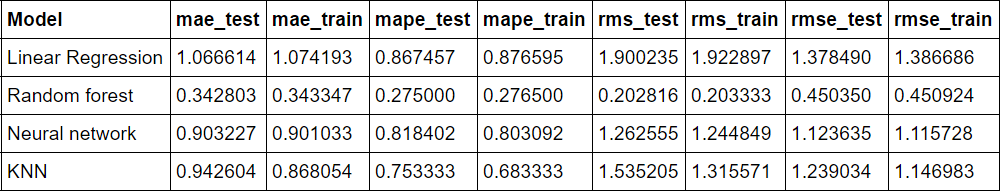
**Cluster 0:**



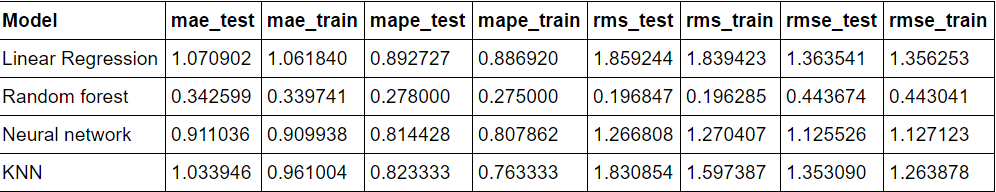
**Cluster 1:**



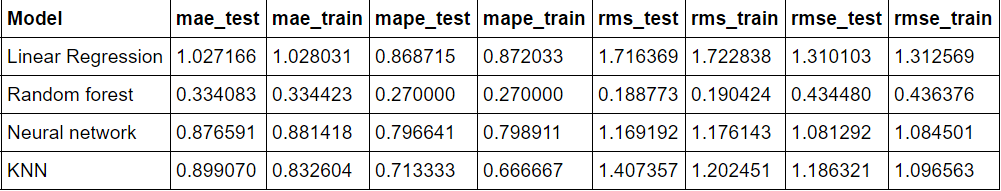
**Cluster 2:**



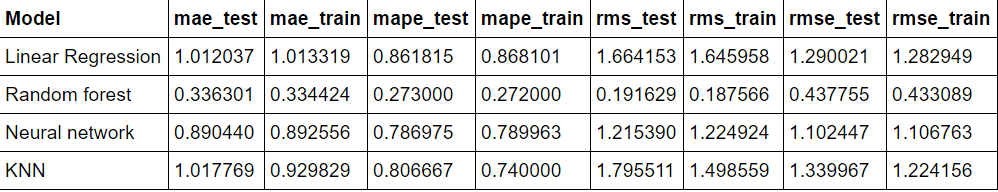
**Cluster 3:**



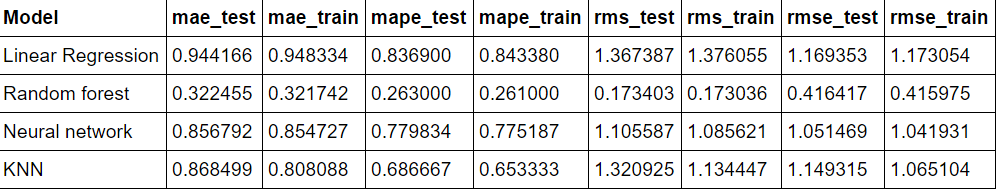
**Cluster 4:**



**Cluster 5:**



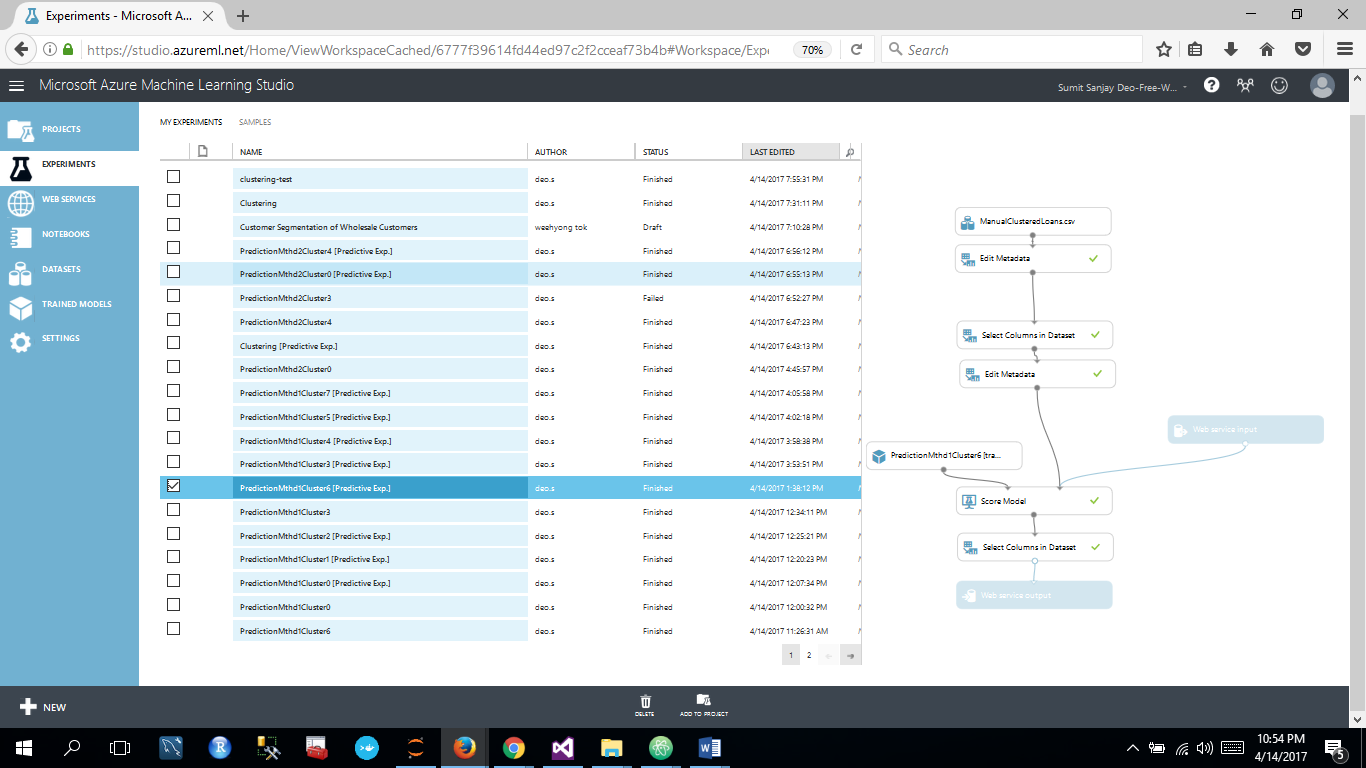
**Cluster 6:**



**Cluster 7:**



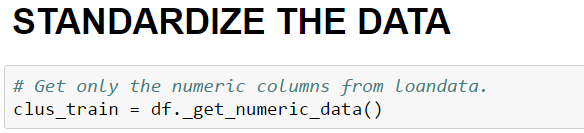
* Comparing the RMSE values of all the algorithms in each of the cluster, our best algorithm came out to be Random Forest.
* We then deployed Random Forest on Azure Machine Learning studio for all the 8 clusters.



## Clustering using algorithm

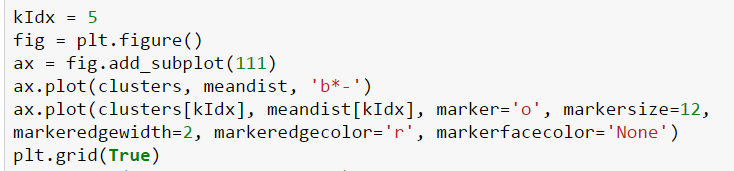
We have used k-means clustering algorithm to cluster the dataset

**Step1:** Get only the numeric data to standardize the dataset.



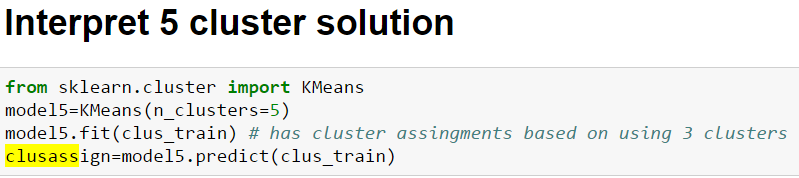
**Step2: Find the value of k using Elbow Method:**

Analyzing the curve, we have taken value of k as 5

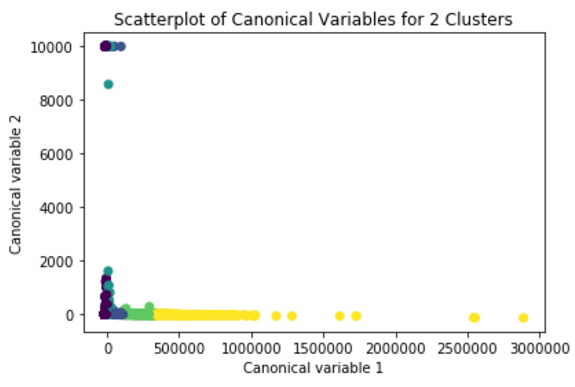


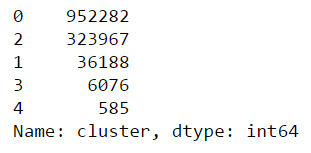


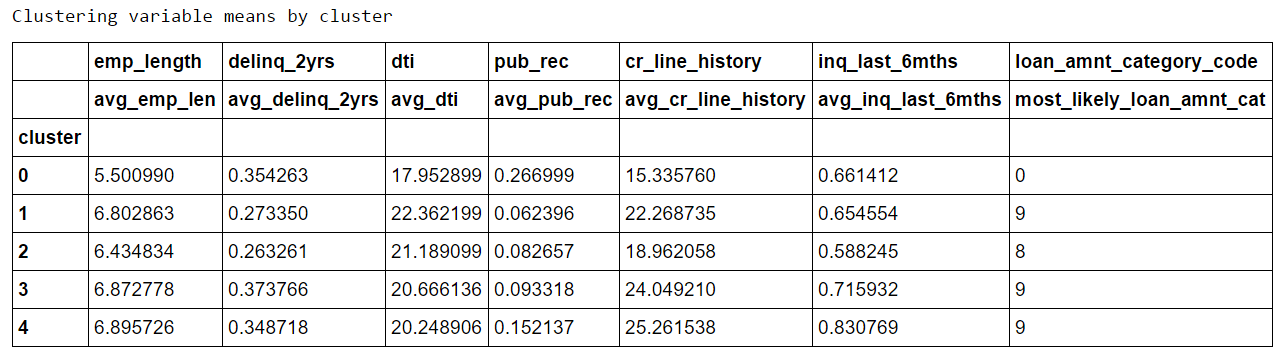
**Step3:**



**Step4:**



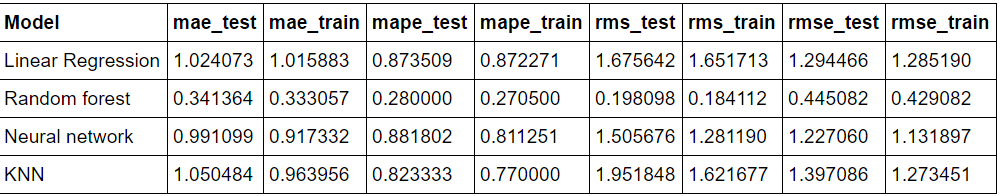




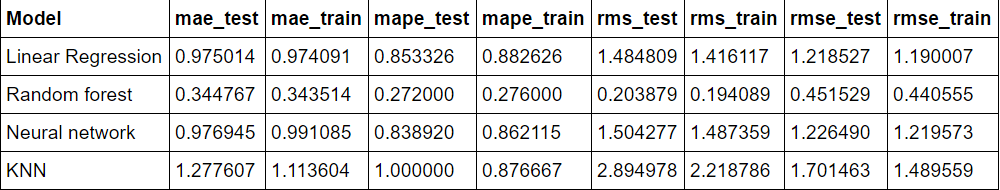
## Prediction on Clusters by Algorithm:

### We have performed 4 prediction models on the 5 clusters generated from the k-means algorithm. Summary measures of each clusters are listed below:

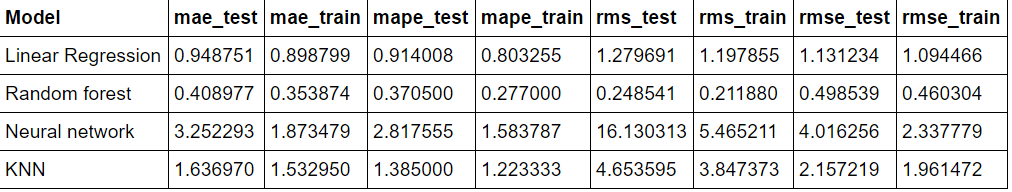
1. **Cluster 1:**



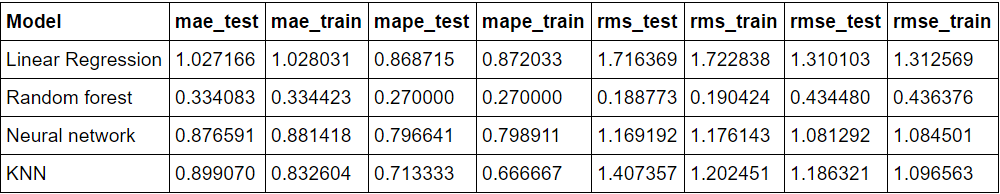
1. **Cluster 2:**



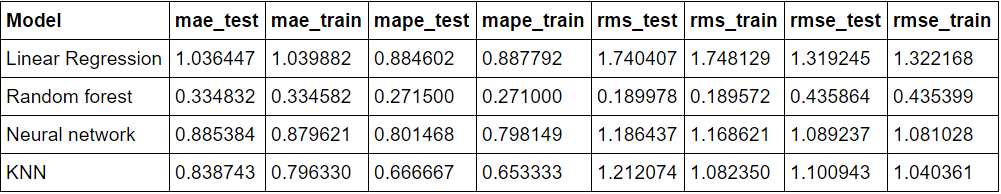
1. **Cluster 3:**



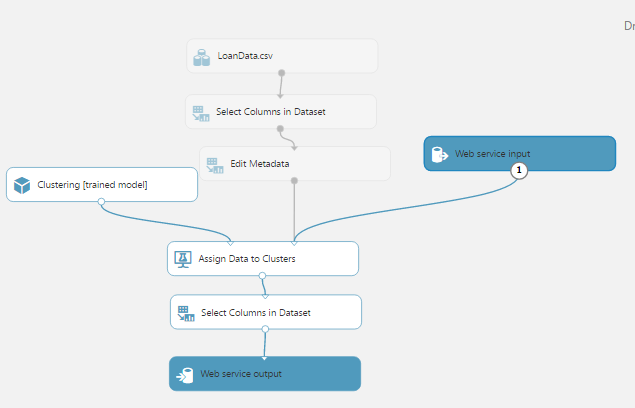
1. **Cluster 4:**

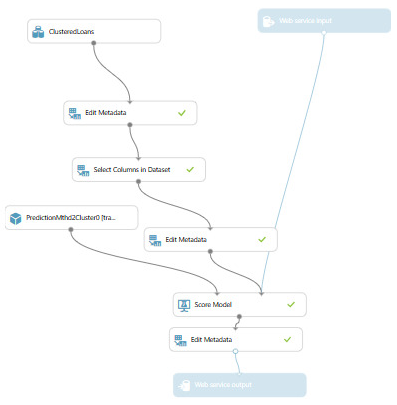


1. **Cluster 5:**



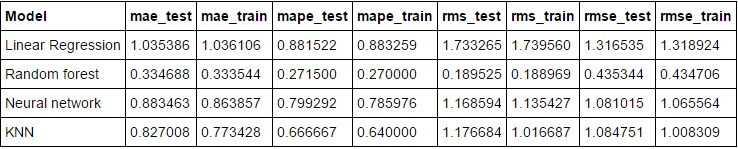
* Comparing the RMSE values of all the algorithms in each of the cluster, our best algorithm came out to be Random Forest.
* We then deployed Random Forest on Azure Machine Learning studio for all the clusters as shown in the image below.



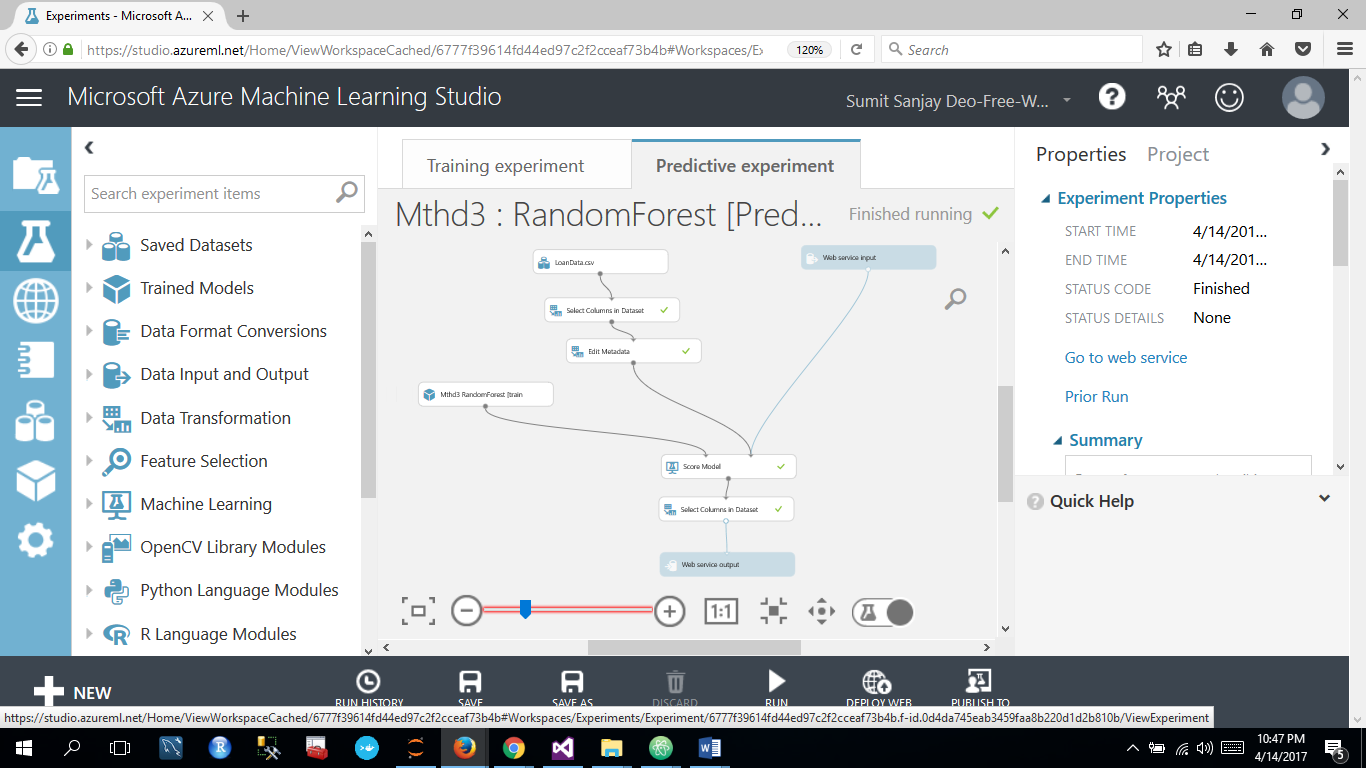


## Prediction Without Clustering

* We ran the above 4 algorithms to our original data set and found Random Forest to be the best again by comparing the RMSE values.



* Then deployed Random Forest on Azure Machine Learning studio for the original dataset

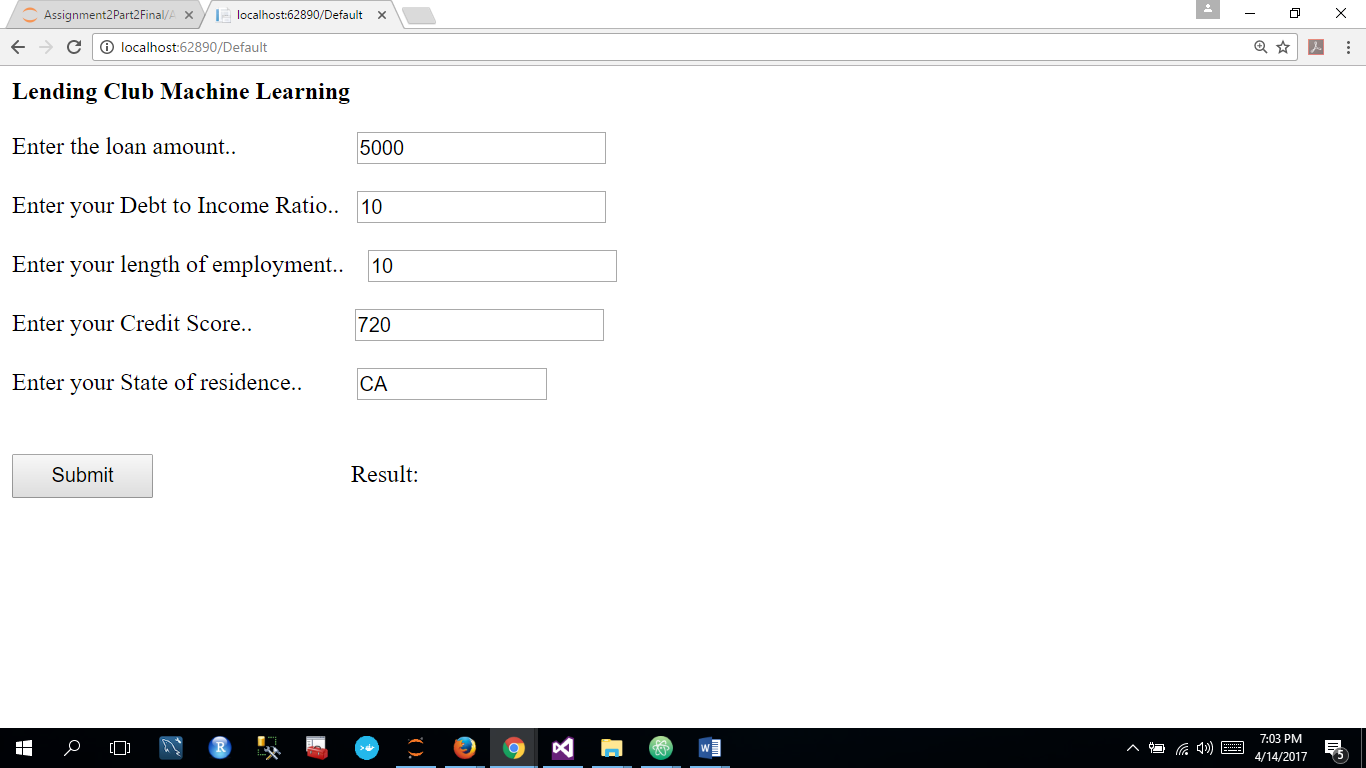


# Lending Loan Club Application (Webpage)

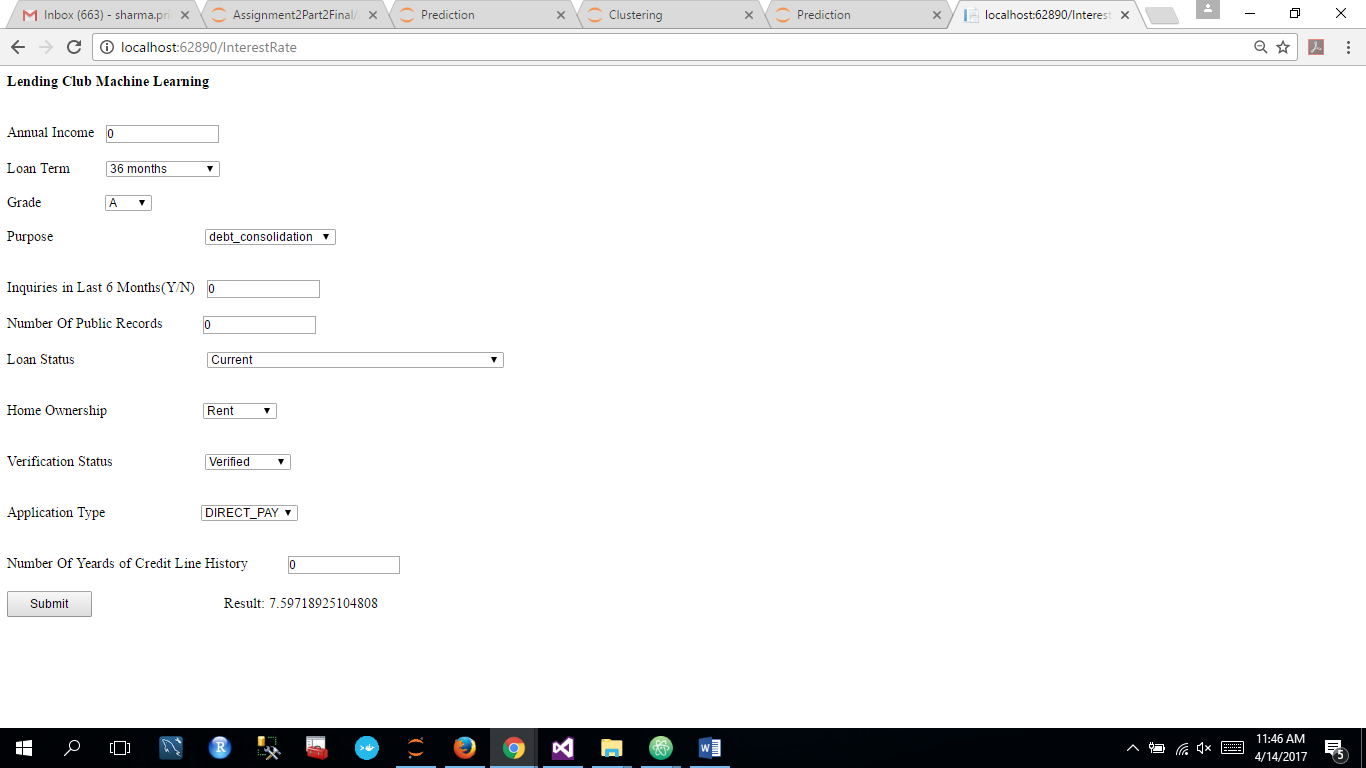
Our application has two parts:

* **Front End:** That captures basic information from the user and tells if the loan can be granted or not.

Webpage: we have used Asp.net to create web forms that take inputs from the users and give the prediction interest rate as result.



If loan can be granted, our application is directed to the next page where more details of the borrower are entered to give the final Interest Rate to the user.



Final Interest Rate is returned from three methods of clustering:

1. Manual Clusters
2. Clusters formed by Applying K means
3. No clusters

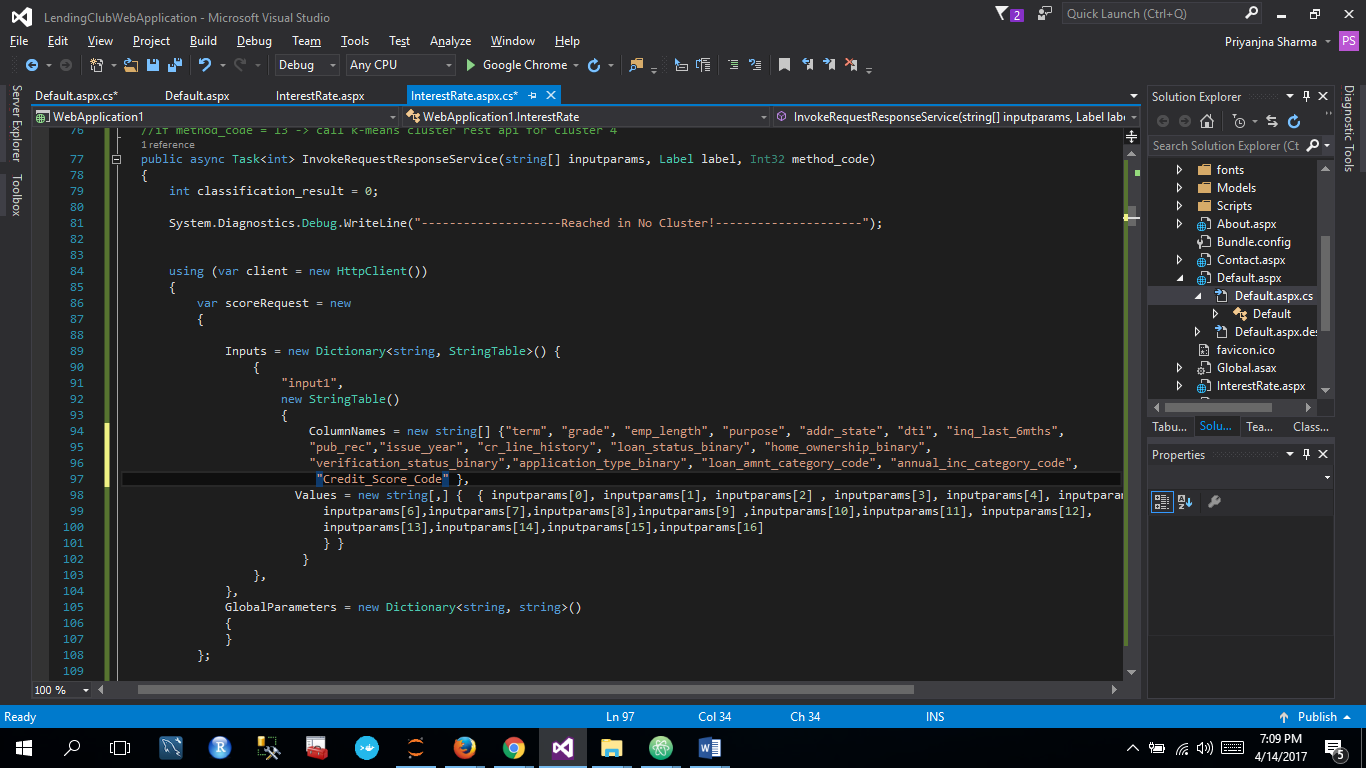
* **Back End:** At the back end, our application accesses the REST API of algorithms implemented on Azure Machine Learning studio to classify and predict the results.

For Classification: it accesses the REST API of Random Forest Algorithm and tells us if loan can be granted or not

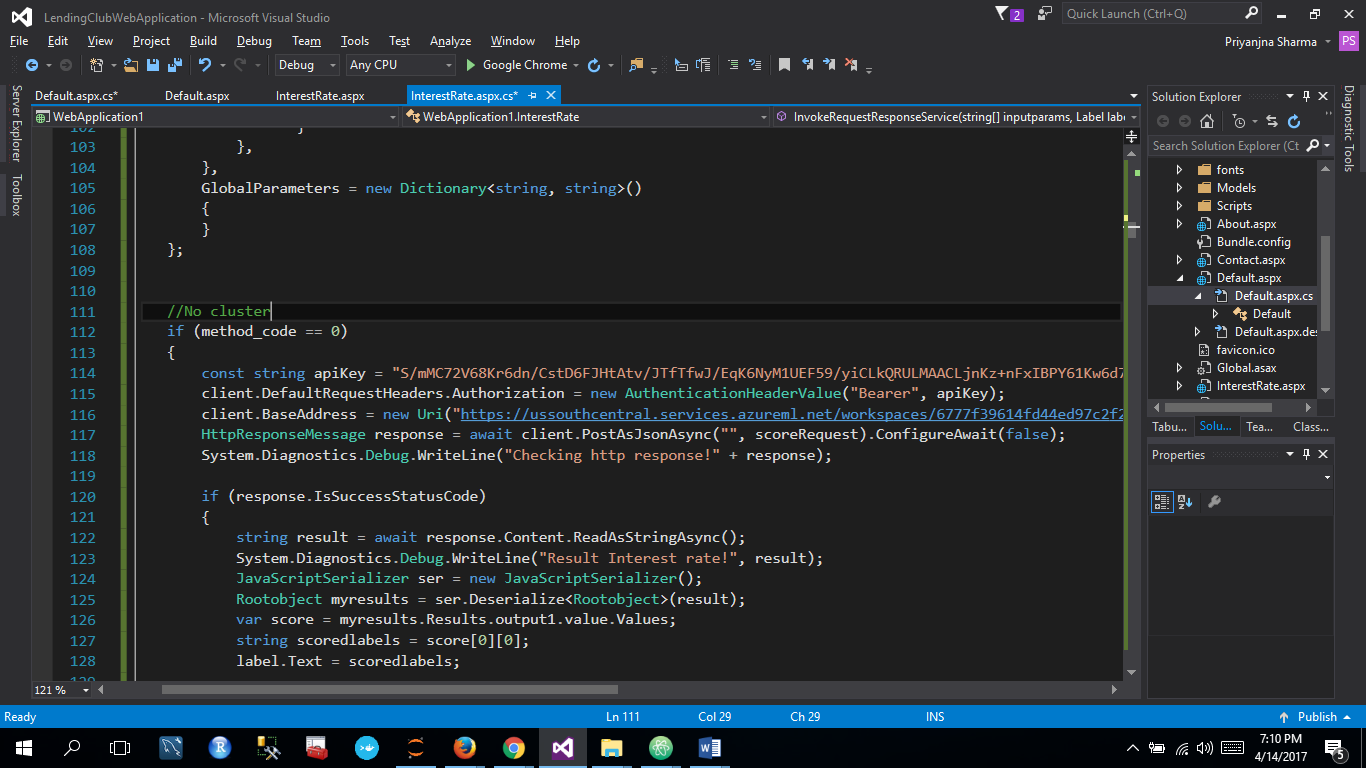
For Prediction: it accesses the REST API of Random Forest Algorithm for Prediction and gives the predicted interest rate to the user.

**Calling the REST API**

1. Defining the input columns and values:



1. Defining the api key and client base address and getting the response



## CLASSIFICATION

This is the first step of our application. This step takes the below mentioned parameters as the input and classifies if the loan can be approved or not.

The input parameters are supplied to REST API of our best classification model i.e Random Forest Classification

### Inputs to REST API

#### Loan Amount

The loan amount required by borrower

#### DTI – Debt to Income Ratio

The current Debt to Income ratio value of the borrower

#### State

State that the borrower resides in

#### Length of Employment

Number of years for which the borrower has been employed

#### Credit Score

Current Credit Score of the borrower (FICO score)

## PREDICTION

### Inputs to REST API

### *Term*

Term can be either 36 months or 60 months as shown in the dropdown.

### *Grade*

Grade as given by Lending Club ranging from A to G

### *Purpose*

Purpose of the loan

* debt\_consolidation
* credit\_card
* home\_improvement
* small\_business
* major\_purchase
* medical
* car
* moving
* house
* vacation
* renewable\_energy
* wedding
* educational

### *Inquiries in Last 6 months*

Number of inquiries made by the borrower in last 6 months

### *Public records*

The number of public records that the borrower has

### *Loan Status:*

Created a Loan\_Status\_Binary which passes the result to the REST API:

If loan status is in:

* "Charged Off",
* "Default",
* "Does not meet the credit policy. Status: Charged Off",
* "In Grace Period",
* "Default Receiver",
* "Late (16-30 days)",
* "Late (31-120 days)"]

Then set value of input parameter as 0 else 1

### *Home Ownership:*

Created home\_ownership\_binary which passes the result to the REST API:

* “Rent”
* “Mortgage”
* “Own”
* “Other”

If it lies in “Rent” or “Other” then set value of input parameter 0 else 1

### *Verification Status:*

Created a column: Verfication\_status\_binary which passes the result to the REST API where:

If verification\_status can be:

* Not Verified
* Verified

If verification status is “Not Verified” then verification status binary is 0 else 1

### *Application Type*

The values in application\_type can be:

* Direct\_Pay
* Individual
* Joint

If application type is “Direct\_Pay” then Application Type Binary is 0 which is passed to the REST API

If application type is “Individual” then Application Type Binary is 1 which is passed to the REST API

If application type is “Joint” then Application Type Binary is 2 which is passed to the REST API

### *Loan Amount*

We converted loan amount to a loan amount range by categorizing it into the following categories. The final output to the REST API for prediction is Loan Amount Category which is handled at the back end.

|  |  |  |
| --- | --- | --- |
| loan\_amnt\_category\_code | loan\_amnt\_From | loan\_amnt\_To |
| 0 | 500 | 5000 |
| 1 | 5000 | 7000 |
| 2 | 18550 | 22000 |
| 3 | 9450 | 10750 |
| 4 | 15000 | 18550 |
| 5 | 22000 | 28000 |
| 6 | 28000 | 40000 |
| 7 | 12950 | 15000 |
| 8 | 10750 | 12950 |
| 9 | 7000 | 9450 |

### *Annual Income*

We converted annual income to a annual income range by categorizing it into the following categories. The final output to the REST API for prediction is Annual Income Category which is handled at the back end.

|  |  |  |
| --- | --- | --- |
| annual\_inc\_category\_code | annual\_inc\_From | annual\_inc\_To |
| 0 | 74600 | 85000 |
| 1 | 85000 | 100000 |
| 2 | 57000 | 65000 |
| 3 | 50000 | 57000 |
| 4 | 125000 | 9573072 |
| 5 | 0 | 35000 |
| 6 | 100000 | 125000 |
| 7 | 65000 | 74600 |
| 8 | 42000 | 50000 |
| 9 | 35000 | 42000 |

## Result 1: Manual Clustering

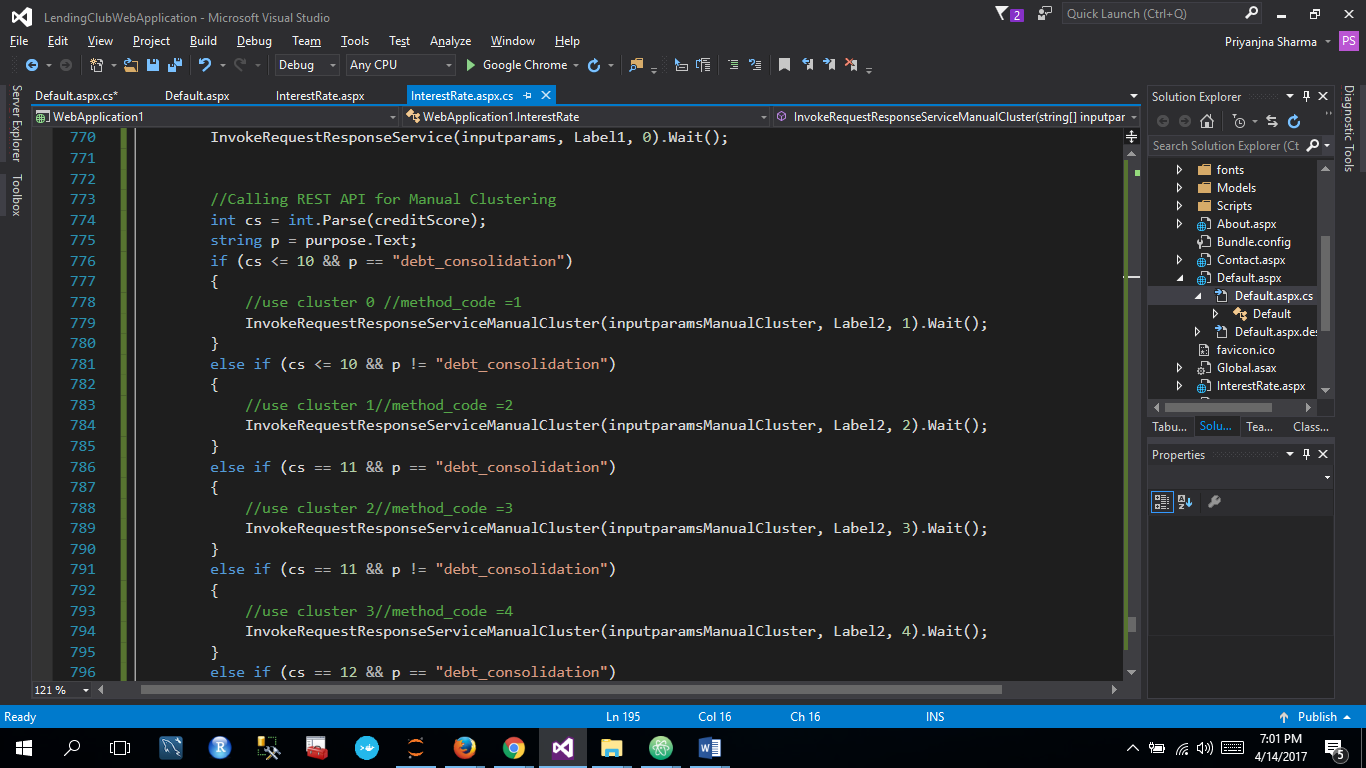
For Manual Clustering, we have 8 clusters based on

* Credit Score of the borrower and
* the Purpose selected by the borrower

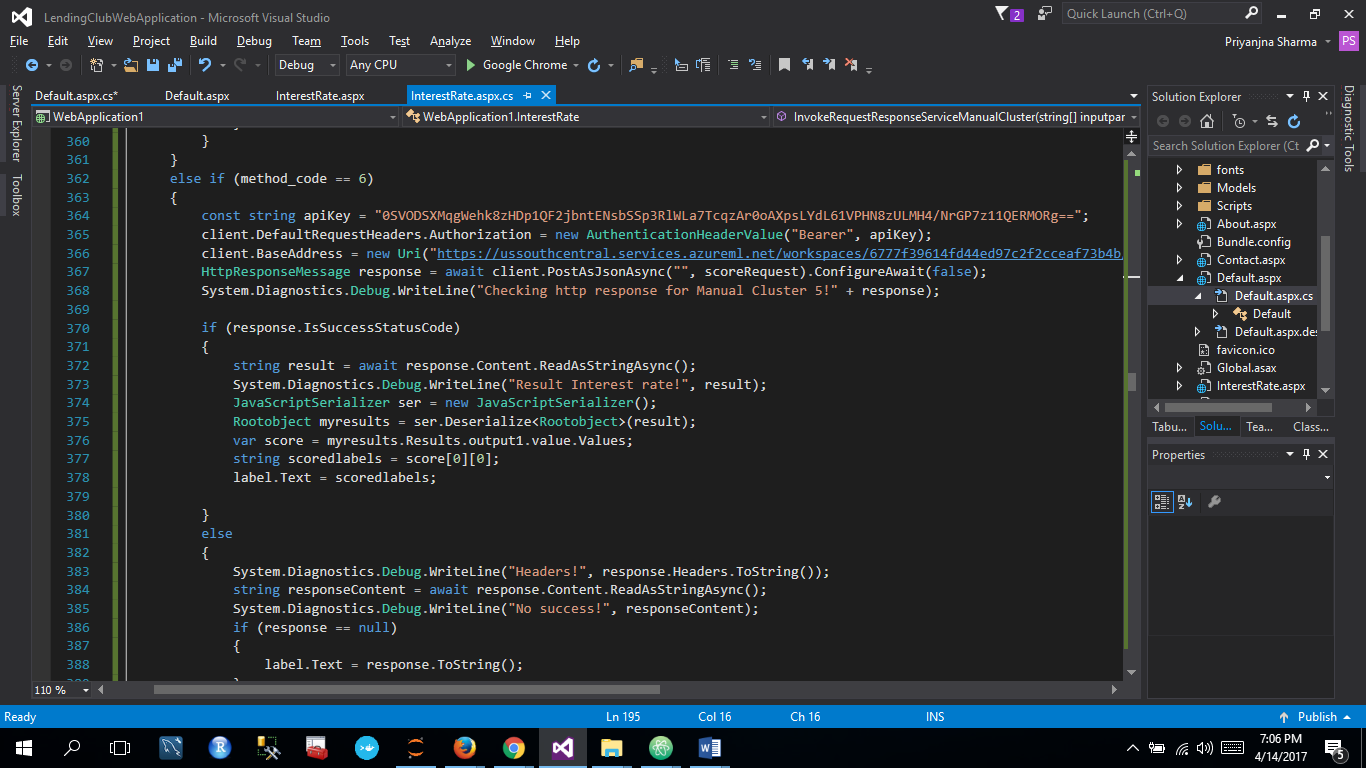
as defined above in Manual Clustering Topic.

We take the above parameters as input and find the appropriate cluster they lie in. After getting the cluster, send the parameters to REST API of the corresponding cluster.

* Finding the appropriate cluster



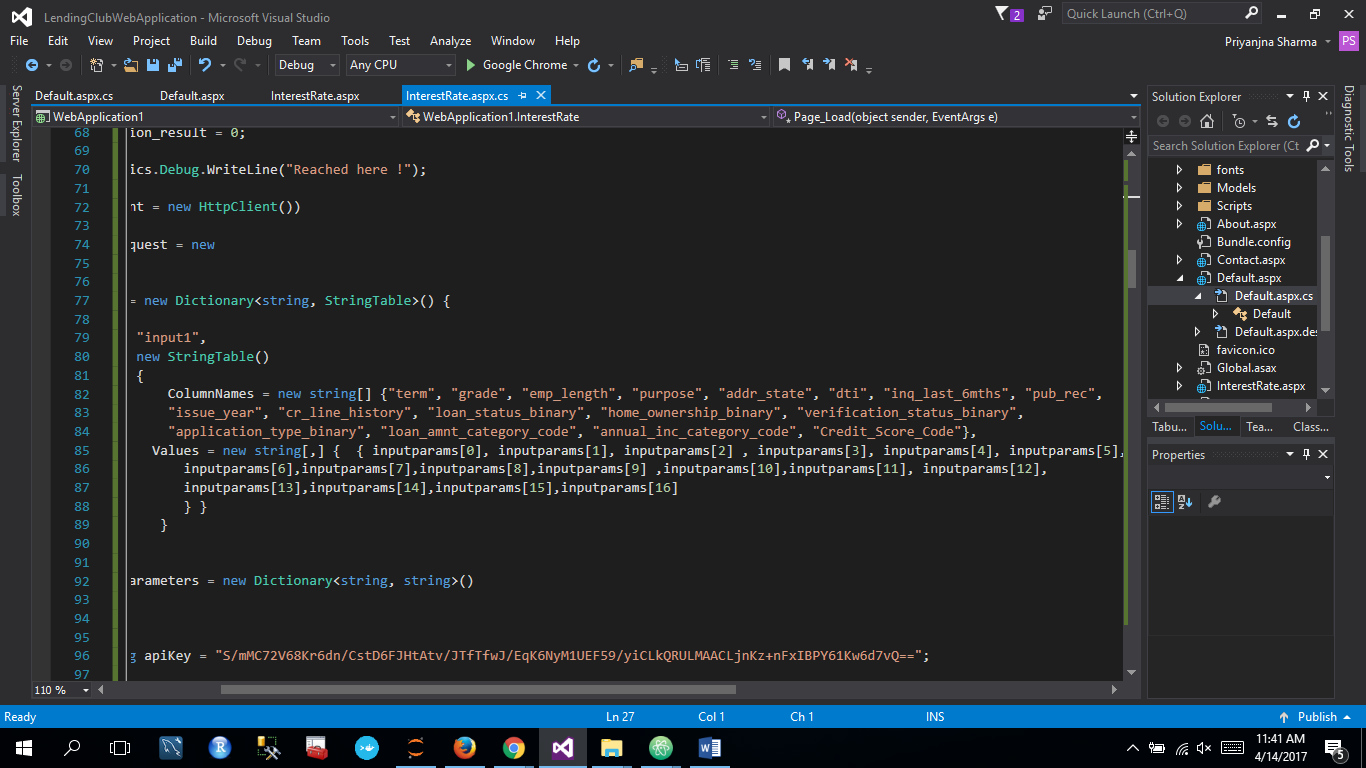
* Calling the appropriate REST API



## Result 2: K-means Clusters

## Result 3: No Clusters

* Passing all the input parameters to the corresponding REST API and getting result.



* Handling the response from API call

## 

|  |
| --- |
|  |

# Contribution

# 

# REFERENCES

1. <https://en.wikipedia.org/wiki/Lending_Club>
2. <https://www.lendingclub.com/info/download-data.action>
3. <https://www.credco.com/assets/pdfs/datasheets/FICO-booklet.pdf>
4. <http://www.lendingmemo.com/average-investor-return-lending-club-dropping/>
5. https://your.vantagescore.com/interpret\_scores