**REPORT**

**INFO 7390 Assignment 2 – Fall 2017 (Lending Club dataset)**

**Professor - Sri Krishnamurthy**

**By - TEAM 03**

**Sonali Chaudhari**

**Madhumathi Prakash**

**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.

**Part 1: Data wrangling and exploratory data analysis 1(a)Data Download and pre-processing:**

**Data Download:**

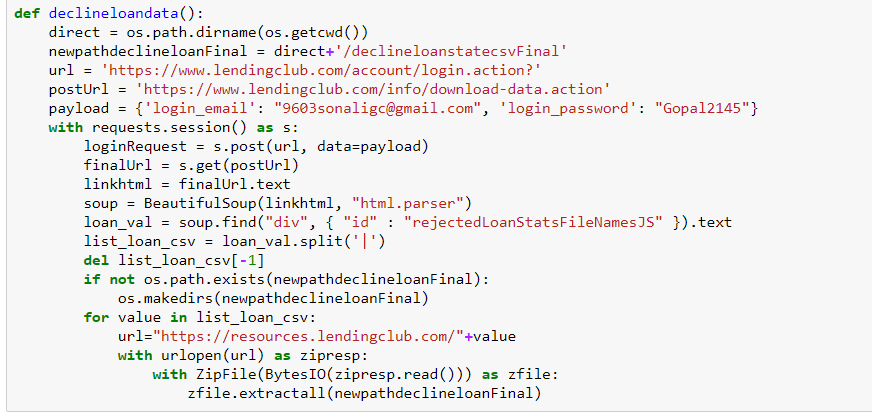
**Steps:**

**Scraping Data from lending Club**

Since the lending club has mode data when signing in, we coded it so there is a login and password. We used requests.session() to access that session we logged into.



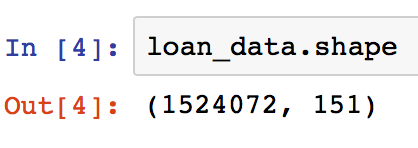
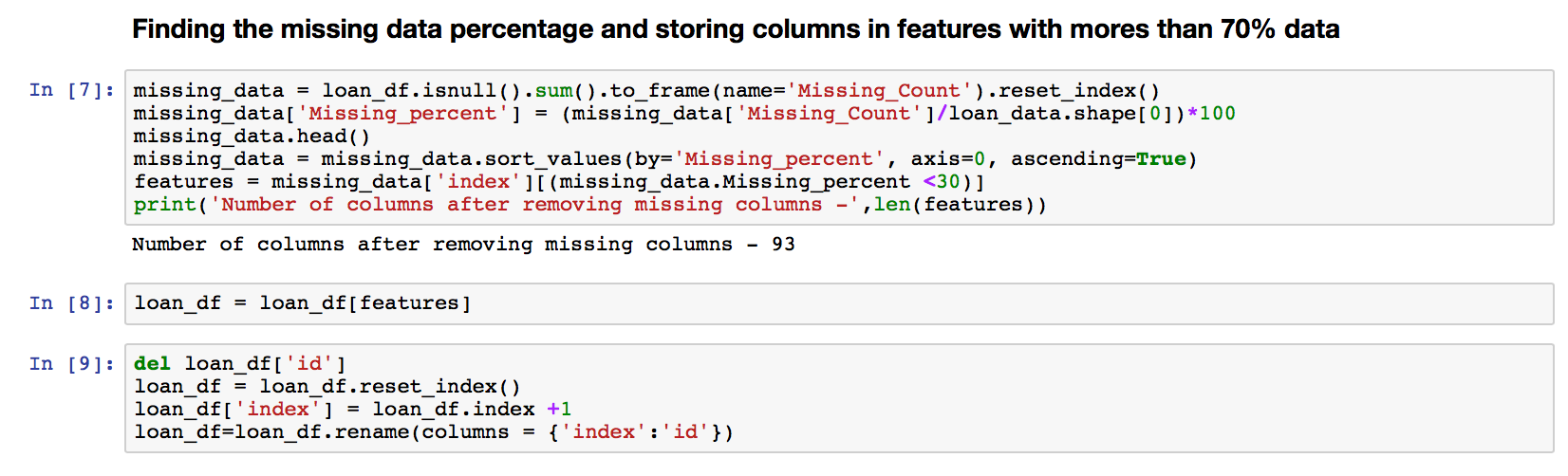
We used beautiful soap to find the hidden class with id of loanStatsFileNameJS and same thing but with rejectedStateFileNameJS.

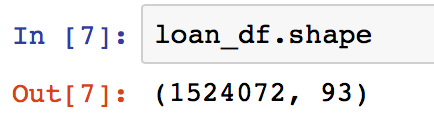


**Data Preprocessing:**

**Loan Data**

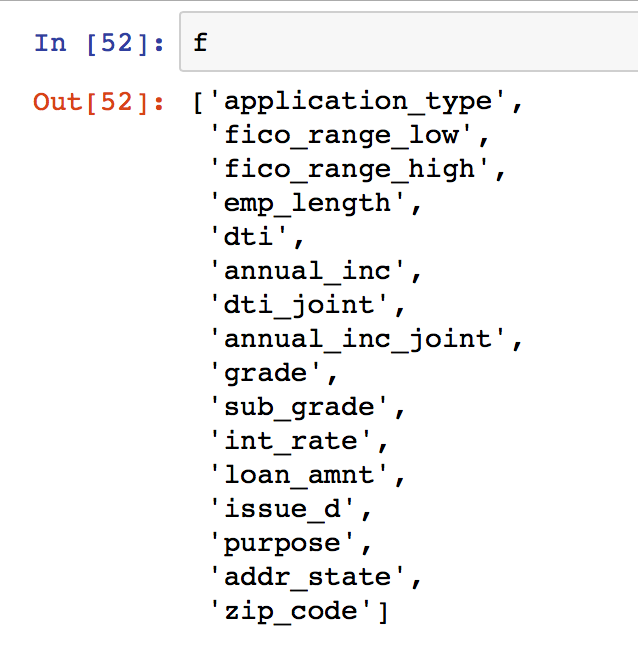
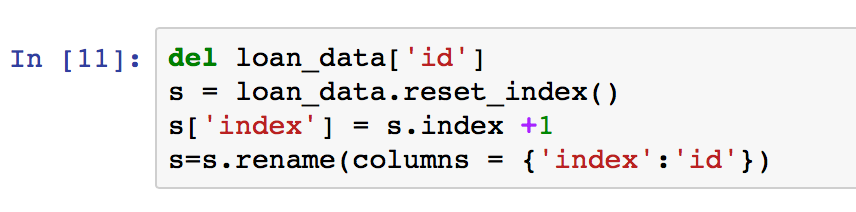
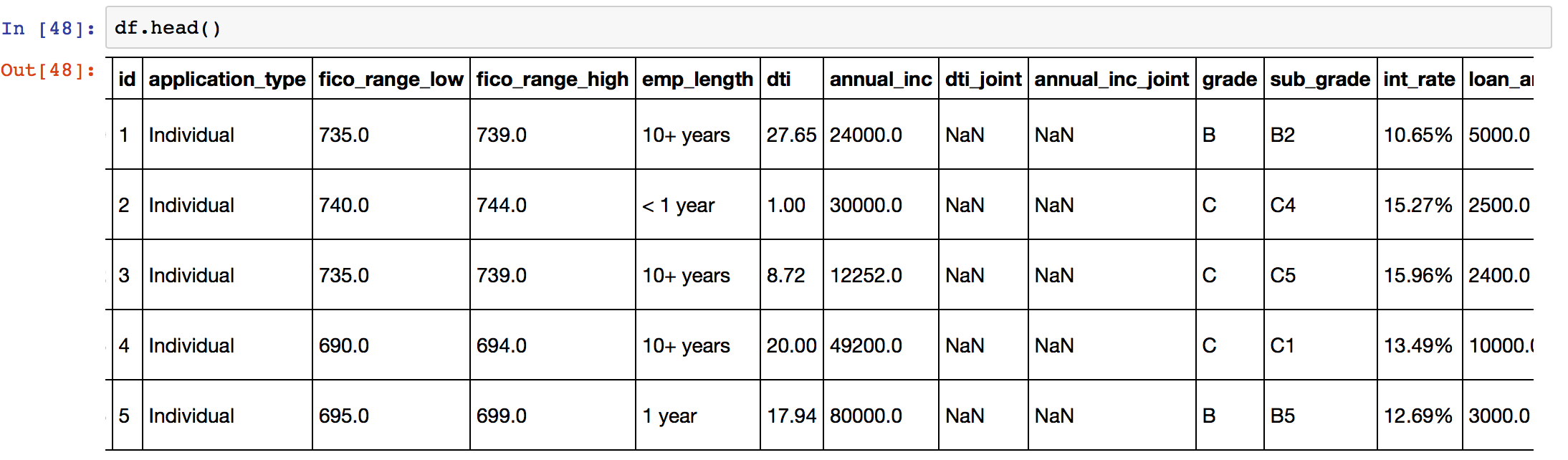
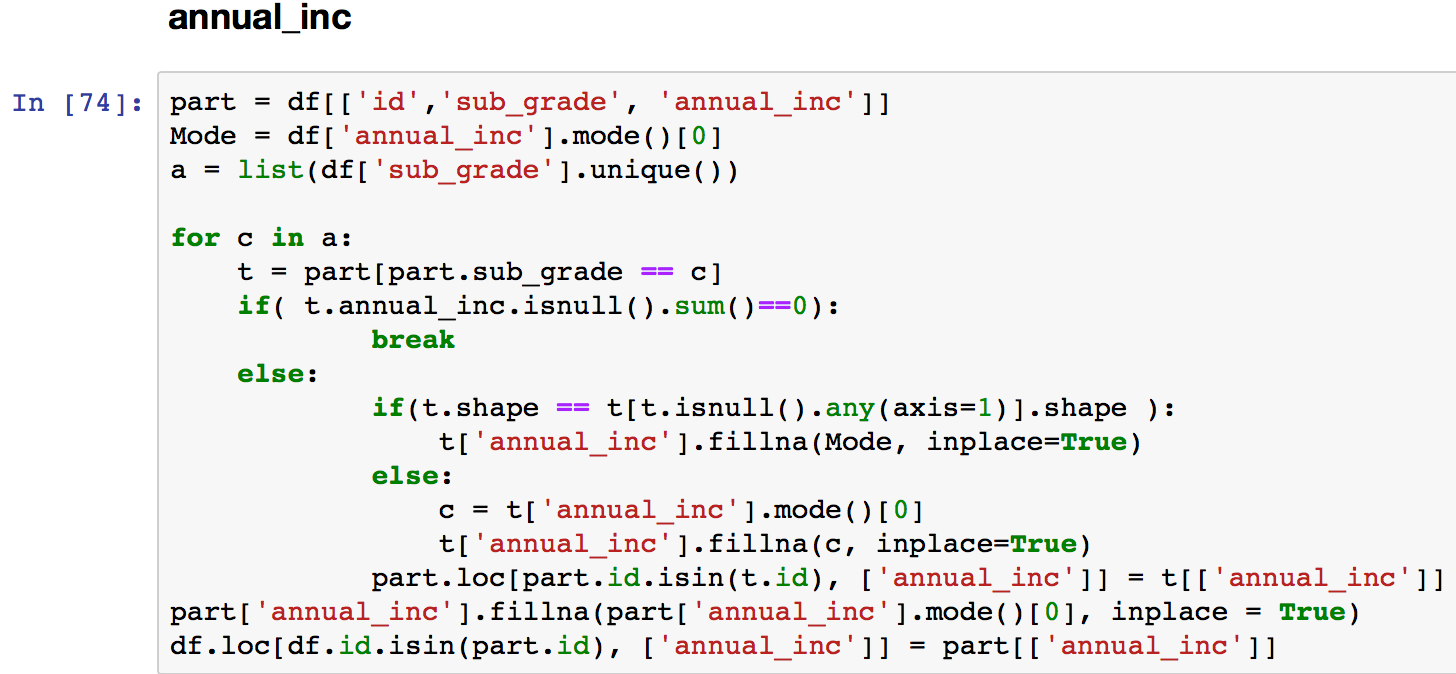
Steps:

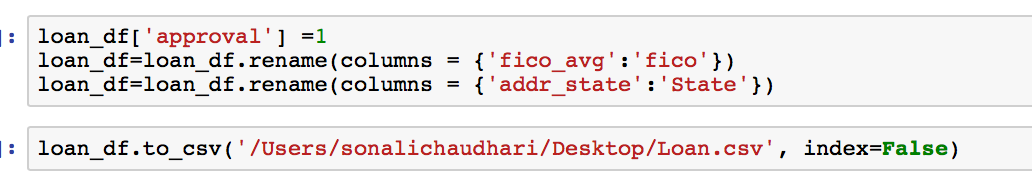
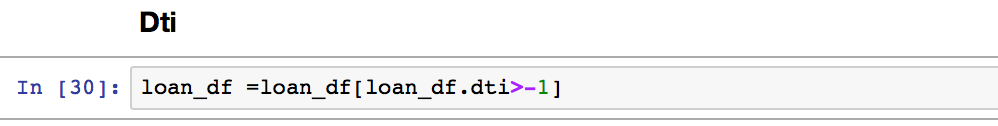
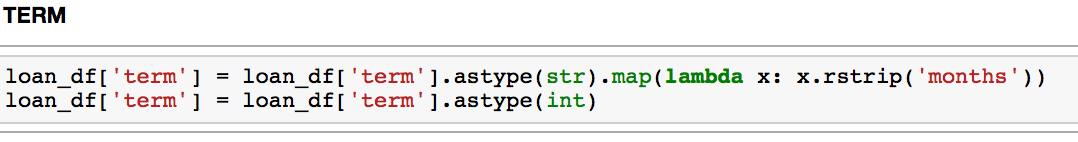
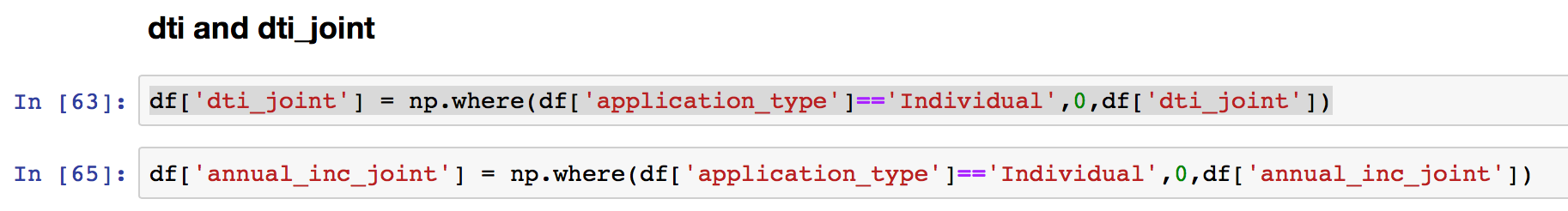
1. Loaded each programmatically downloaded CSVs in separate dataframe.
2. Concatenated each dataframe in a single dataframe ‘loan\_data’.
3. Going through each column and understanding the meaning of each column through the dictionary provided on the LC website ‘LCDataDictionary.xlsx’
4. Getting rid of the columns have more than 70% missing data. During this process we get rid of the column ‘id’ which has all rows null.



1. Out of the 151 columns most of the columns are generated after the loans application procedure has taken place. So these columns leak data from the future. We need columns data that potential customer provided to LC during their application.

**Feature Engineering**

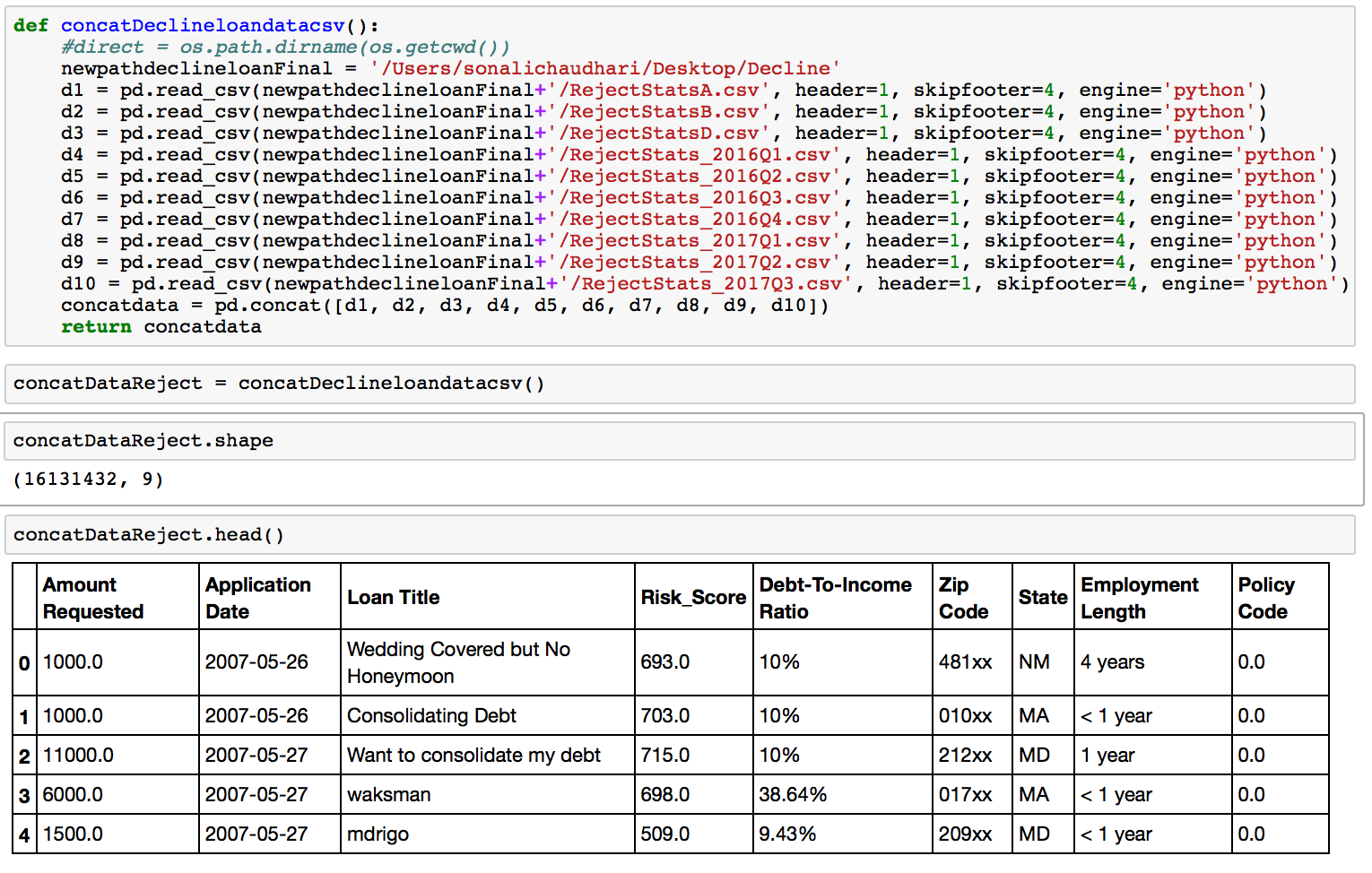
1. Created a list ‘f’ a features that are important for and received during initial borrower’s application.
2. Created a dataframe ‘df’ consisting of all the features in the list ‘f’ and generated a column ‘id’ for uniquely identifying each entry.
3. Retrieved the count of nulls in each column in ‘df’ to deal with missing data in every column.
4. Getting rid of the row having most of the columns null.
5. Getting the dataframe to have consistent formats for each column entry and logically replacing the NaN values for some categorial columns
   1. The columns like emp\_length, int\_rates, zipcode, dti… are of type object and has no consistent format
   2. Replaced dti\_joint, annual\_inc\_joint null values to ‘0’ for the application\_type ‘Individual’ since those application has no co-borrower and hence dti\_joint, annual\_inc\_joint becomes 0.
   3. Filled the remaining null values for annual\_inc with the mode annual\_inc value in a particular sub\_grade column that the row belongs to.
   4. Similarly for dti\_joint column
   5. There are columns ‘fico\_range\_high’ and ‘fico\_range\_low’; so generated a new column ‘fico\_avg’ having the average of both the column values
   6. Change the datatype of necessary columns and adding new column approval having value 1 for the loan data.

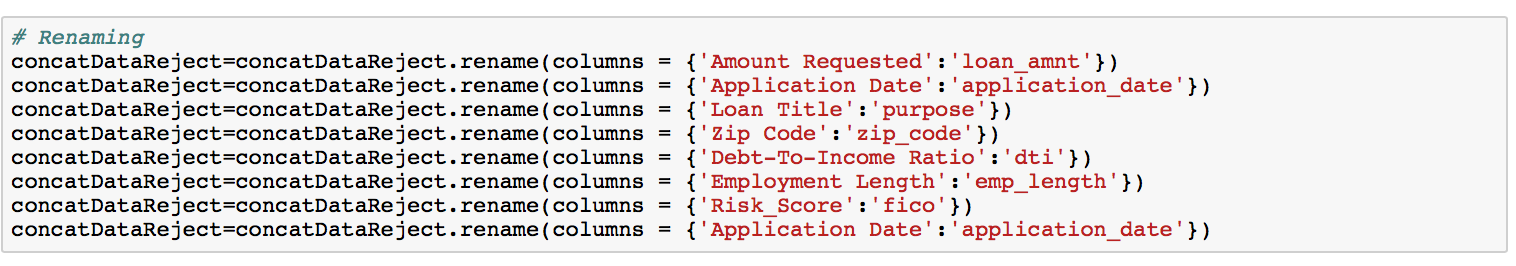


**Decline Data**

Steps:

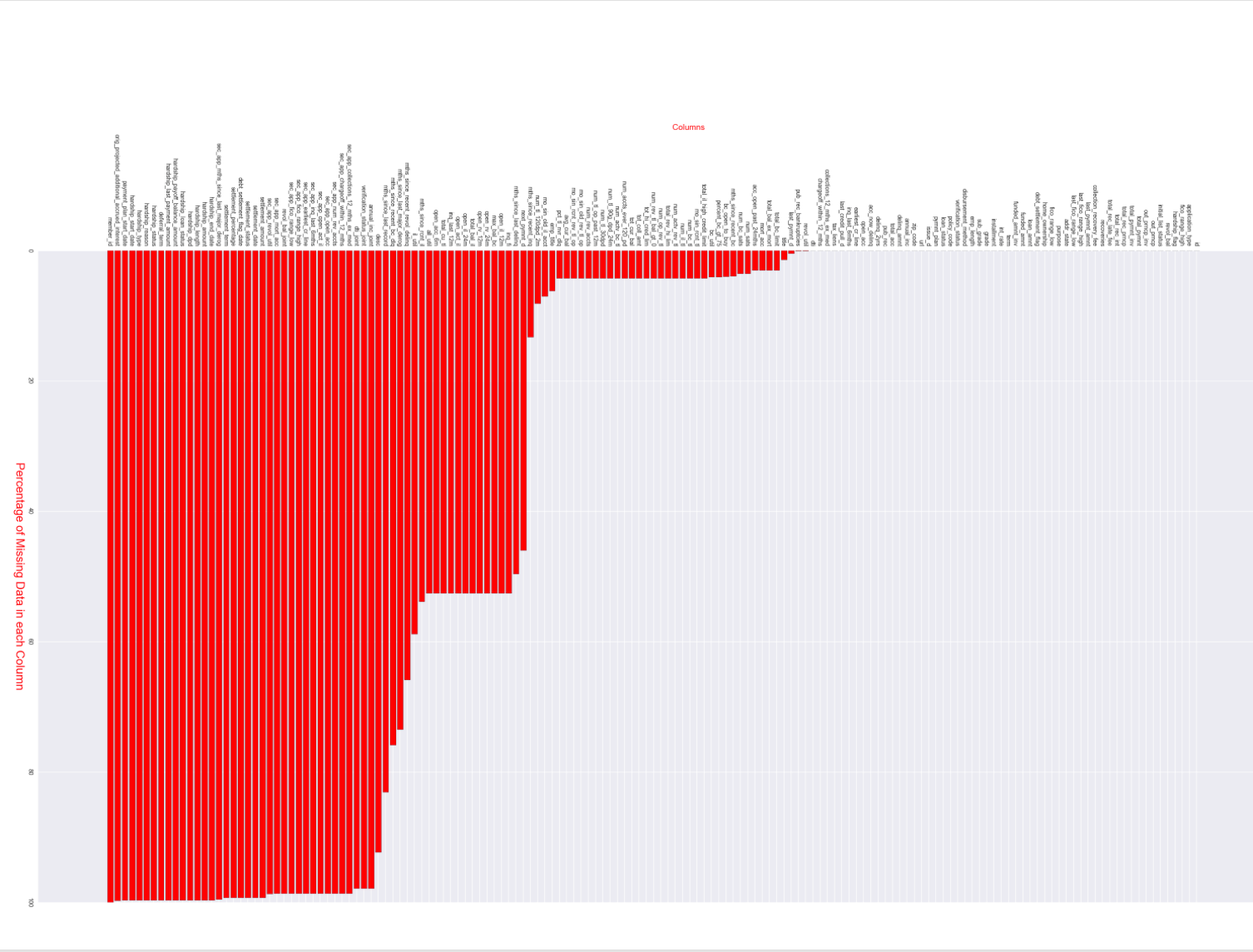
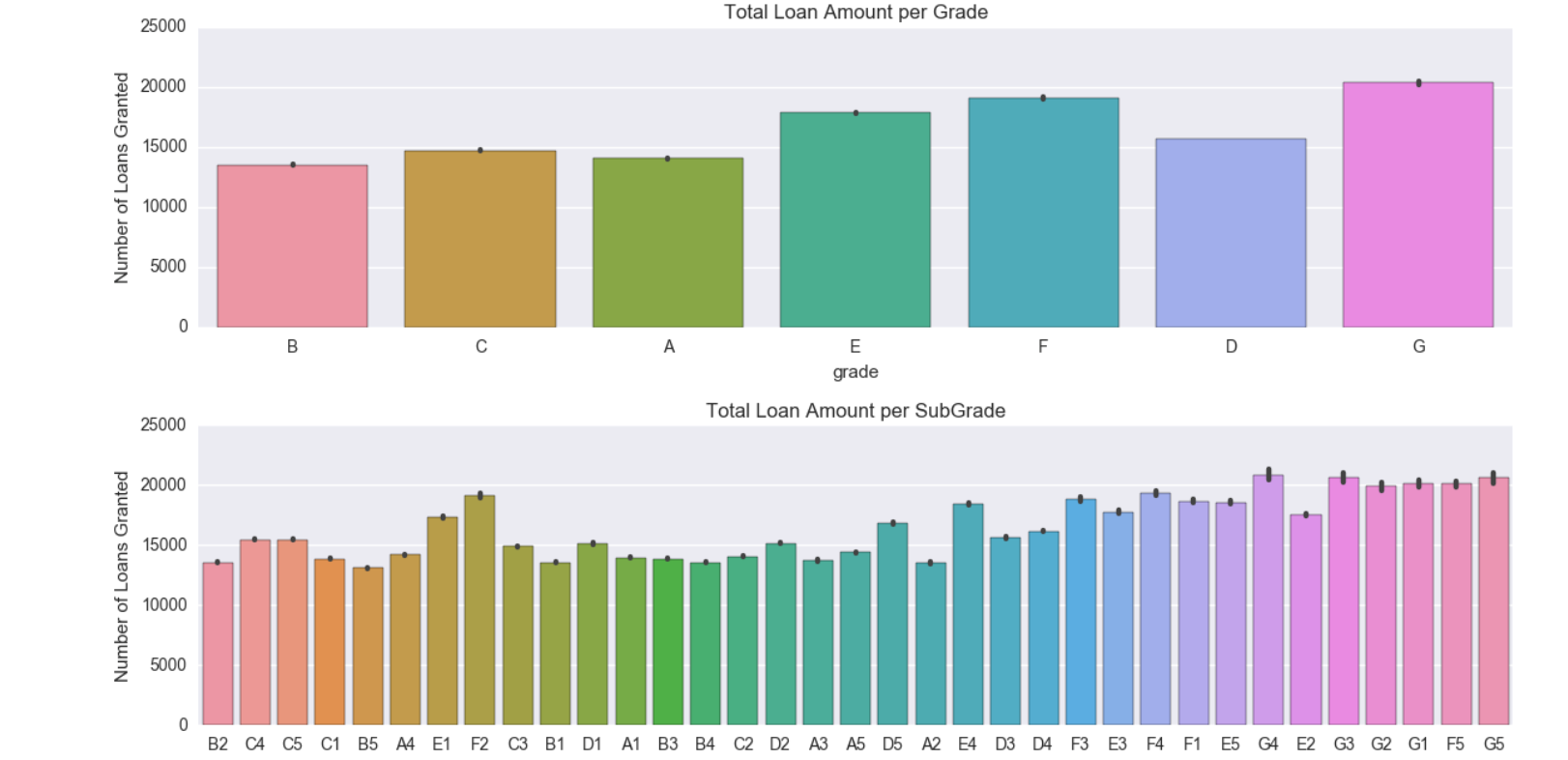
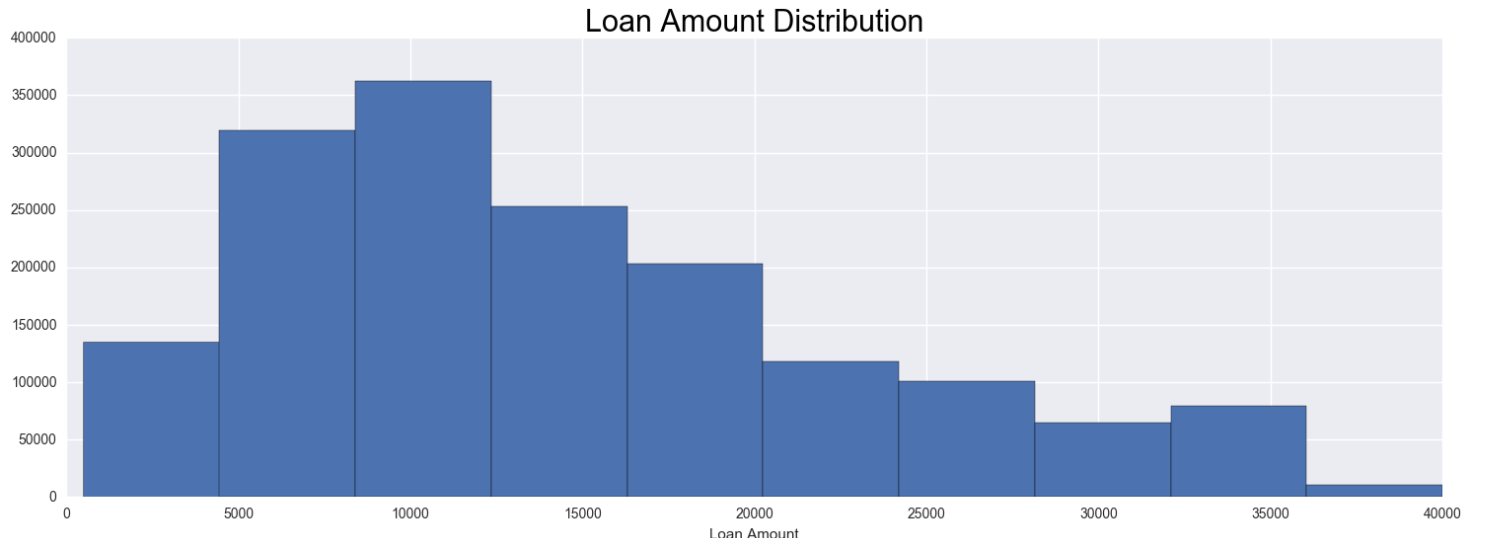
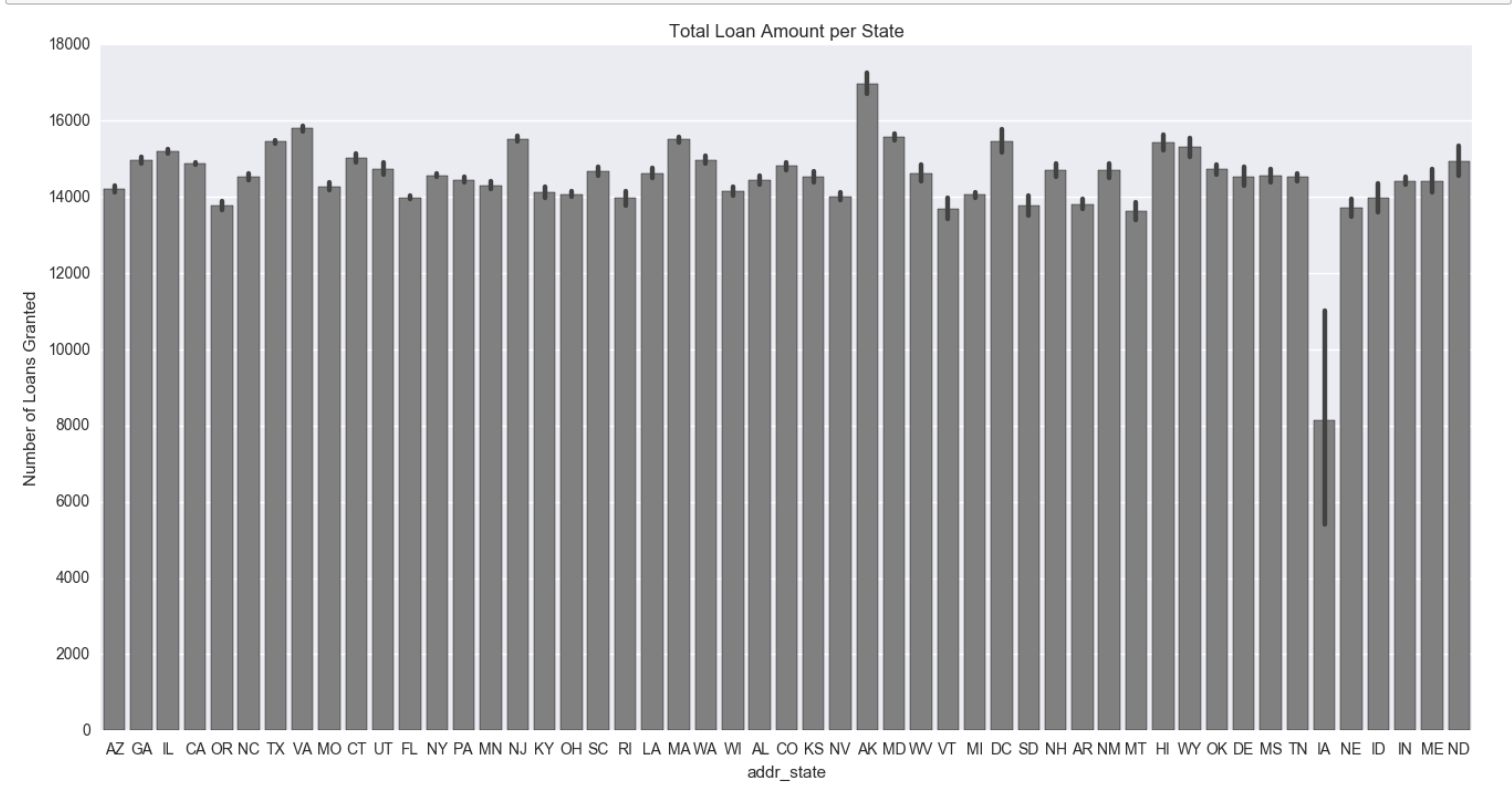
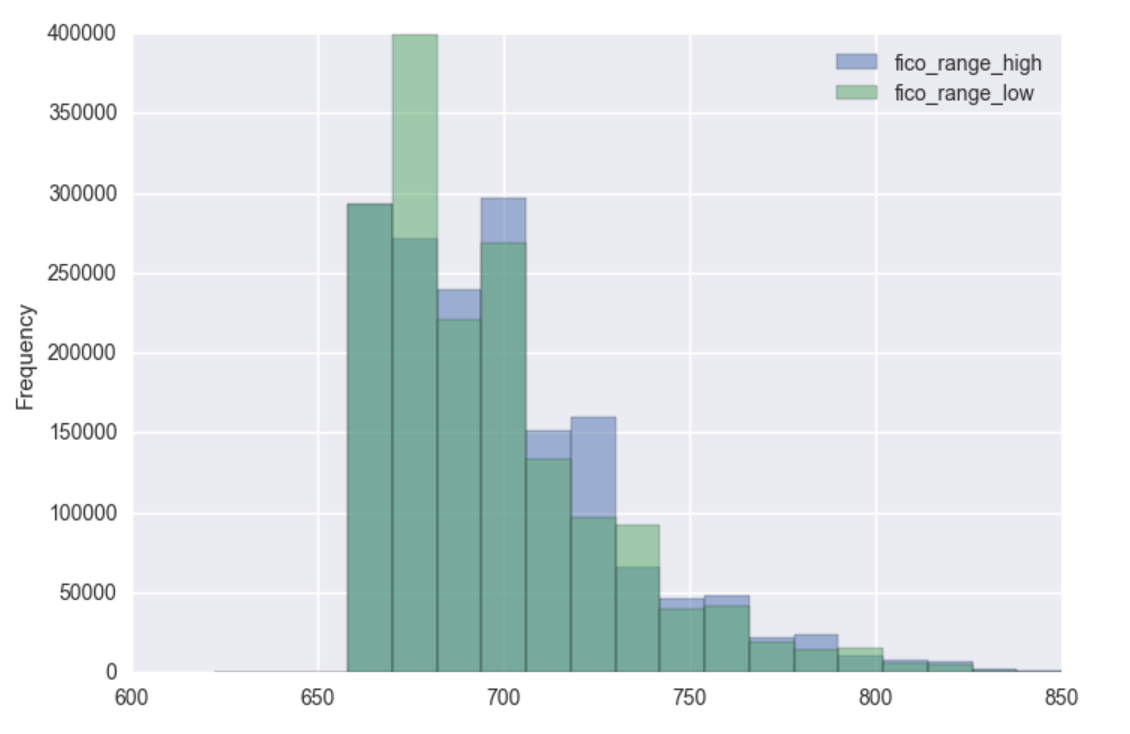
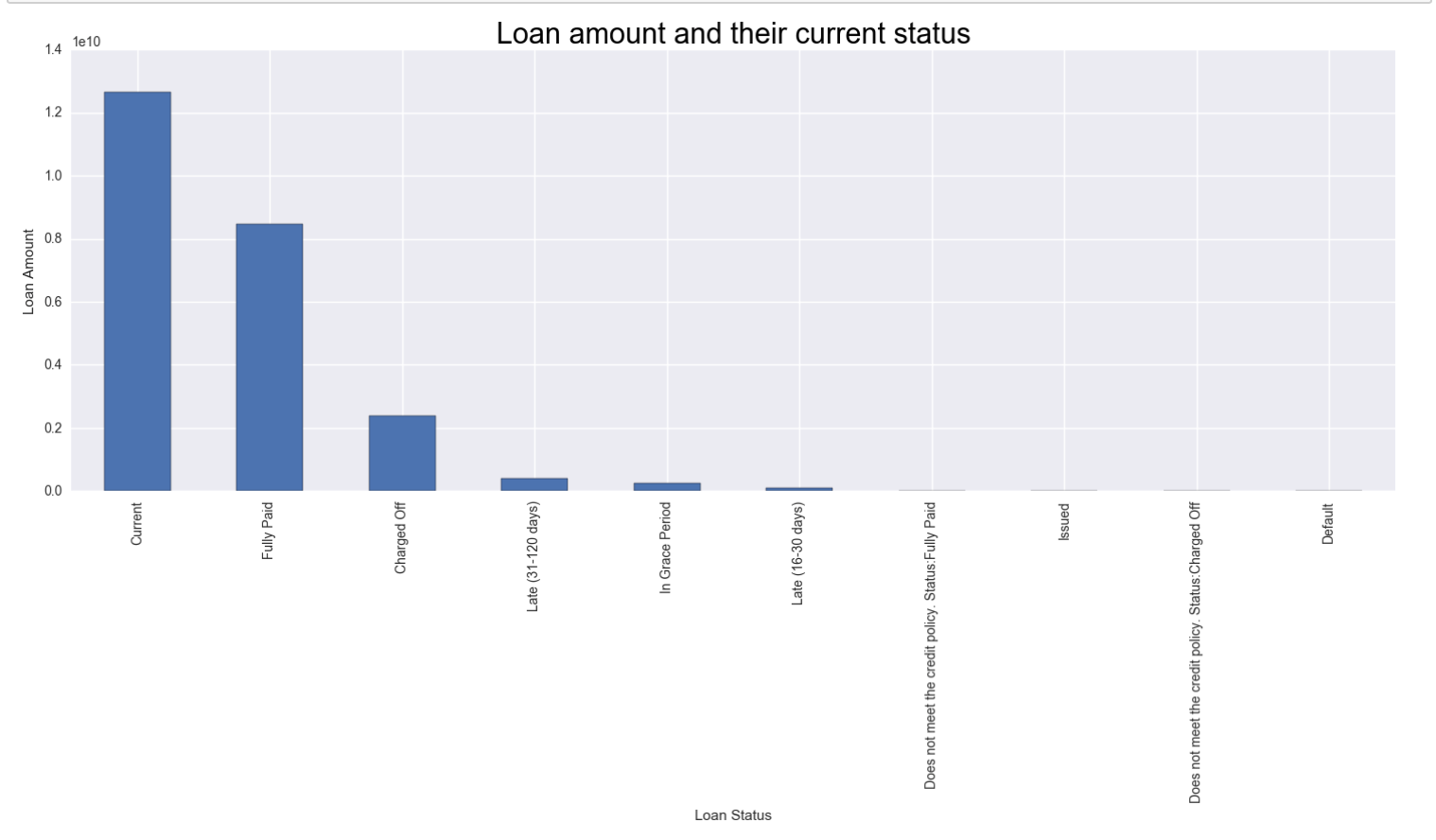
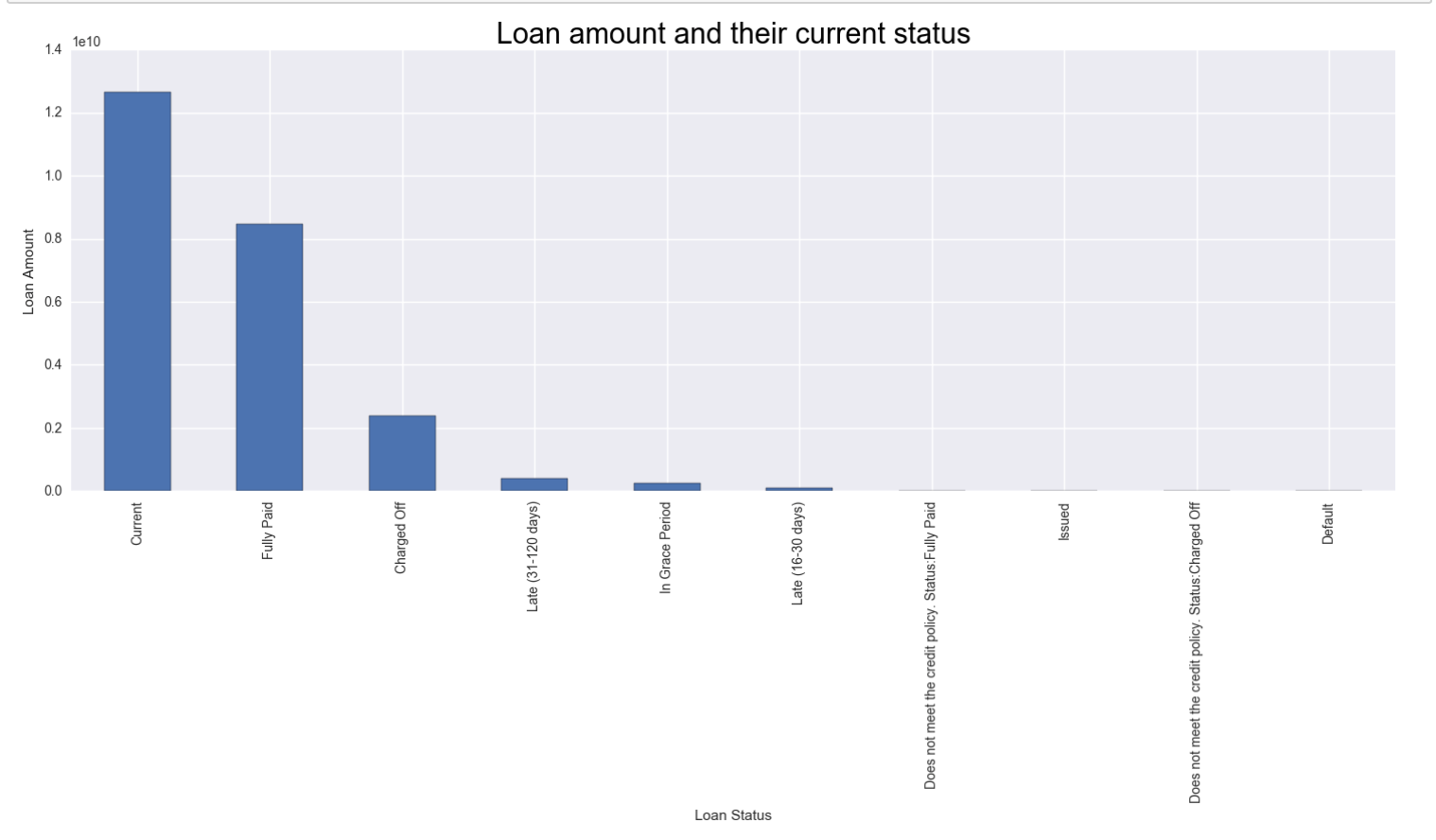
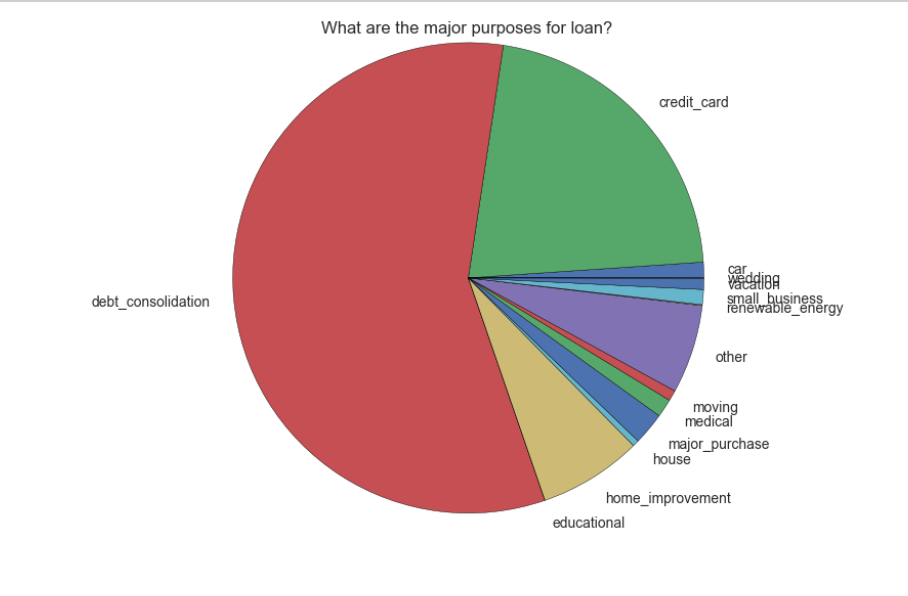
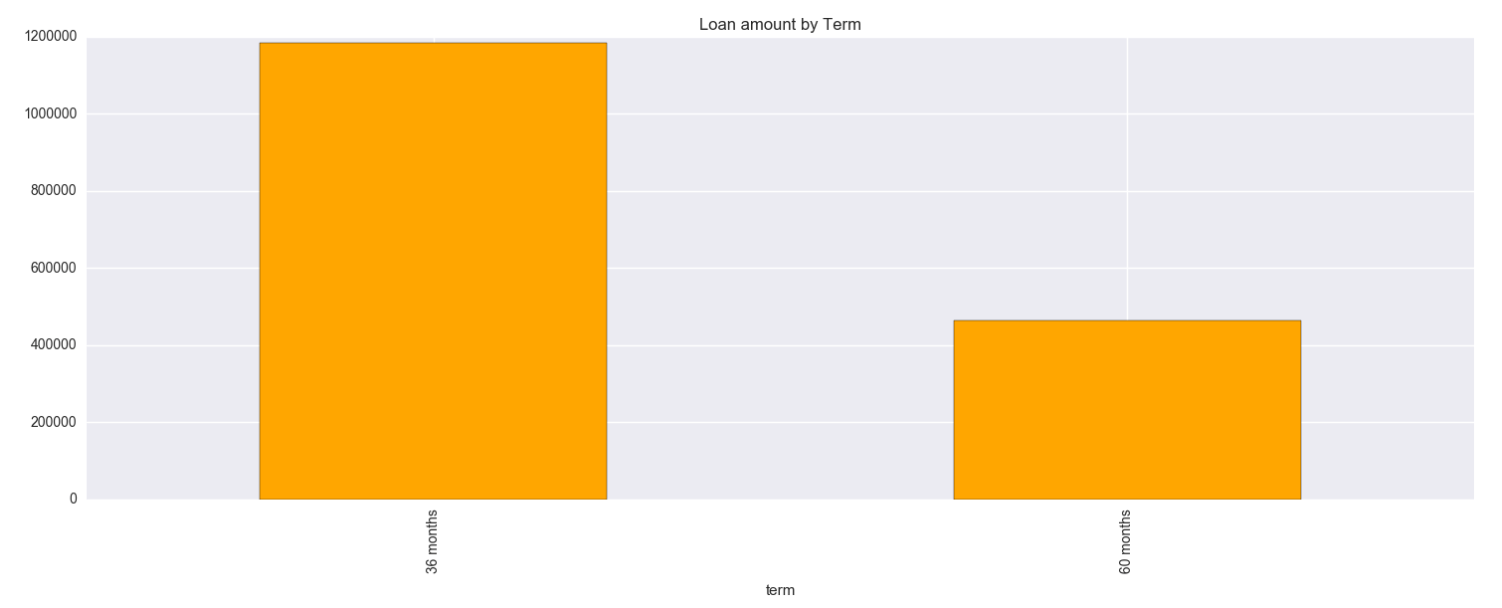
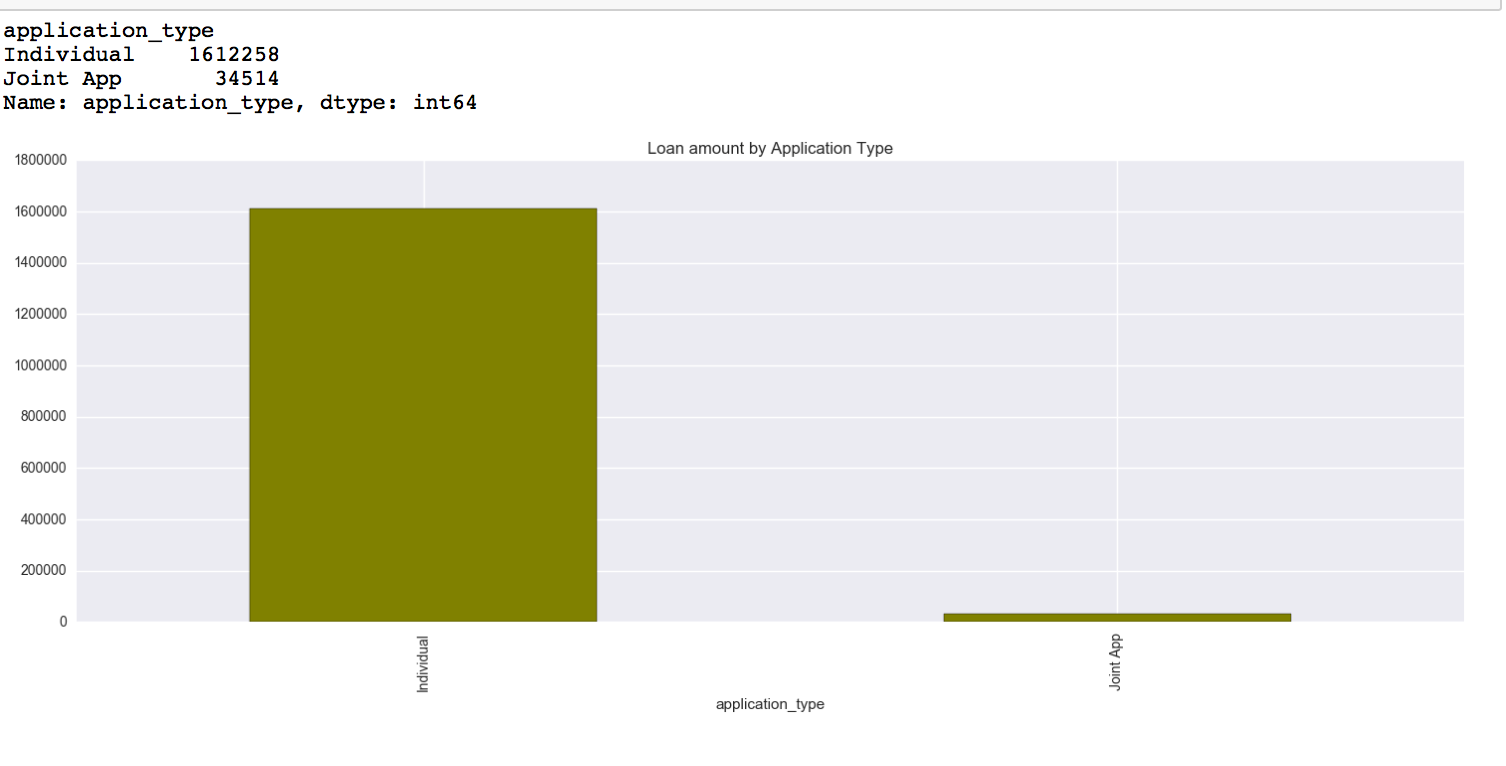
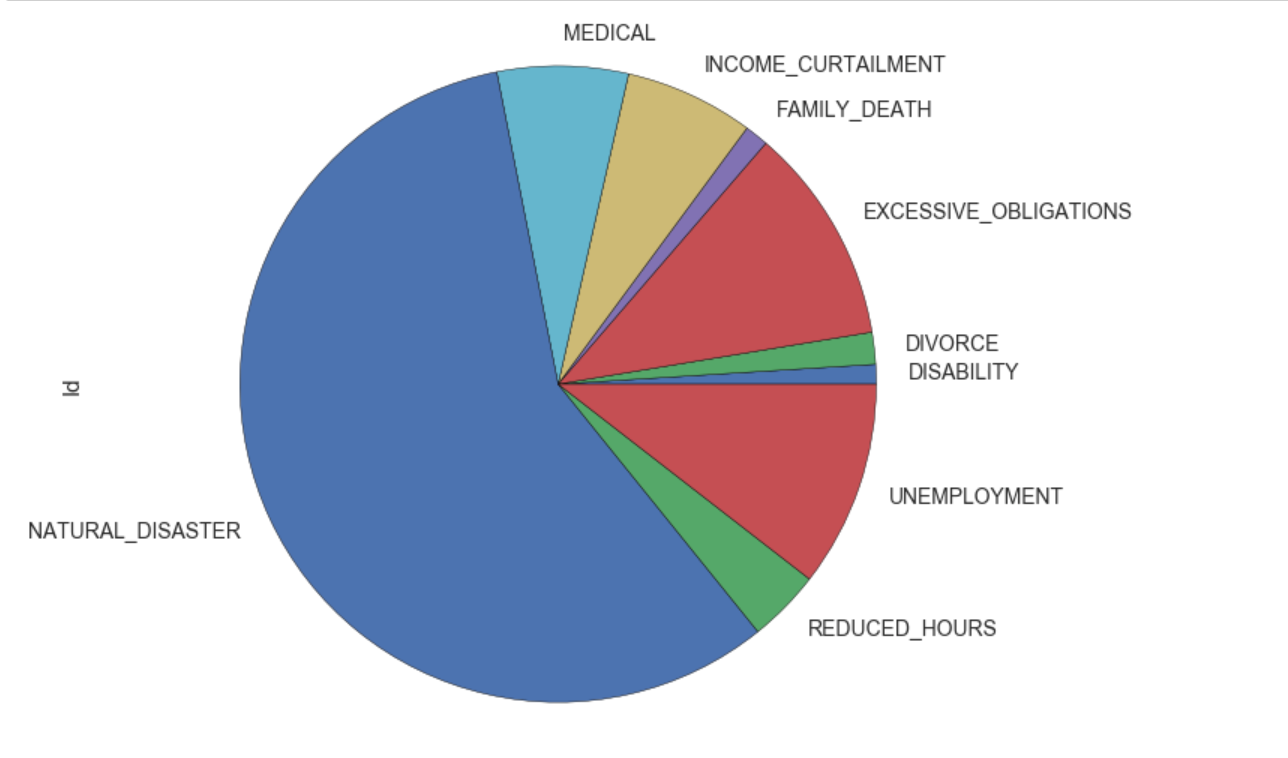
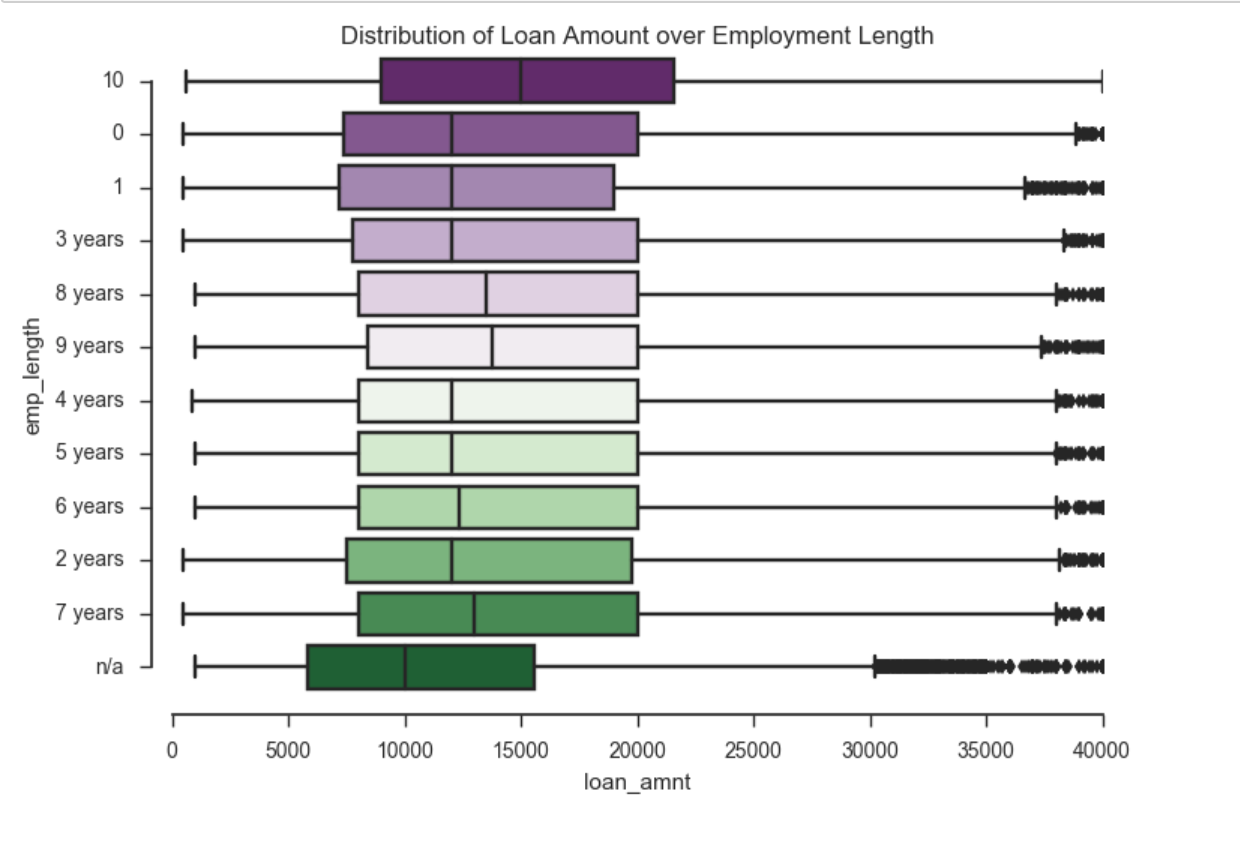
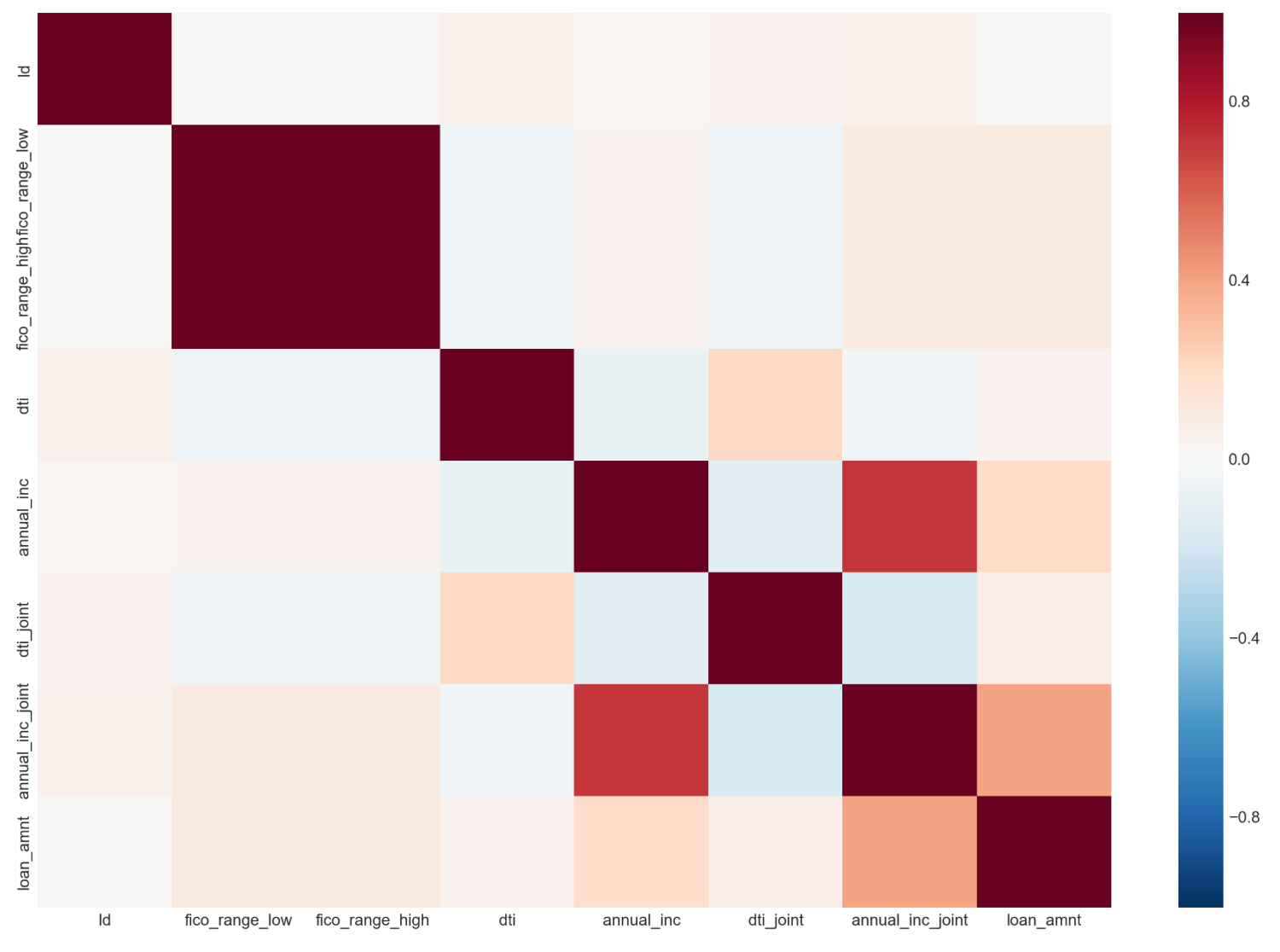
1. Loaded the csvs in each data frames and then concatenating it one data frames.
2. Clean the data
3. Changing the column names and datatype
4. Adding new column ‘approval’ having values ‘0’ for the decline data
5. Exporting to Csv



**1(b)Exploratory Data analysis:**

We performed exploratory data analysis the loan data and decline data separately and few EDAs for combine data.

**Loan data**

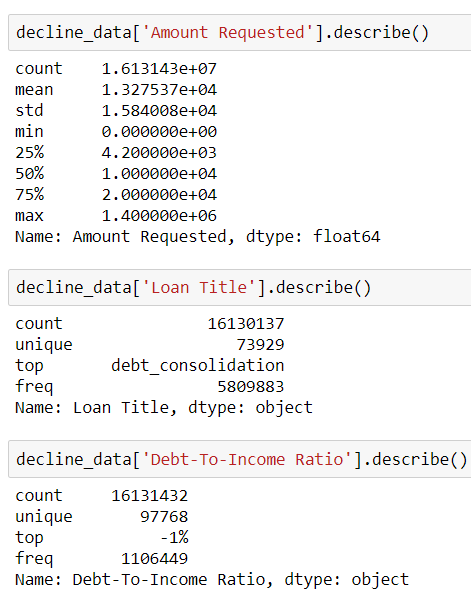
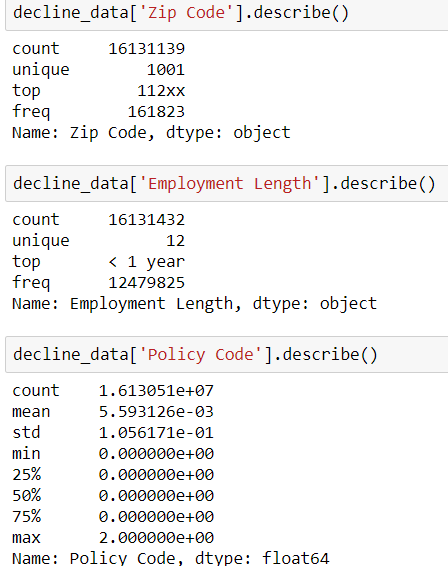
1. Missing data Analysis
2. How many loans have been approved per grade and sub grade?
3. Distribution of Interest Rate
4. Distribution of Loan Amount
5. Loans Granted per State
6. Zico high and fido low analysis
7. Loan Amount by loan status
8. Distribution of Loan amount over purpose
9. Distribution of Loan amount over term
10. 
11. Distribution of Loan amount over application type
12. Loan Amount count per Hardship Reason
13. Distribution of Loan Amount over Employment Length
14. Heat Map of columns details provided during application for Loan

**EDA for Rejected Loans**

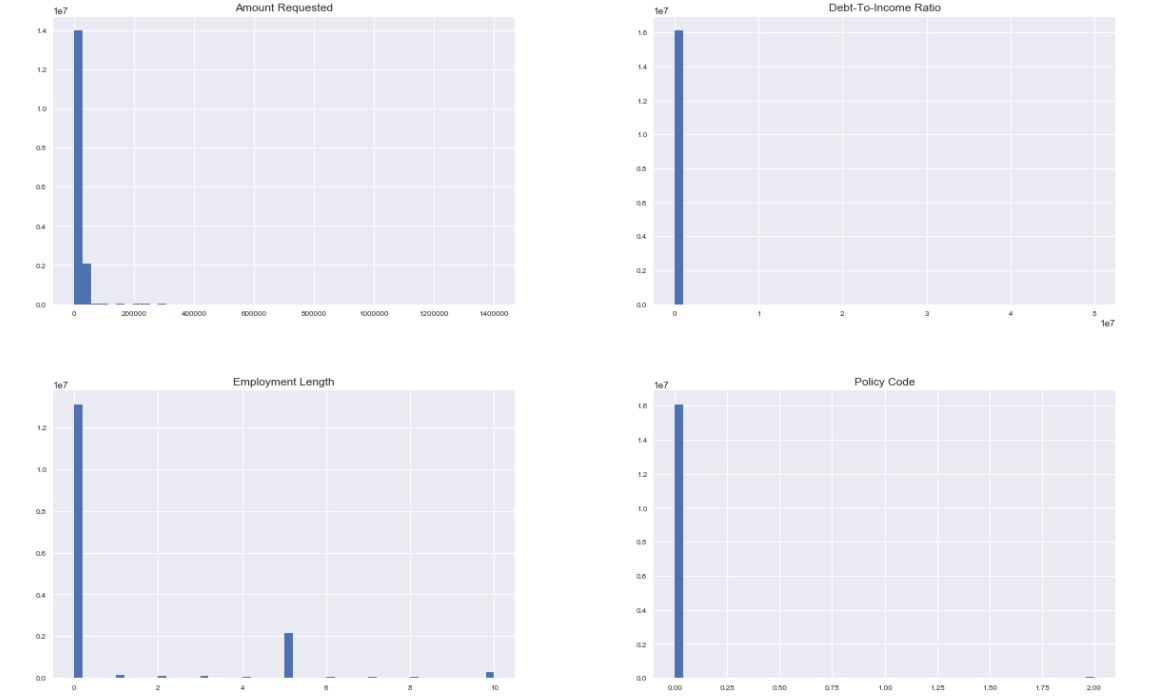
Looking at the rejected data, it is clear that there are only 9 columns. We are finding all the missing data above 70% here and see that only risk score has more than 70% missing data.

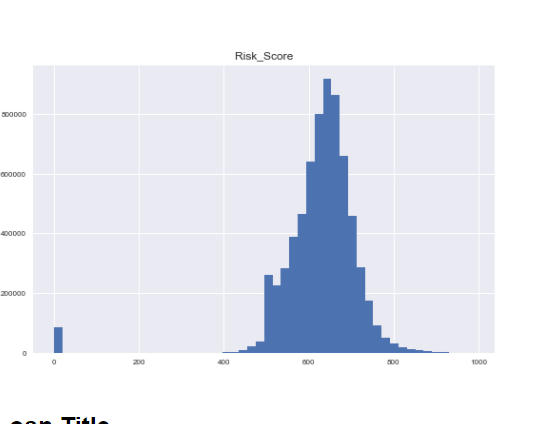


We have then dealt with looking at description of each of important columns to see the min, max, mean , and mode.



We then verified and checked the distribution of the columns and the data to see if there are a lot of outliers and how skewed the data was.

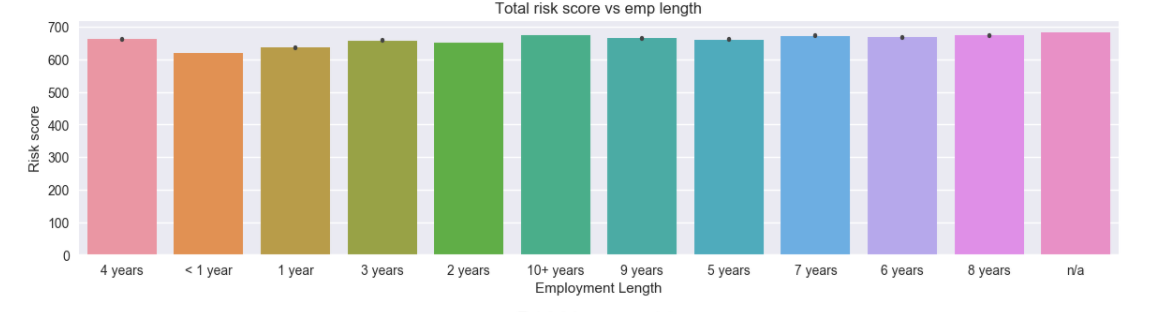




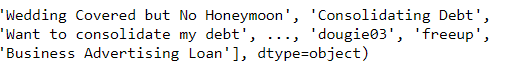
We compared the state and to the risk score



We then compared the employee length to the risk score.



We also verified all the unique values of the loan title to see the types of loans people were taking out.



**1(c)Luigi Scripts:**

Luigi makes it easier to automate the pipeline.

I created a py file. The luigi script has 2 files and 2 classes in each it has a run and output methods. One class is for rejected and another for loan accepted.

In Loan Luigi we have one class for parsing web and another for dealing with missing data and feature selection.

**Part 2: Building and evaluating models**

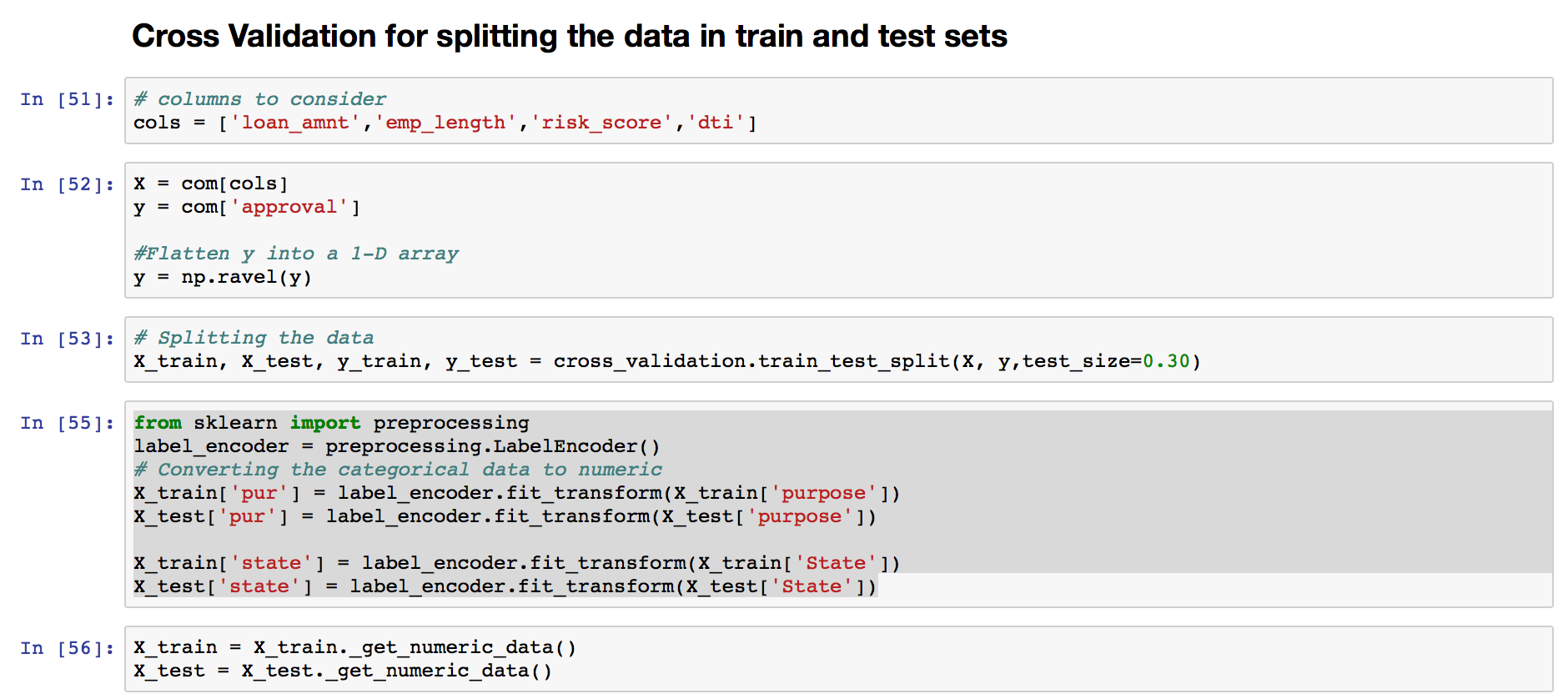
**2(a): Classification**

* For classification, we combined the loan and decline data keeping only the columns in both the datasets. We have generated a flag column named ‘approval’ during the cleaning process.

approval ->1 Loan data  
 approval ->0 Decline data

* We create a dataframe having equal number of rows(1640753) from loan and decline data.
* Features we are dealing with for classification:

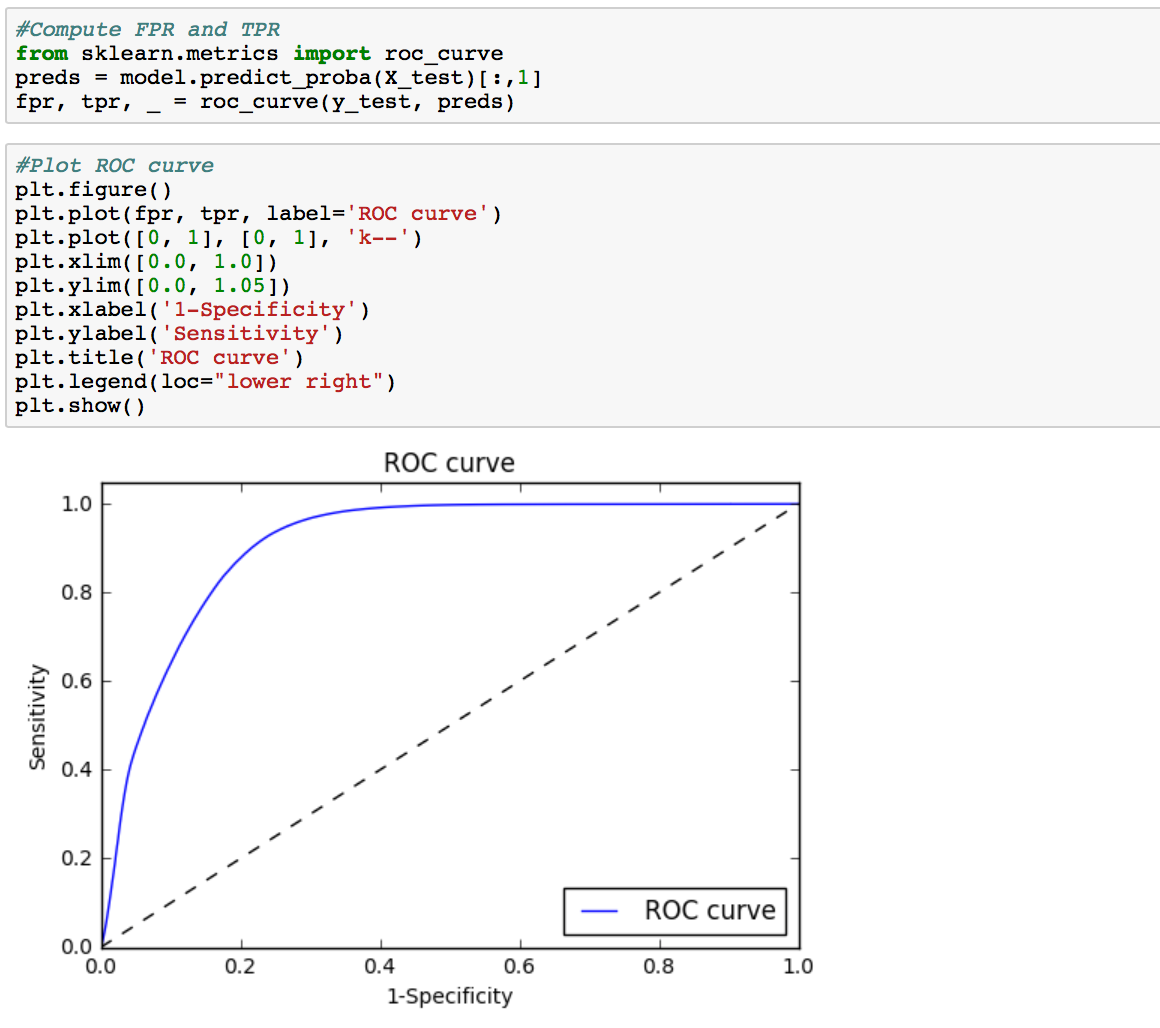
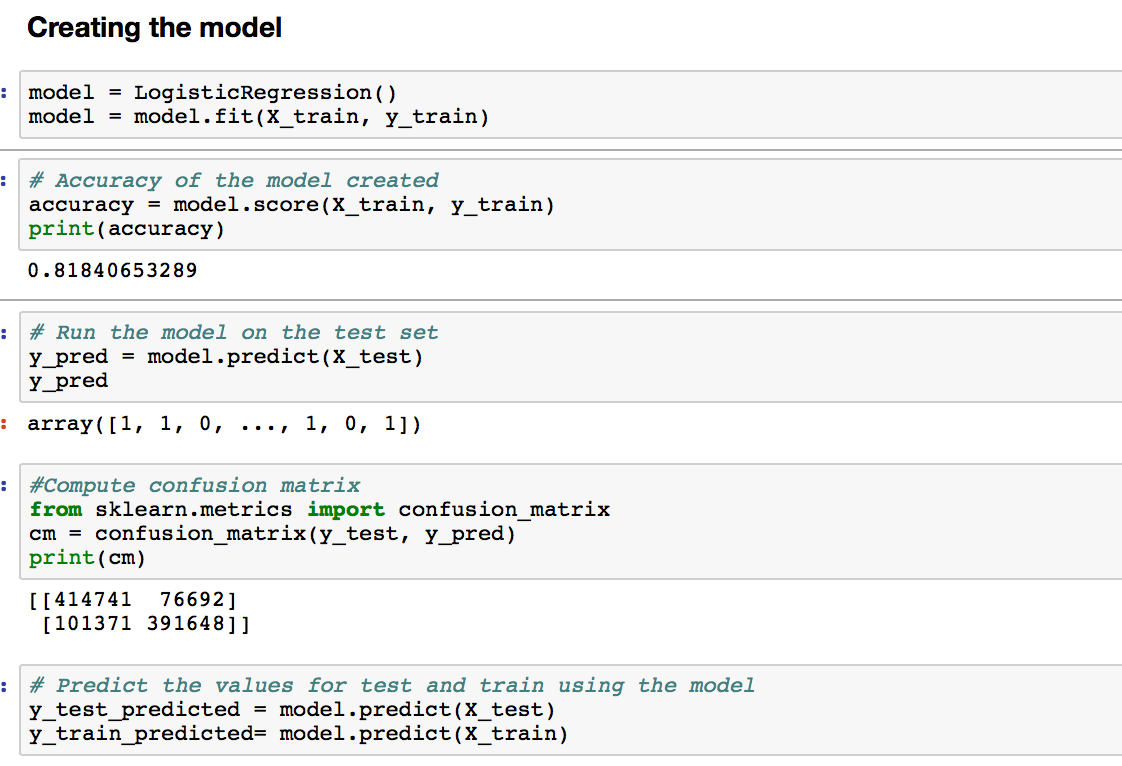
['id','loan\_amnt', 'purpose', 'fico', 'dti','zip\_code', 'State', 'emp\_length', 'Year', 'Month', ‘approval']]

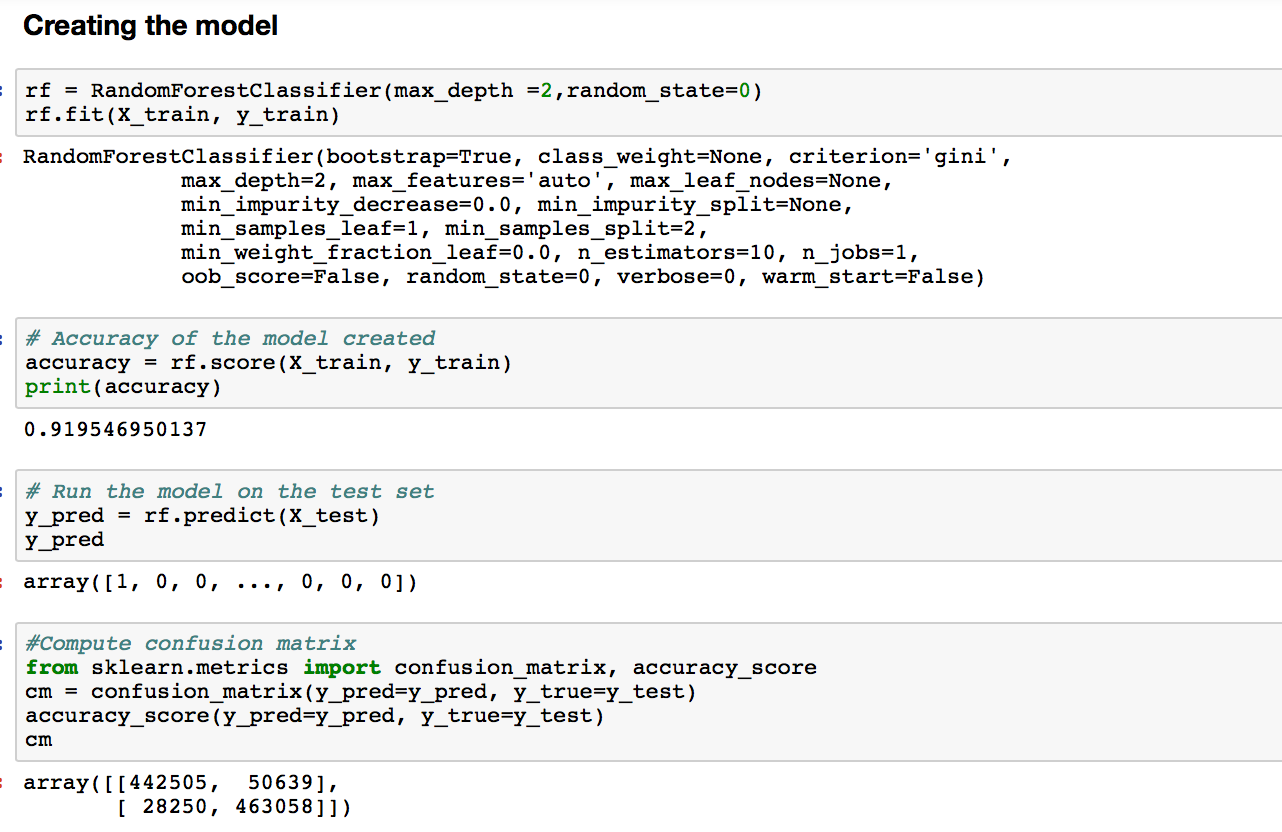
* Then we spilt the data into train and test having 70/30 split.

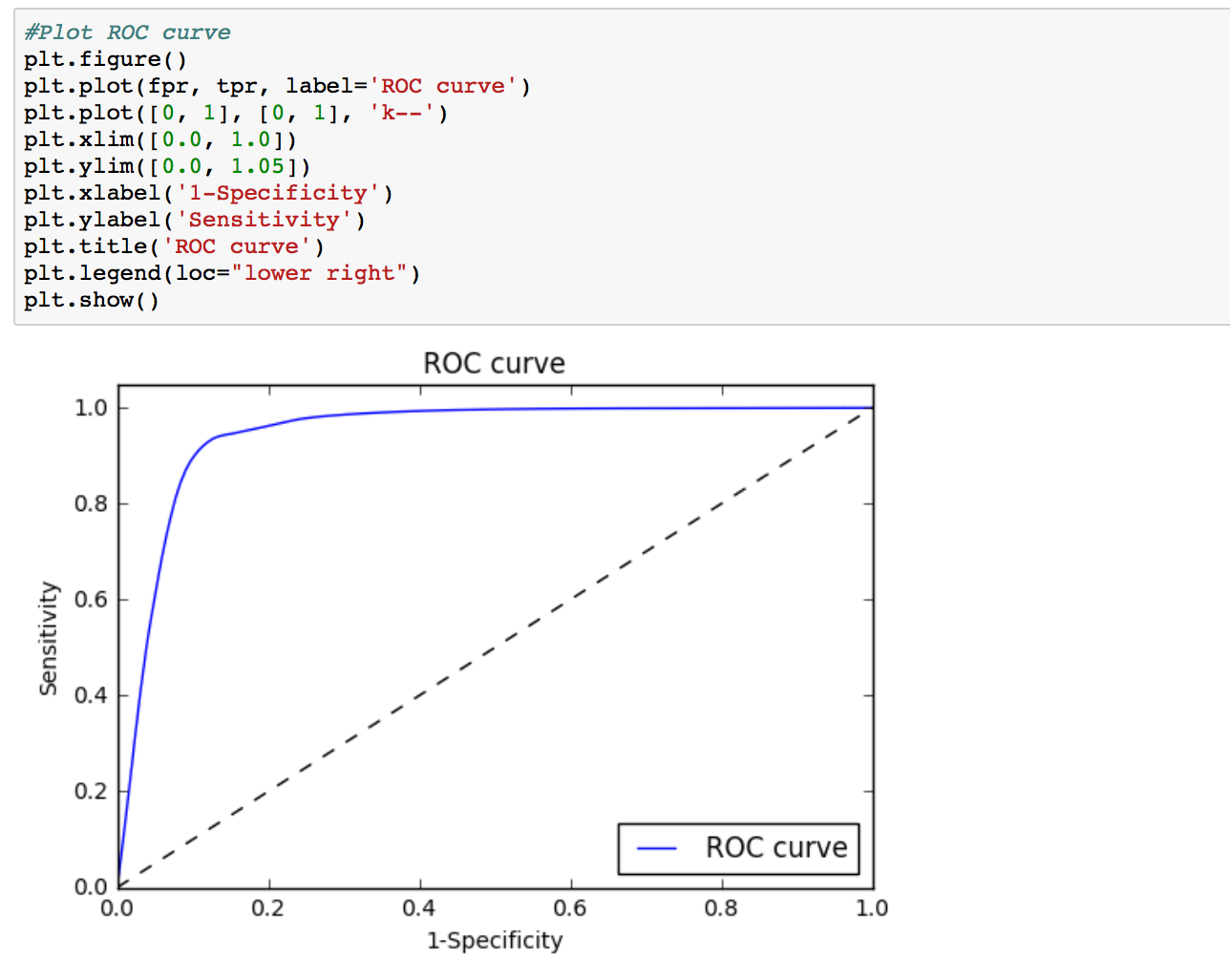
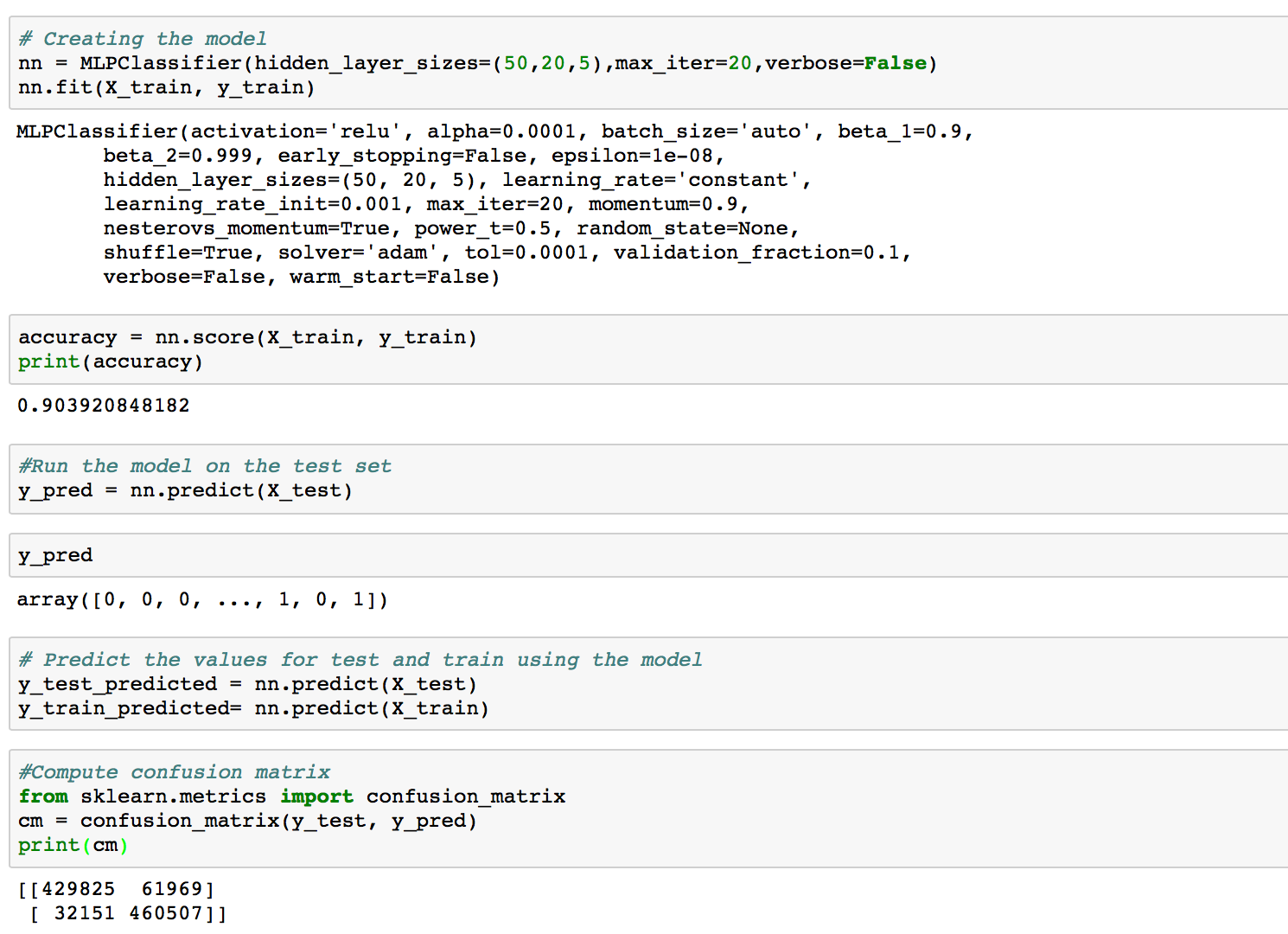
**Models Implemented**

We implement **Logistic regression, Random Forest, Neural Network** models algorithms.

**Random Forest** worked best for the classification.

**Logistic Regression:**

**Random Forest:**

**Neural Networks:**

**2(b): Clustering**

We have segmented our loan data broadly in ways:

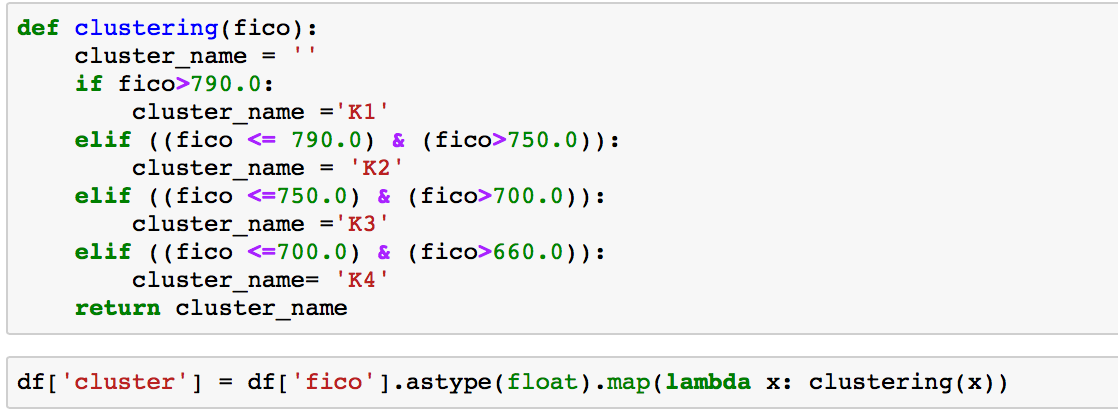
* 1. No cluster(Zero Cluster)
  2. Manually
  3. Clustering Algorithm(K-Means)

**NO Cluster:**

We loaded the loan data as is.

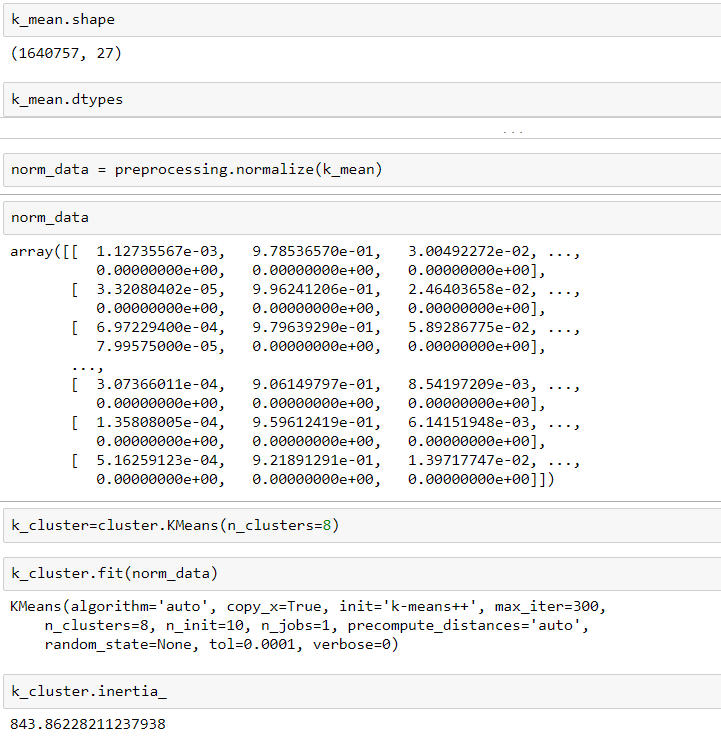
**MANUALLY:**

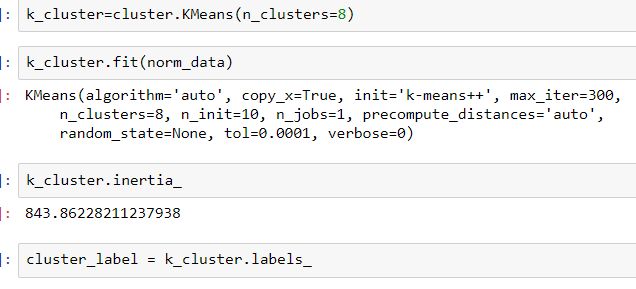
We cluster data based on the fico score in to four clusters. And generated a new column ‘cluster’ for loan data frame. Based on the fido score in each row of loan data we fill in the cluster column with the corresponding cluster name. Then exporting each cluster to a separate CSV.



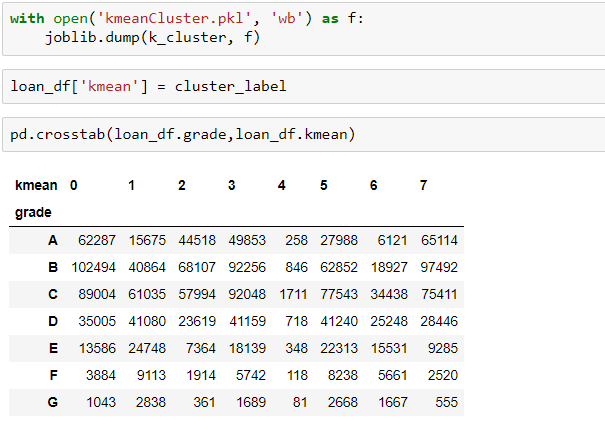
**K- Means Clustering:**

Here we cluster into 8 different clusters and also we are actually dealing with same features as manual cluster. We have normalized data before creating cluster.



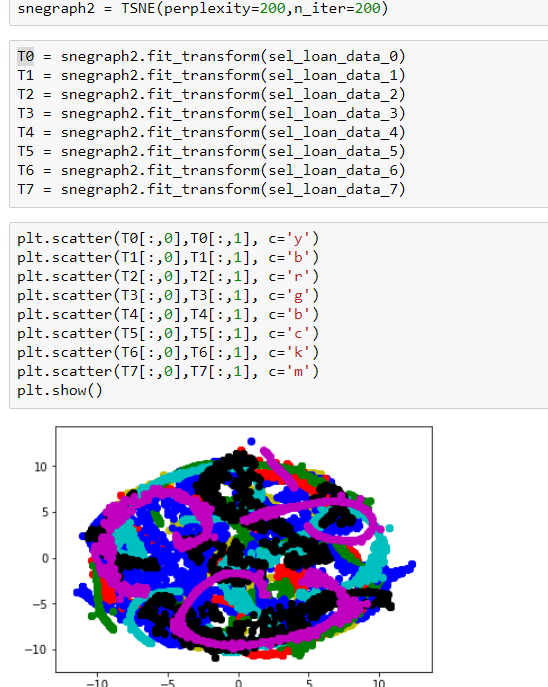


We then create a pkl file which exports the model. Also, we validate the code based on the grade to see if all clusters return data.

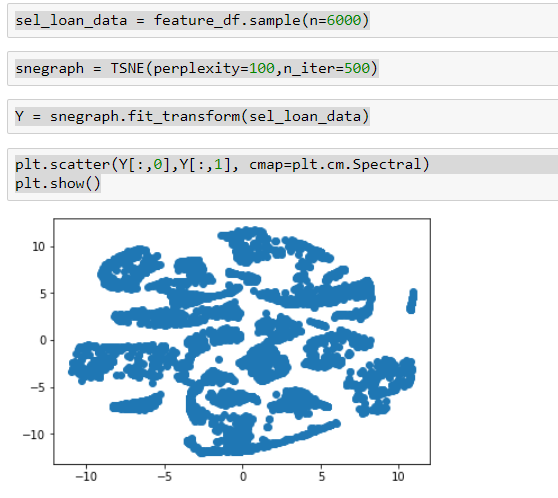


**Here we Have the t-SNE graphs**

t-sne code for each k-means cluster and the output is shown in the graph below. All the k-means are overlaid on each other below.



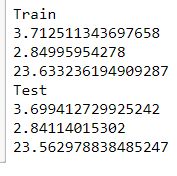
Below is t-sne for 0-cluster.



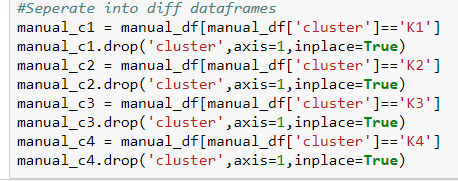
**2(c): Prediction**

We have a file that is created for regression and in that we have run all 13 clusters.

So a model for no cluster with below result:



Then for manual cluster being split into 4 different data frames then run through.

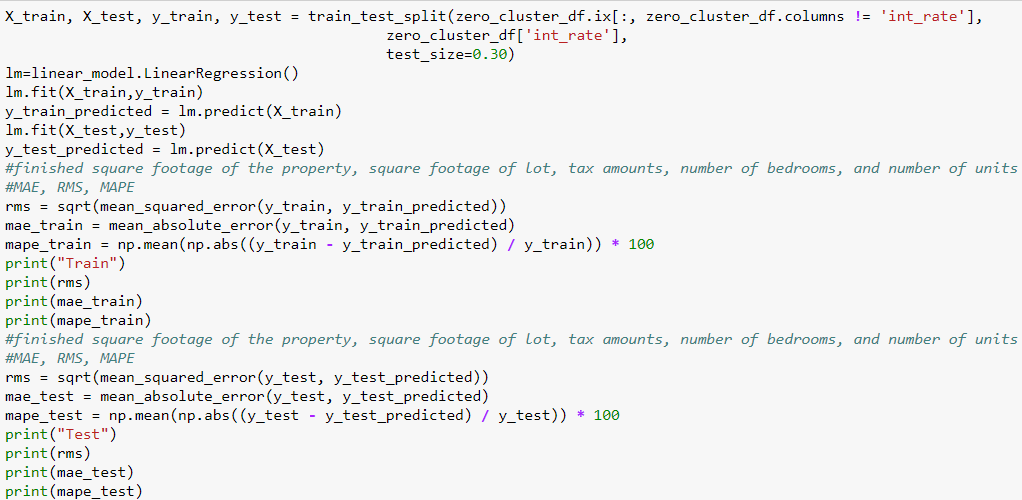


Lastly we ran for K-Means Cluster

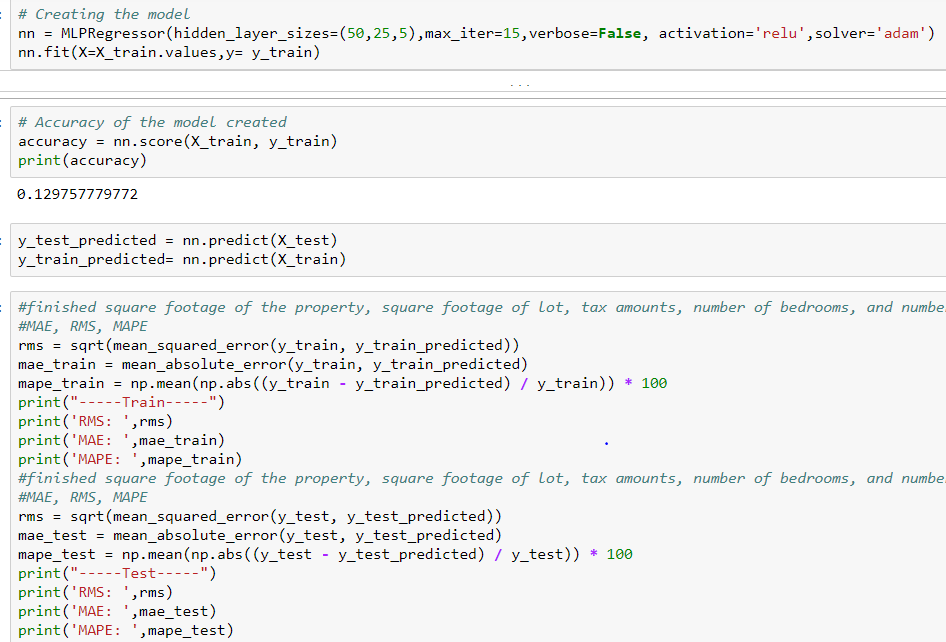


**Regression**

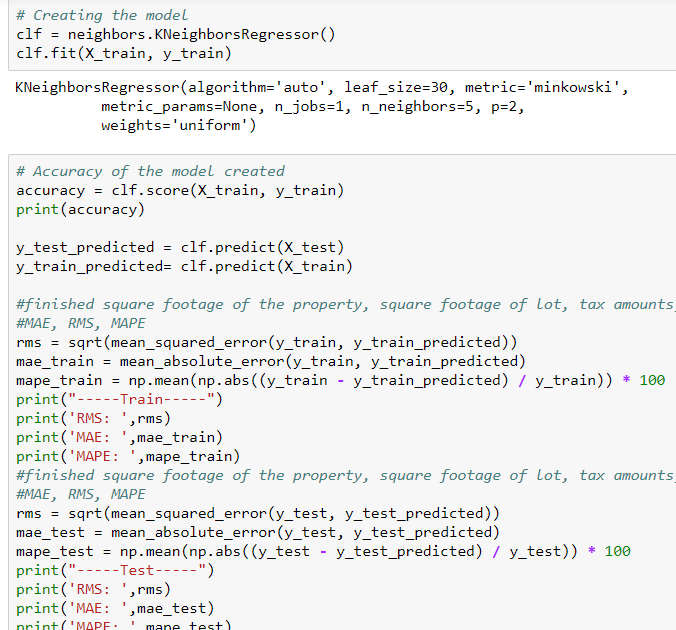
The below code has been run for each cluster.



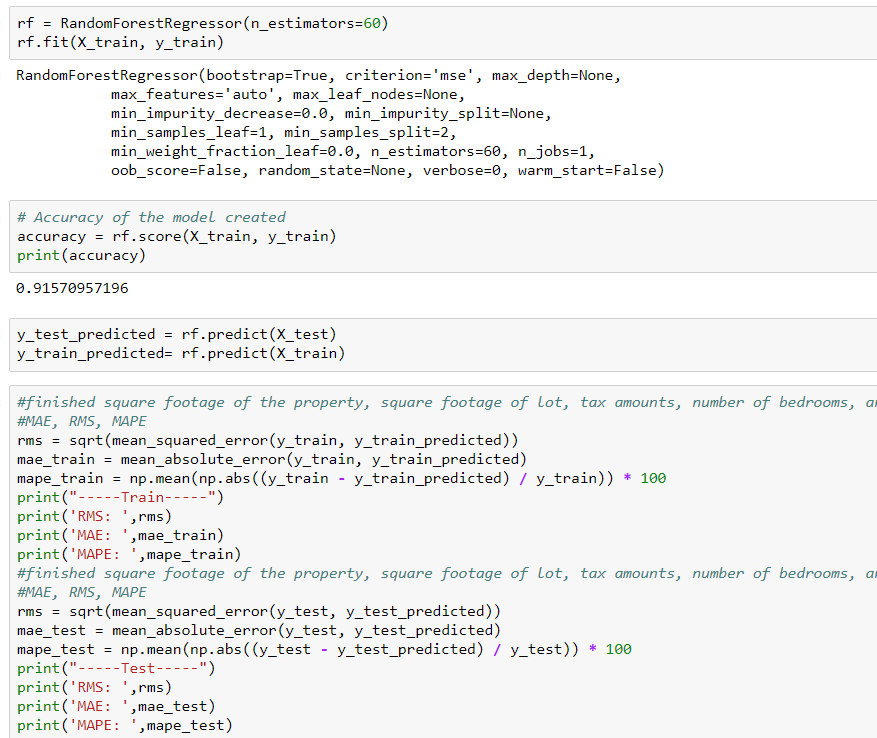
**Neural Network**



**KNN**



**Random Forest**



**We have Created A Results\_Cluster excel file which has all the training and testing outputs for prediction.**

**2(d): Deployment**

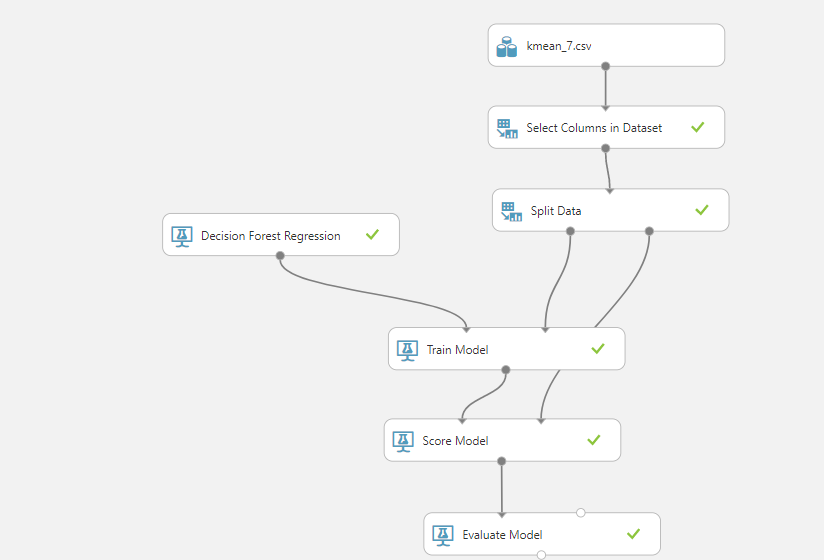
**Microsoft Azur**

We have selected RANDOM FOREST to deploy as the best model for all 13 clusters (1 0-cluster, 4 manual clusters, and 8 K-means cluster)

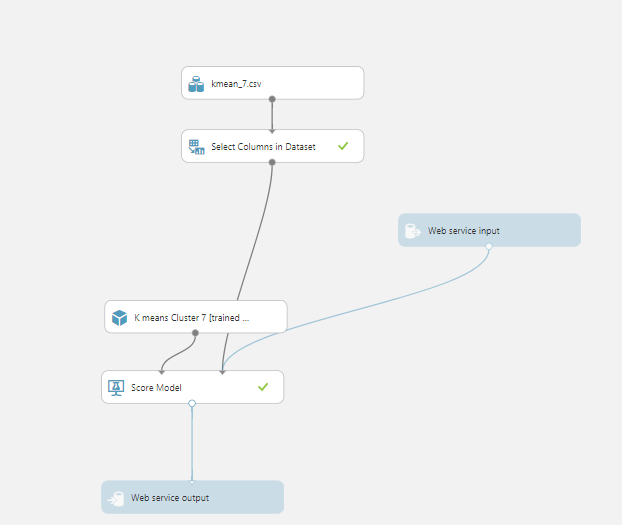
Experiments were created using a csv for each type of cluster.



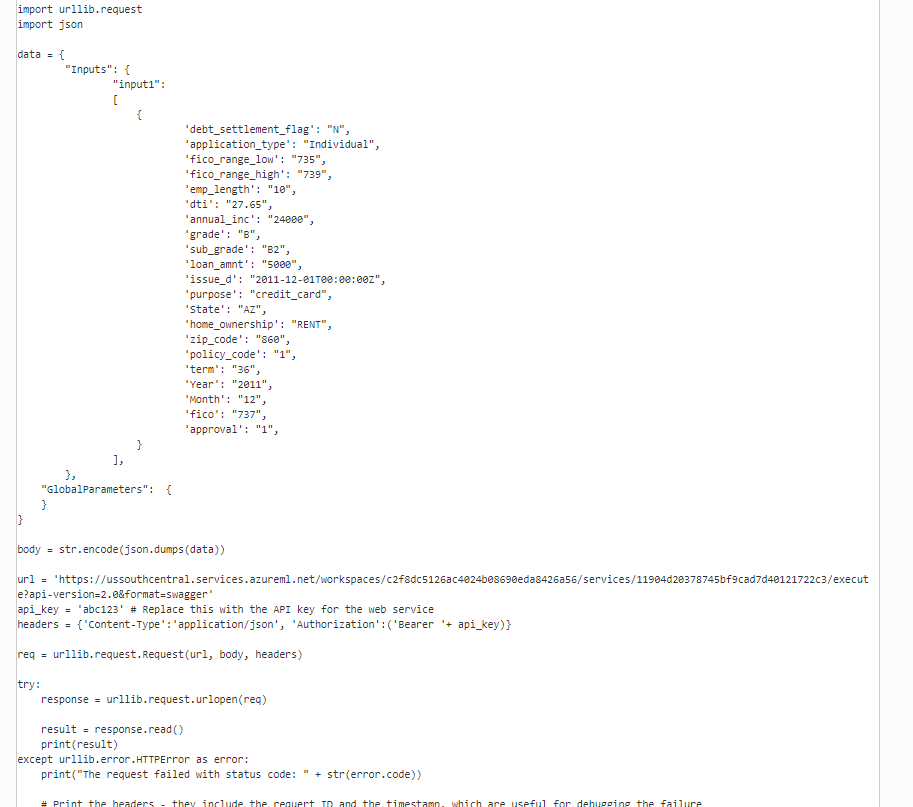
Below you will be able to see how we modeled a sample prediction model using cluster csv. All clusters have same model but diff csv.



We then setup web service. In this step we change the “selected column” and take out the interest rate as we are calculating that, and any other columns that we do not want.

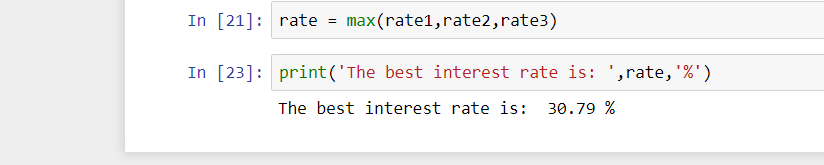


After that we deploy and get API and we go to “Test” and go to config to get python 3+ code. We have 13 AIP and below we can see the code for python.



We then wrote a jupyter script to use these AIP and payload to first use classification to determine if to give loan or not. Then we run for each of the 3 clusters and pick the best interest rate.

**Deployment**

* There are bunch of input values a user has to give like
  + Loan Amount
  + Income
  + Debt to Income Ratio
  + Zipcode
  + State
  + Year
  + Month
  + Length of Employment
* No Cluster: First we pass these parameters and pass to the No Cluster deployment code inputs and get the predicted interest rate after parsing the result and store it in ‘rate1’
* Manual Cluster: Then we check for the fico score the user entered and get the cluster which it belongs to. The we pass those inputs to those cluster deployment code and get the interest rate predicted and store it on ‘rate2’.
* K-Means: We pass the input parameters to all the 8 clusters we have and get the interest rates predicted for each cluster. We get the max of all the resulted predicted interest rates for K-Means clusters and store it in ‘rate3’.
* Finally we have 3 interest rates predicted for our 3 clustering methods.
* We get the best interest rate by taking the max of the rate1, rate2, rate3.
* 

**Analysis**

While looking at the data we realized that there were 157 columns. We have removed everything about 70% missing. When analyzing this data, we realized that there in decline data, there was a lot of risk score data missing. There was no point of filling in the data as it has no basis of what to fill it o since there was only 9 columns and second we will change the data a lot if we do so. We therefore took a look at the data and removed 1/3rd of the decline data on that basis.

Also in loan data we realized a fico score below 660 is not correct as it was mentioned in the lending club site. A person must have at least that to get a loan. Also, anything below 330 in decline for risk score was removed, since someone cannot have that as a possible fico score.

We have also looked at the relationships between the columns to input the data using knn like annual income and interest rate. Also, there were 2 states when looking into the site that would have to be removed since lending club is illegal there.

Looking at accuracy between classification and prediction, we can see that classification is much better. This would be mostly be because there is a strong distinction as to what kind of data combination will cause someone to get interest rate or not. Also lending club probably only really uses these 9 columns to determine whether to give a loan or not. However, prediction gives an interest rate and we are only considering a certain number of fields. In real life though lending club would be taking into consideration way more than what we did.