**MIDTERM REPORT**

**Advance in Data Sciences and Architecture**

**INFO 7390 - SPRING 2017**

PROFESSOR:

**SRIKANTH KRISHNAMURTHY**

TEAM MEMBERS (Team 9):

**AASHRI TANDON**

**PRAGATI SHAW**

**SARTHAK AGARWAL**

Table of Contents

[Part I: Data wrangling 3](#_Toc477554476)

[(A) Data Download and pre-processing 3](#_Toc477554477)

[(B) Exploratory Data analysis 11](#_Toc477554478)

[Jupyter Notebook Script 11](#_Toc477554479)

[Tableau Dashboard 17](#_Toc477554481)

[Part II: Building and evaluating models 20](#_Toc477554482)

[(A) Prediction 20](#_Toc477554483)

[Script 20](#_Toc477554484)

[Analysis 27](#_Toc477554485)

[(B) Classification 28](#_Toc477554486)

[Script with multiprocessing 28](#_Toc477554487)

[Running Various Scripts in Docker 39](#_Toc477554488)

[Contributions 40](#_Toc477554489)

[References 41](#_Toc477554490)

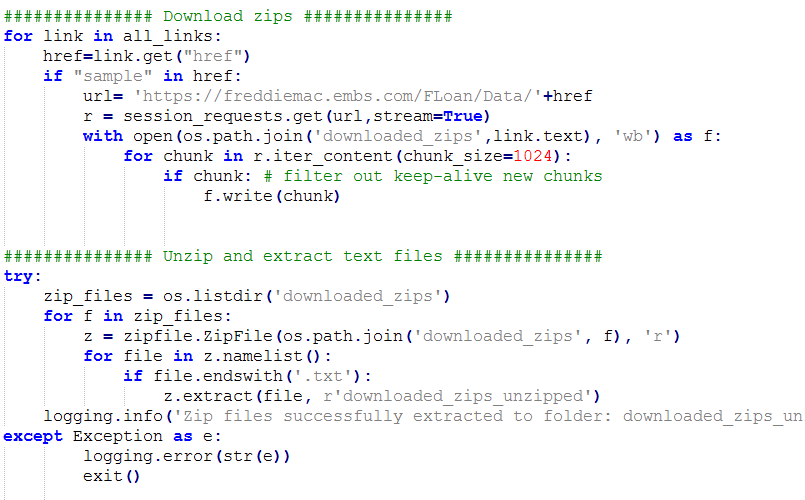
# **Part I: Data wrangling**

## **Data Download and pre-processing**

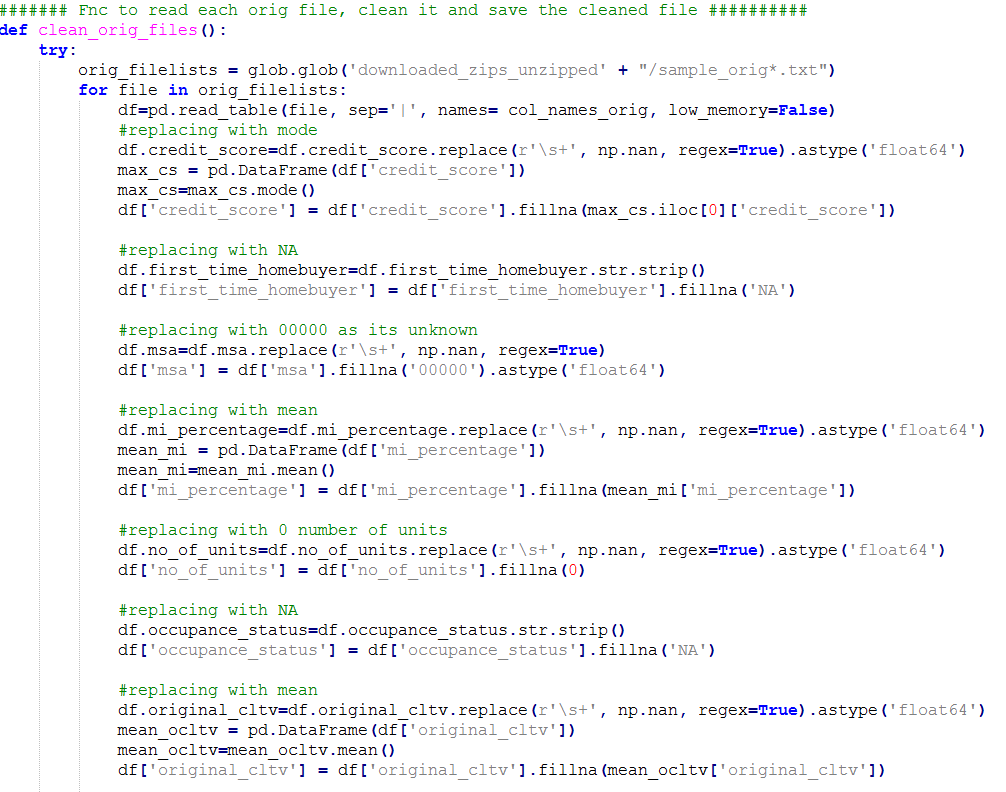
1. Login into Freddie Mac loan website using your credentials.



1. Download the sample zip files and extract them

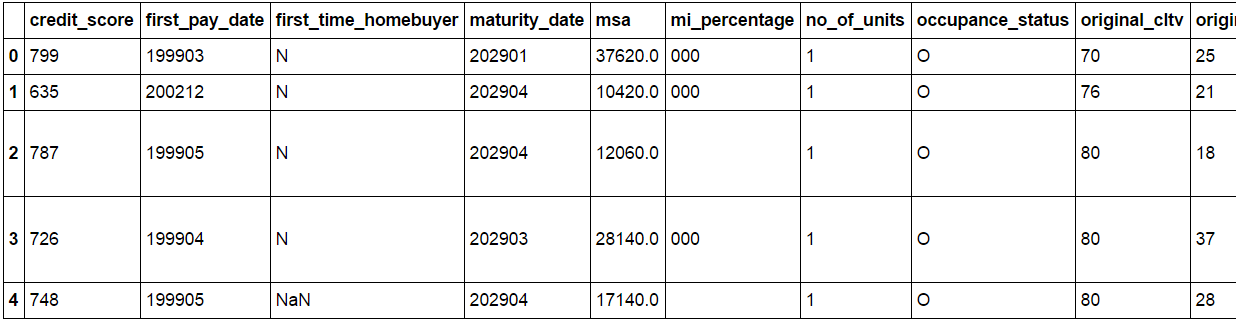


1. Clean the origination file

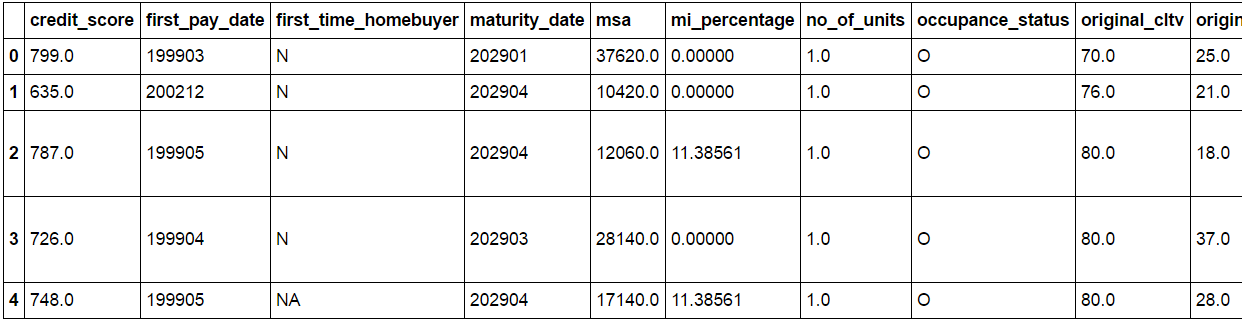




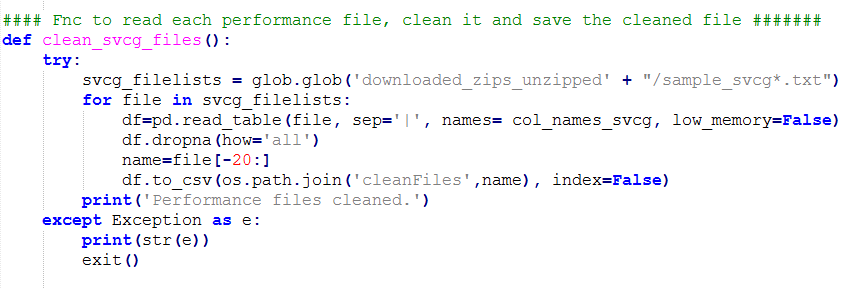
Data before cleaning:



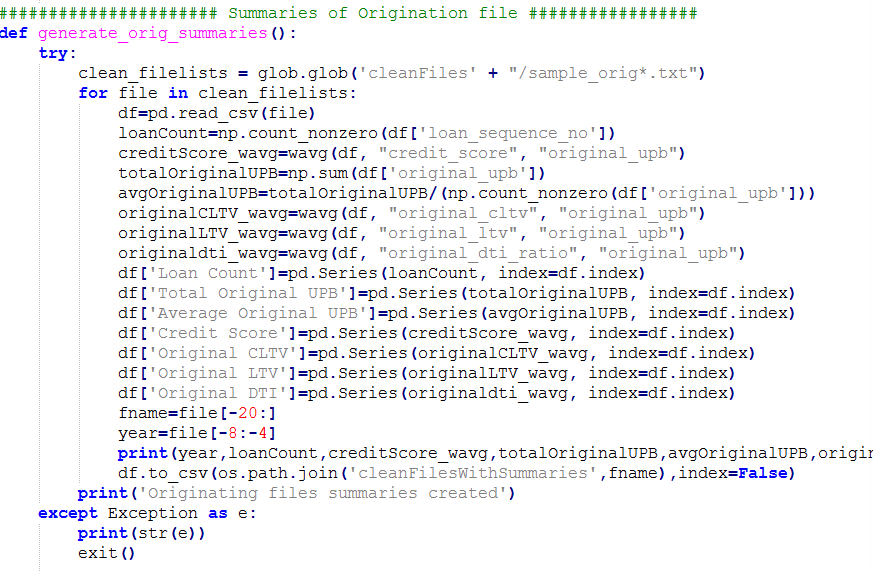
Data after cleaning:

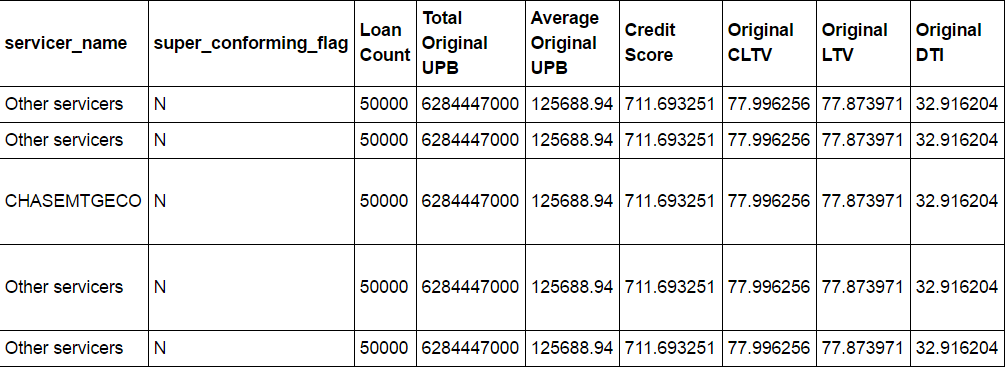


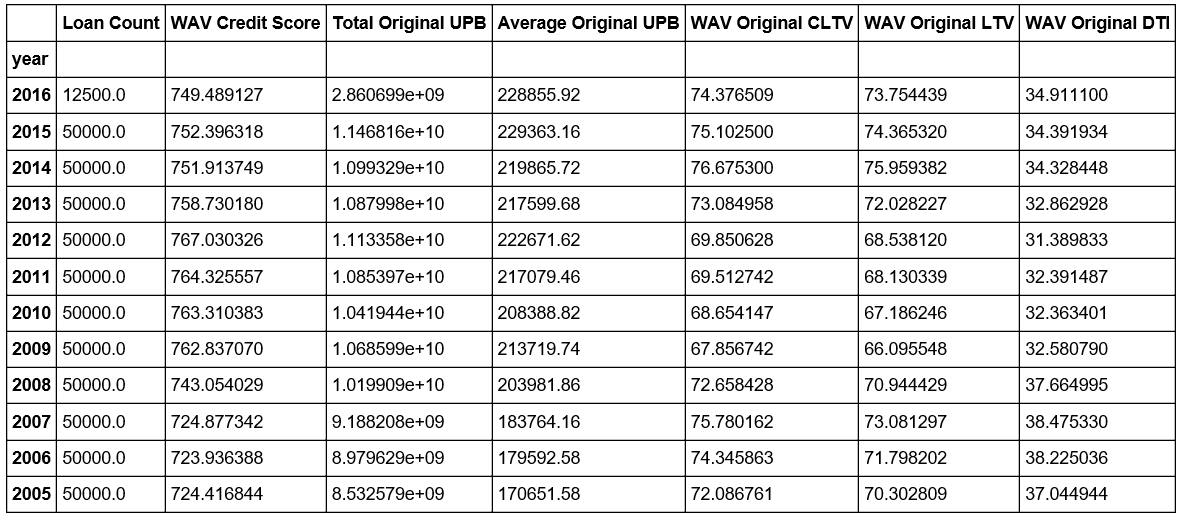
1. Clean the performance file



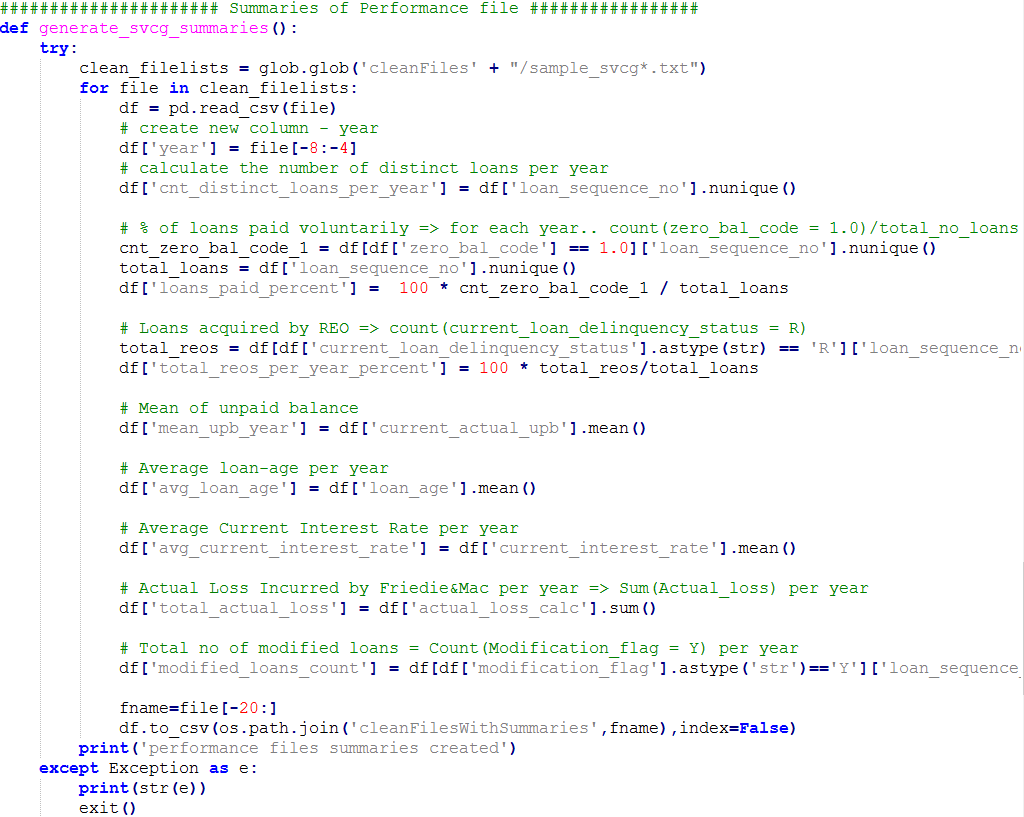
1. Generate Summaries for the origination file. We have computed the following summaries: *Loan Count, Total Original UPB, Average Original UPB, Weighted Credit Score, Weighted Original CLTV, Weighted Original LTV, Weighted Original DTI*



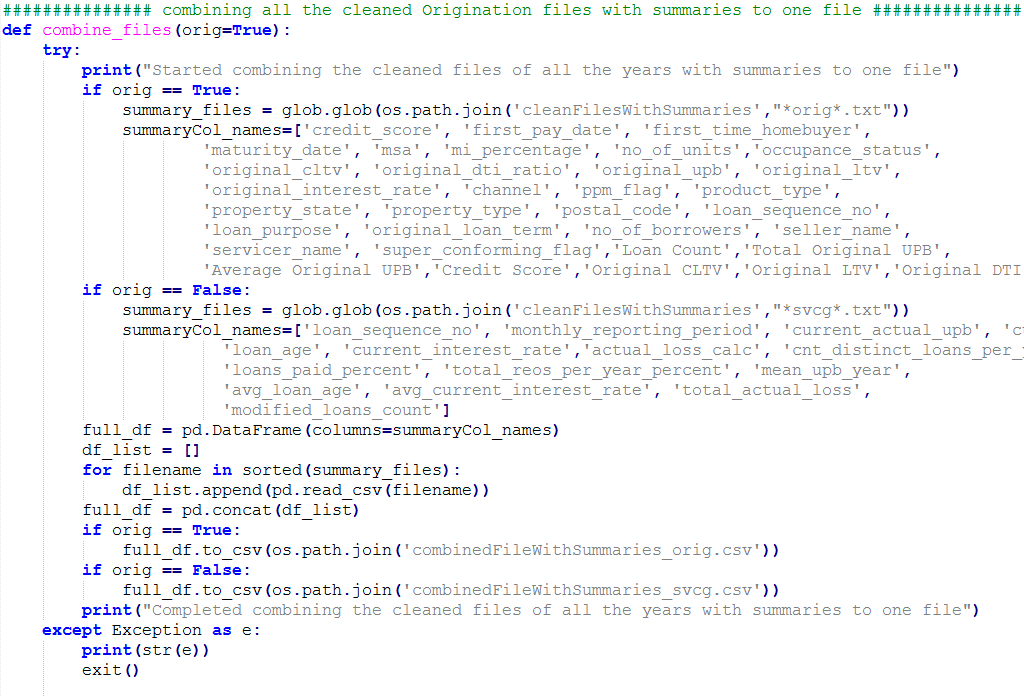




1. Generate summaries for the performance file. We have computed the following summaries: *cnt\_distinct\_loans\_per\_year, loans\_paid\_percent, total\_reos\_per\_year\_percent, mean\_upb\_year, avg\_loan\_age, avg\_current\_interest\_rate, total\_actual\_loss, modified\_loans\_count*



1. Combining the cleaned files with summaries to one file. The function runs for Origination files and then for performance files



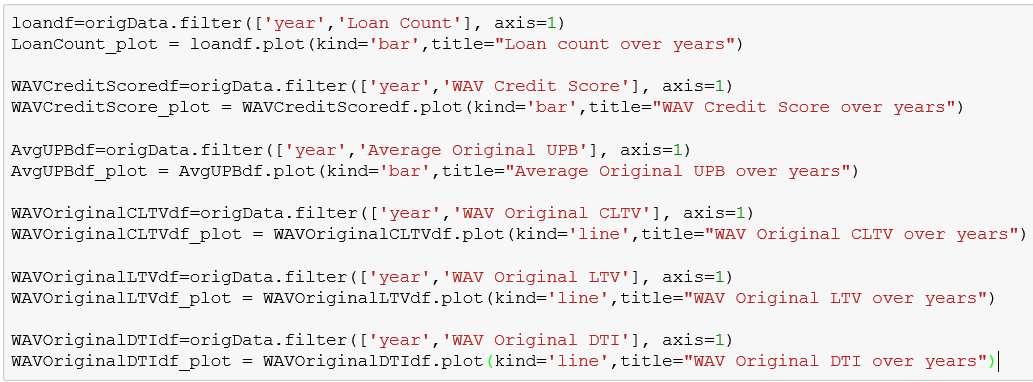
## **Exploratory Data analysis**

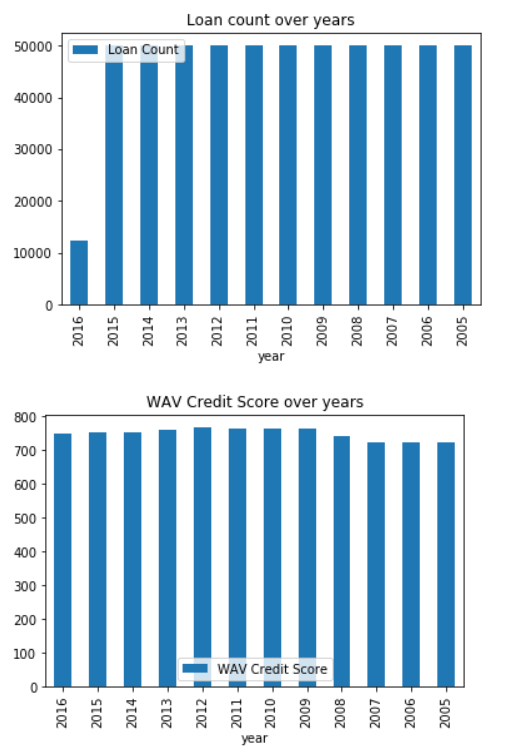
### **Jupyter Notebook Script**

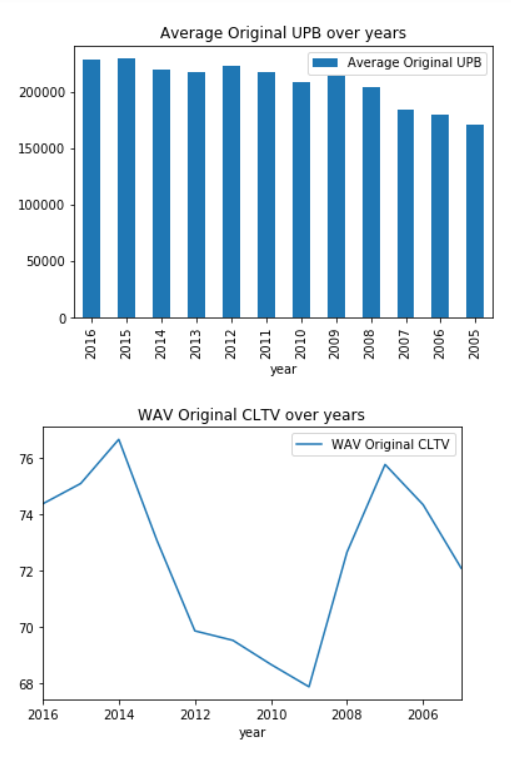
Script to get summary of origination data

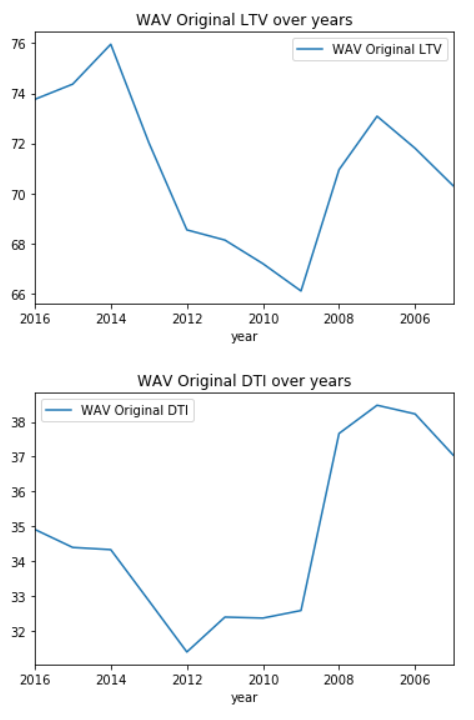


Script to plot summary of origination data:





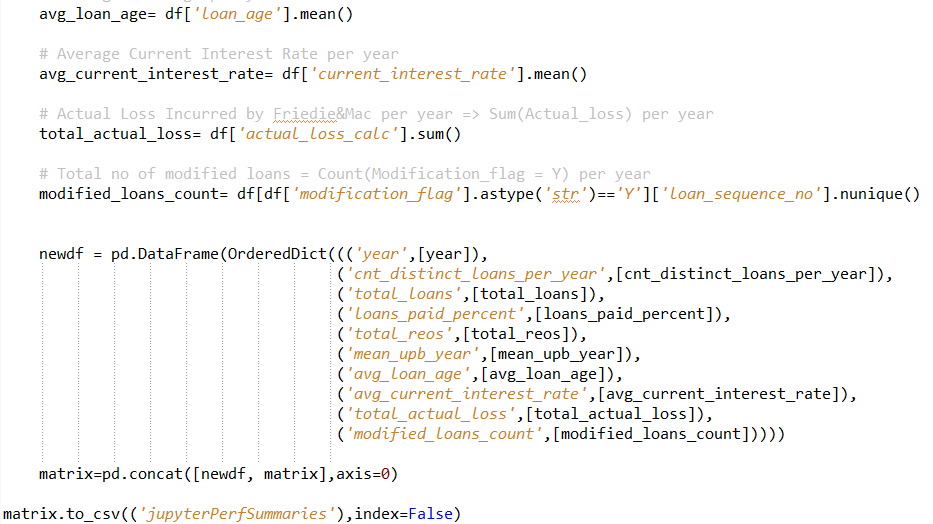




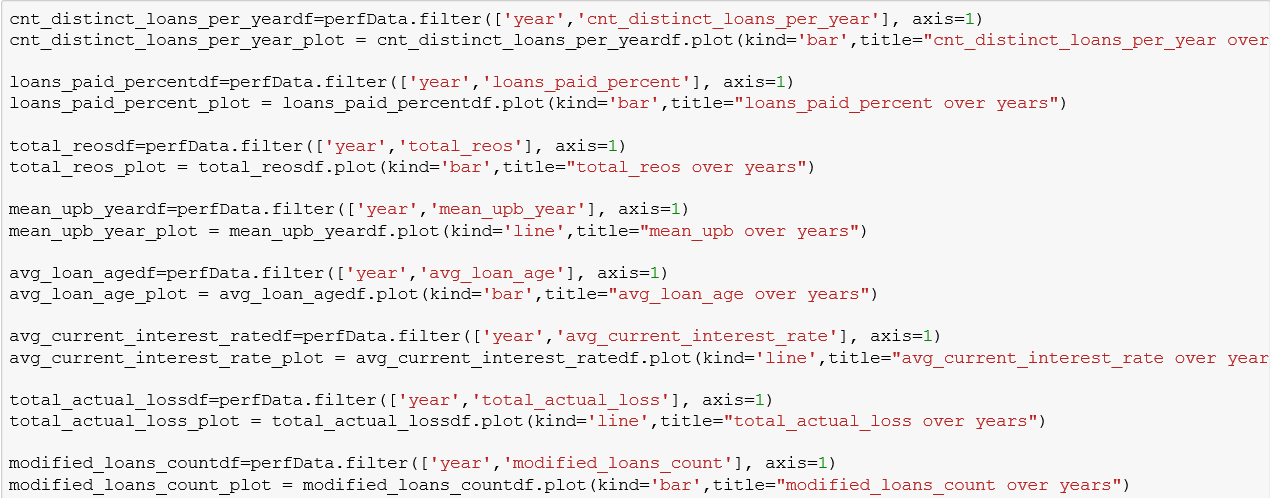
`

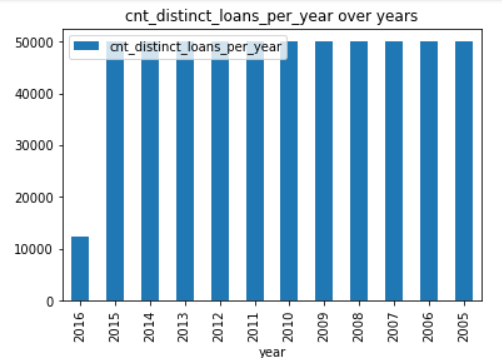
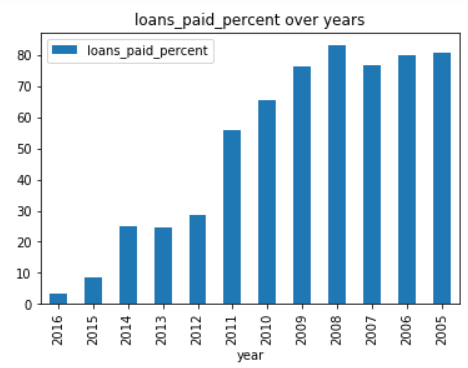
Script to get summary of perforamance data:

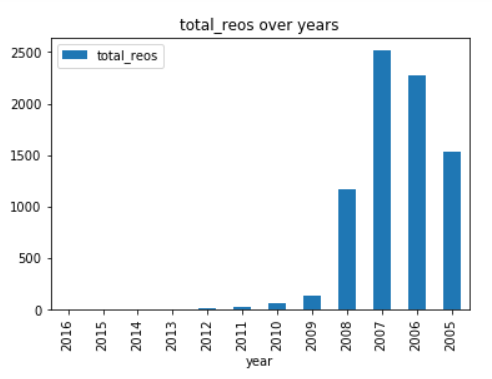
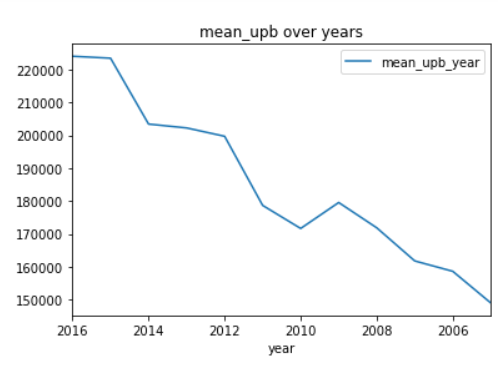
### 

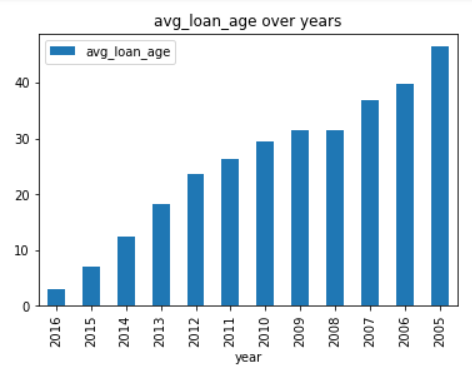
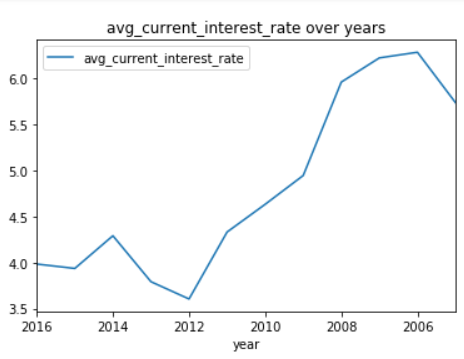


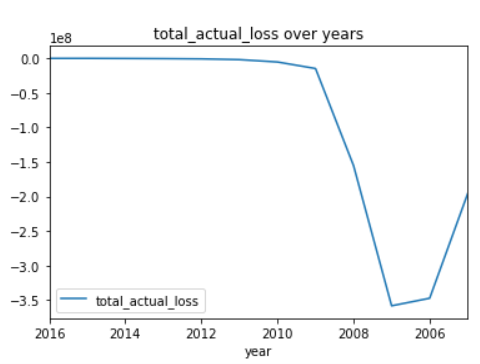
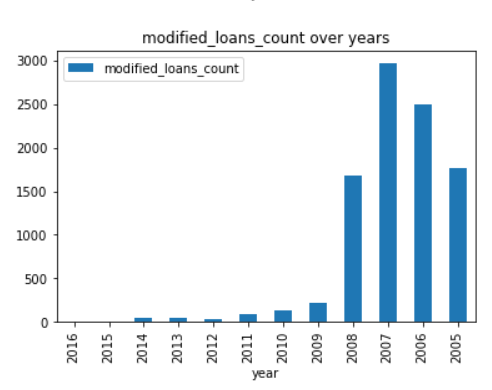
Script to plot summary of performance data:



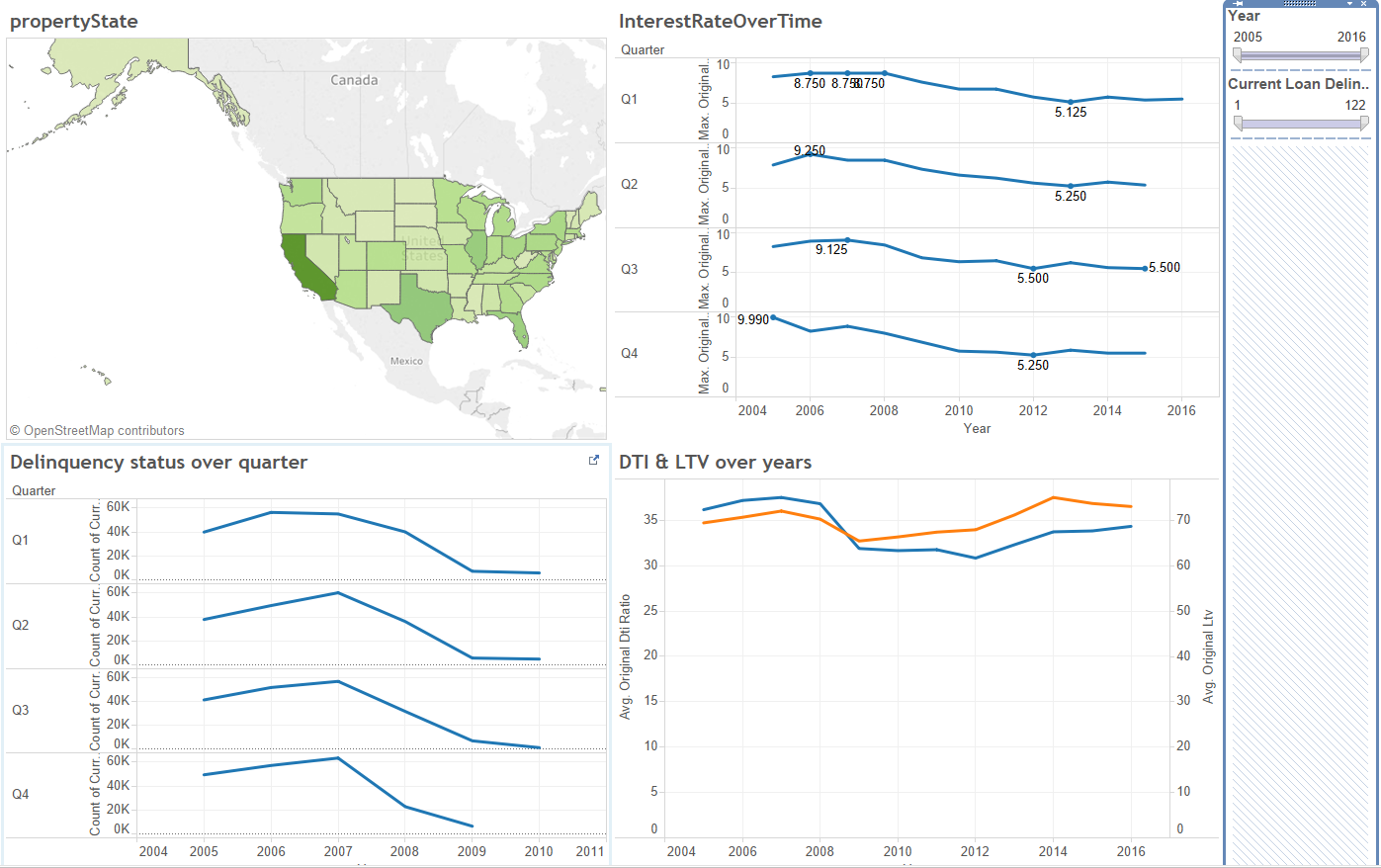
 

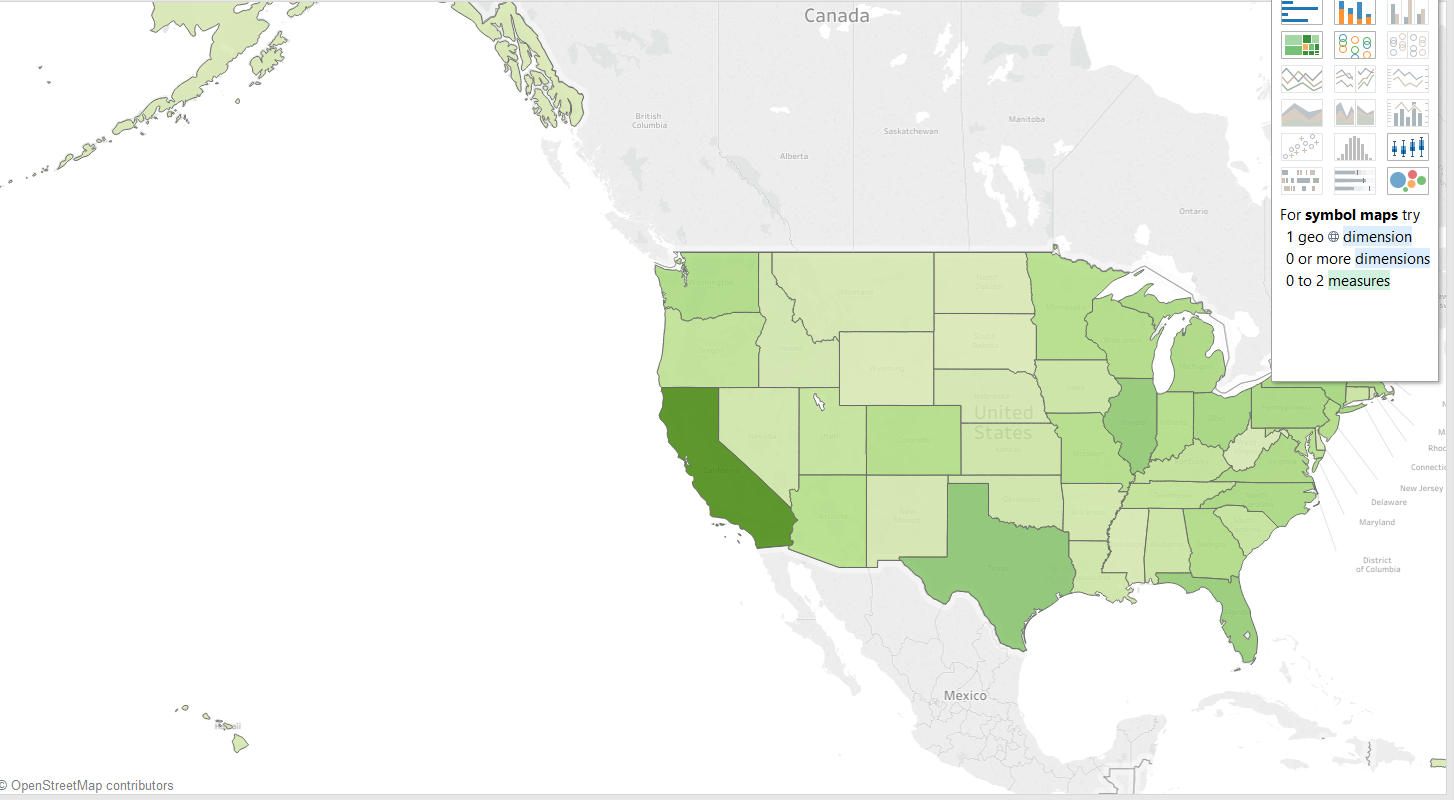
### **Tableau Dashboard**



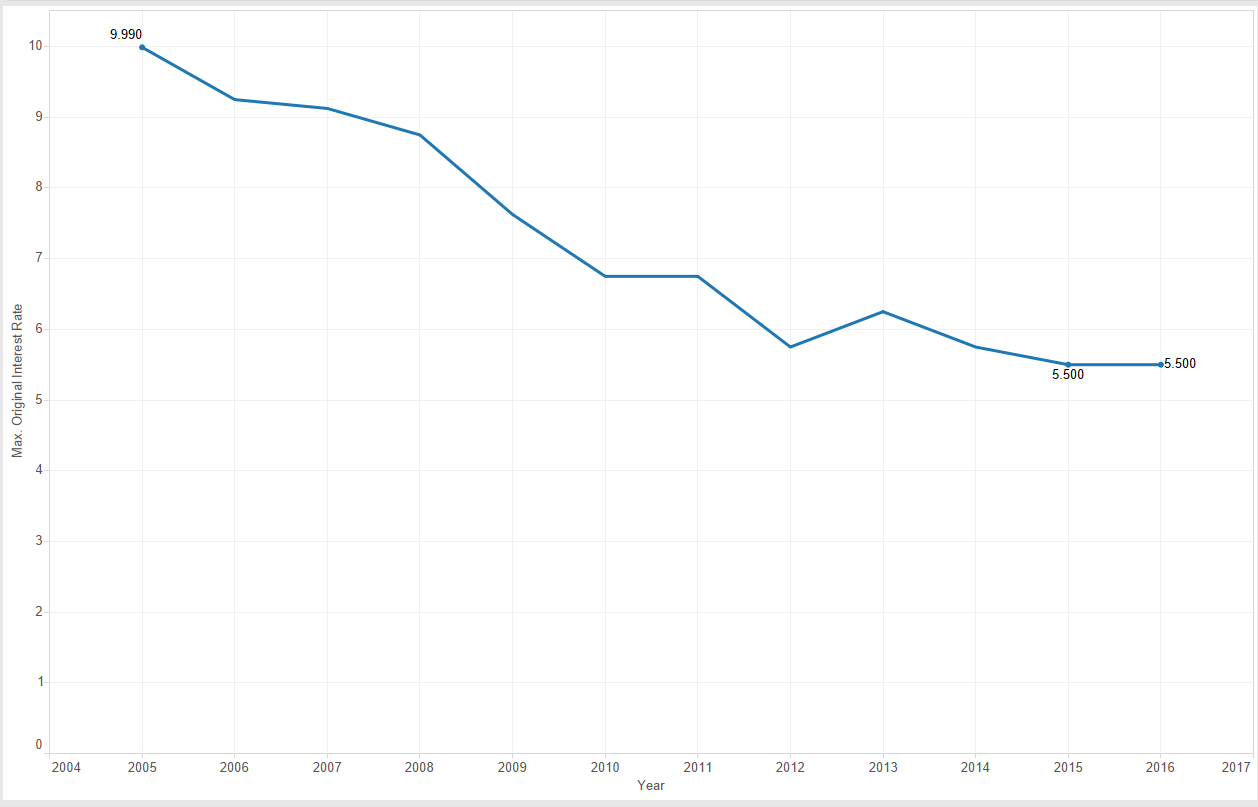
**Analysis:**

Case 1: For California where we see that has highest number of loans, the interest rate per year as well as average DTI ratio had been maximum in the 2007. We see that for the year 2007 the Count of delinquency for California has been highest in Q4. The analysis tells us that there was financial crisis in 2007

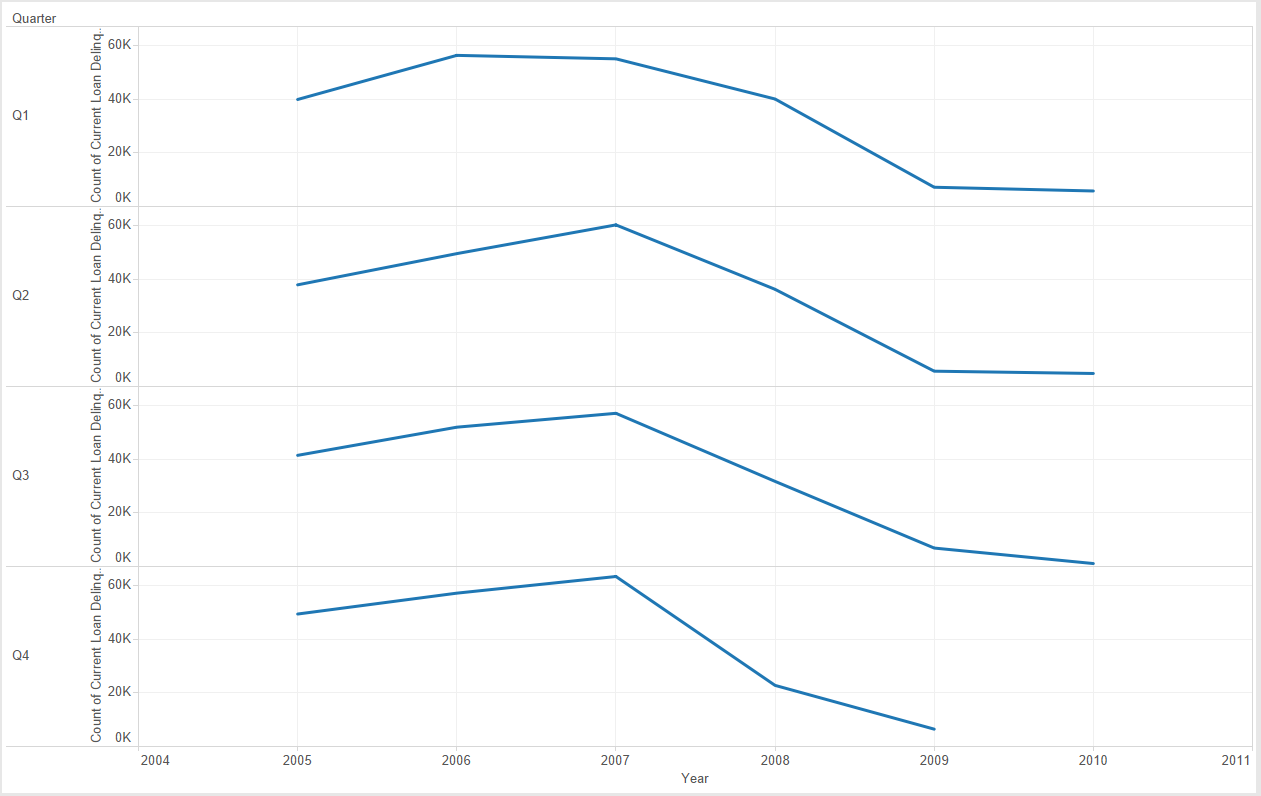
Loan Counts as per property state



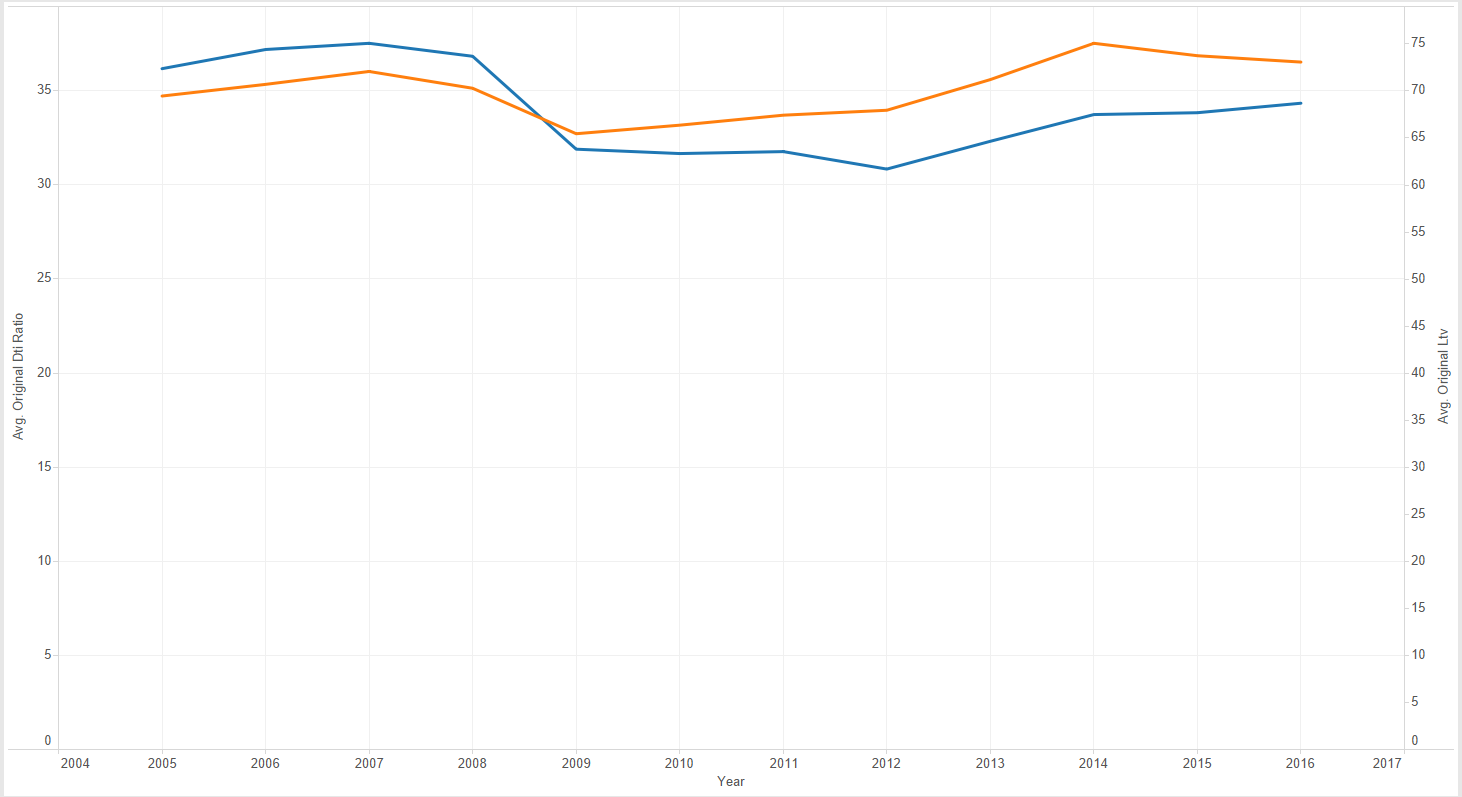
Interest rate trend over the years



Delinquency trends over quarters



Average DTI ratio and average LTV over years



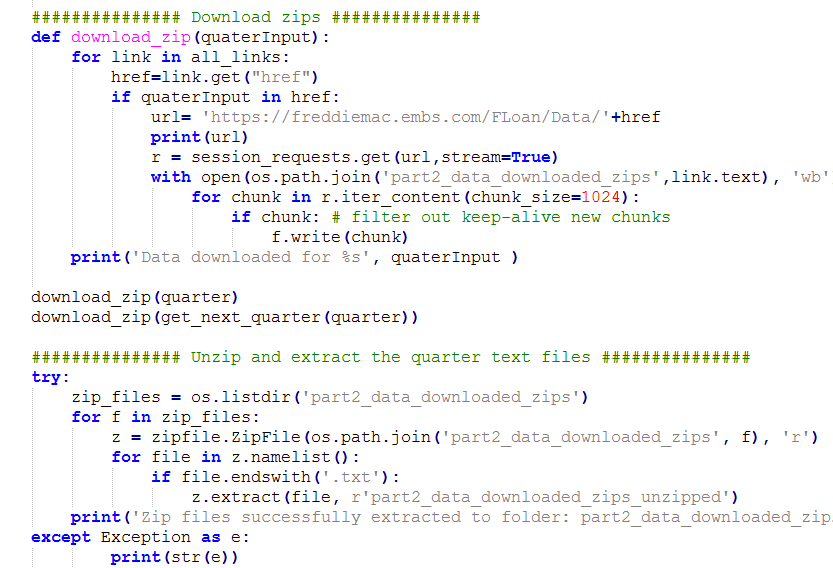
# **Part II: Building and evaluating models**

## **Prediction**

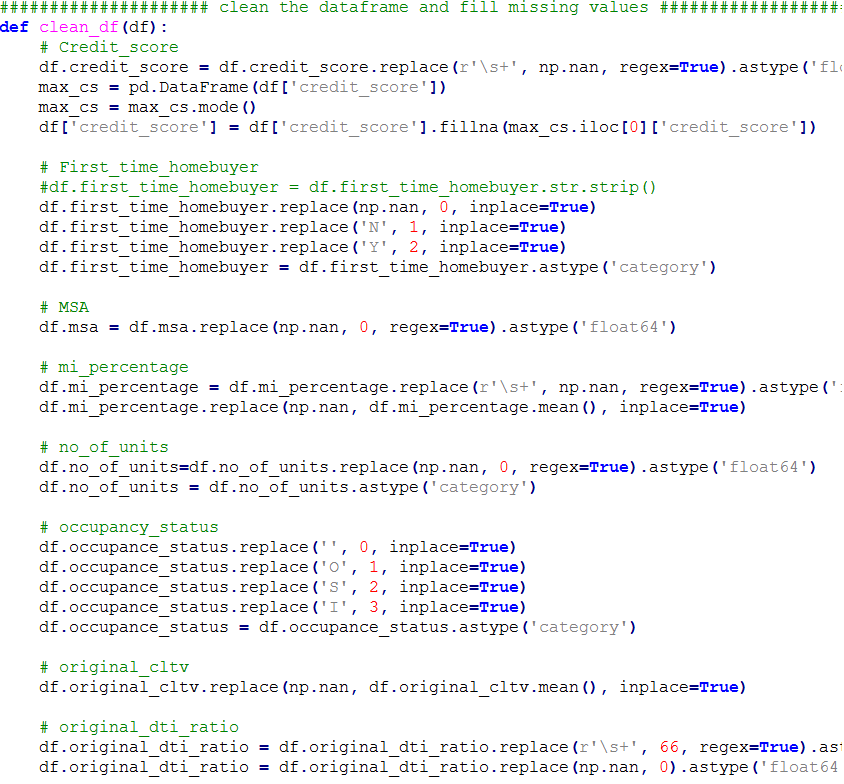
### **Script**

General Flow for Q12005

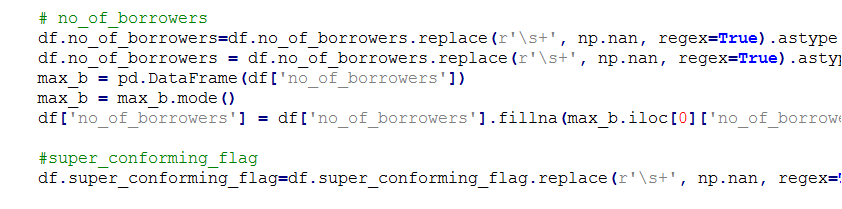
1. We download the quarterly files from the website and unzip them



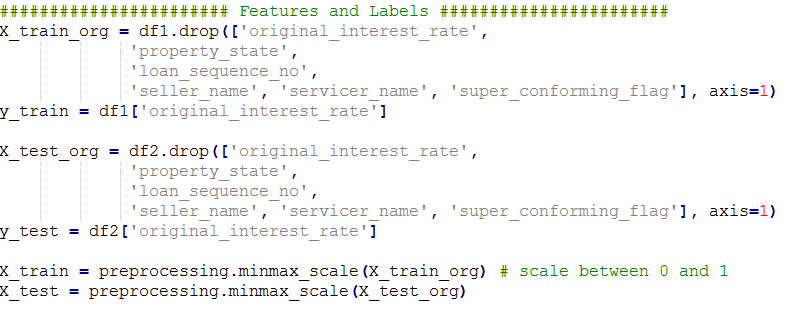
1. Clean\_df() method cleans the data for building regression model



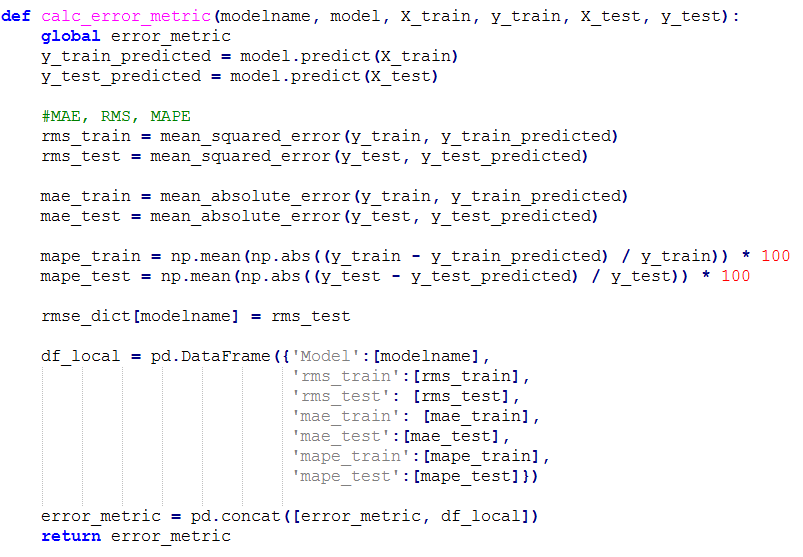




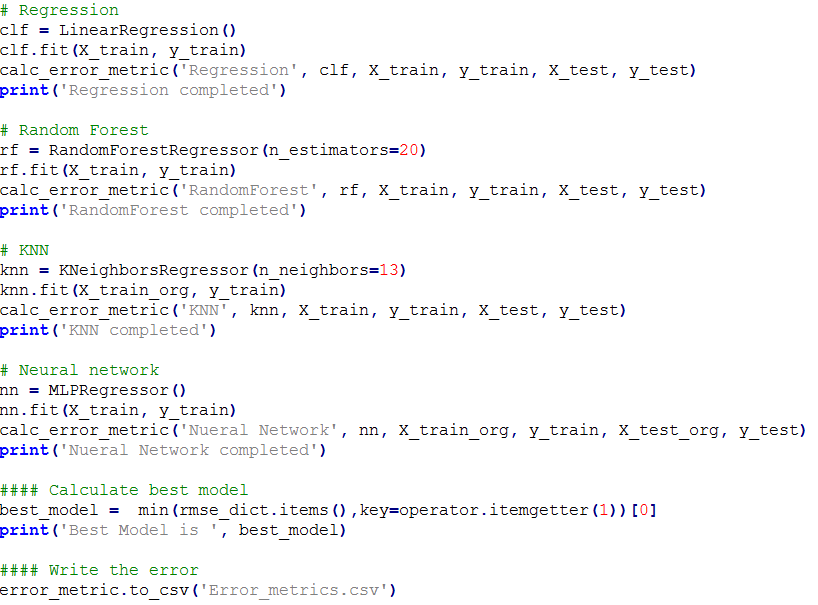
1. Creating Training and testing data and scaling the features. RFE is used to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and weights are assigned to each one of them. Then, features whose absolute weights are the smallest are pruned from the current set features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.



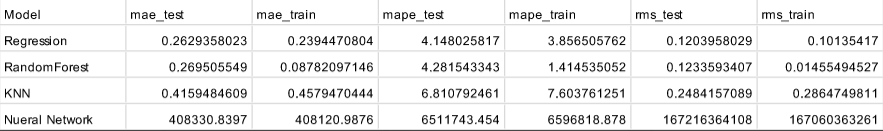
1. Calculation Error Metric for all the models



1. Call the calc\_error\_metric() method for each model and write the metrics to the csv



Output Predicted for Q22007



### **Analysis**

**Financial crisis**

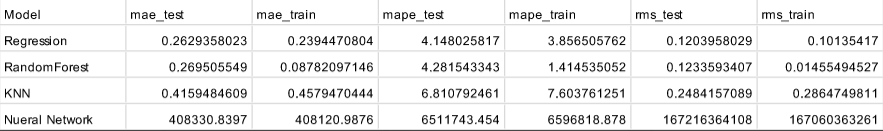
To analyze the global financial crisis of 2007 when there was an increase in perceived credit risk which meant that interest rates were at 45 years low at that point. Firstly, during our initial exploratory analysis, when we compared the average interest rates across all the years we observed that the data also suggest the same.

Hence, when predicting the interest rates on rolling four quarters of 2007 we wanted to predict the same trend and after applying the prediction algorithms we could achieve that.

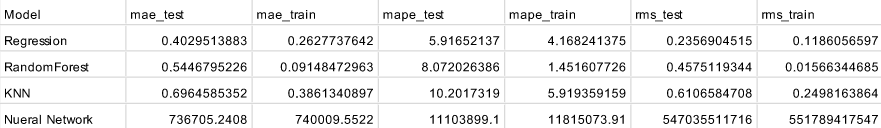
This was a demonstration of a real-world prediction.

Results for year 2007 as input:

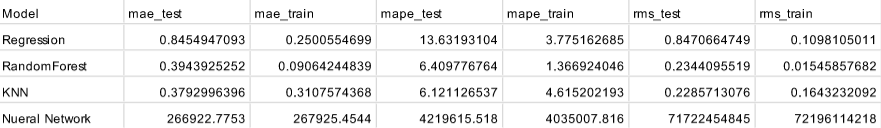
Q22007



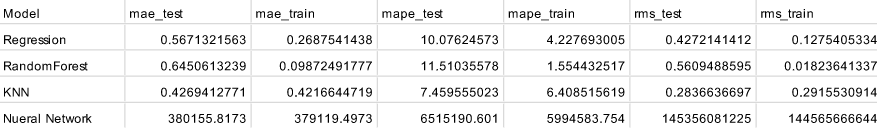
Q32007



Q42007



Q12008



## **Classification**

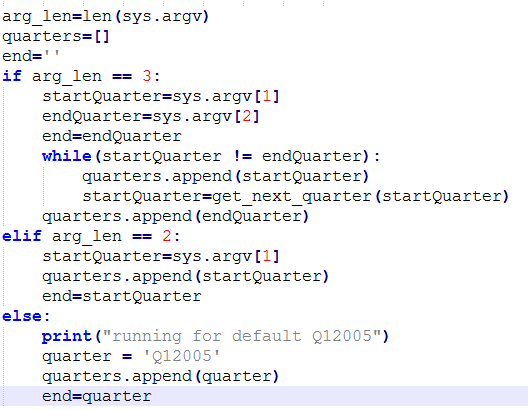
### **Script with multiprocessing**

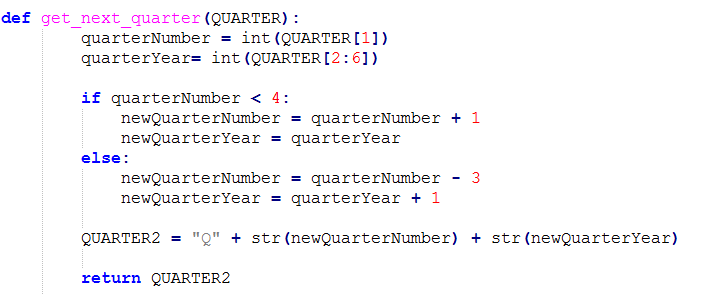
General flow for Q12005

How processes are working:

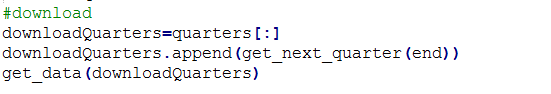
**Steps:**

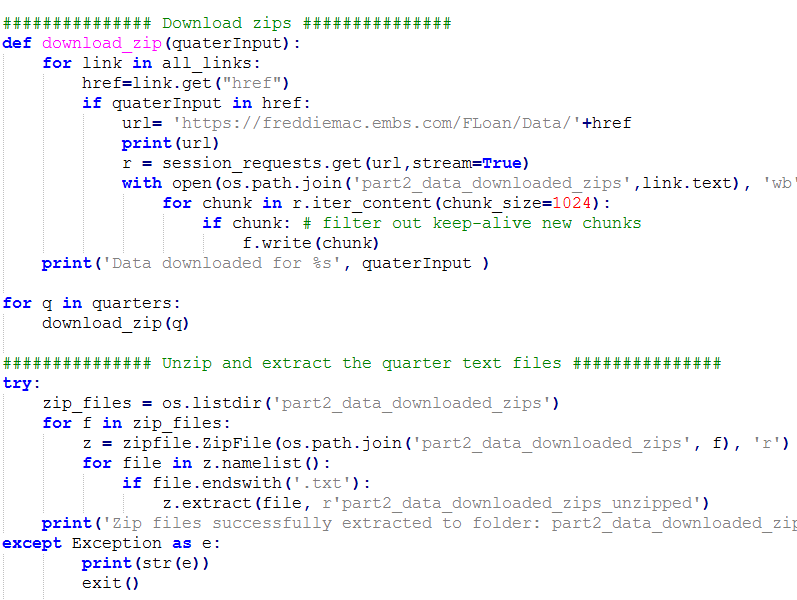
1. The script takes at max arguments. For no arguments, the script runs for default quarter Q12005, for one argument, the script runs for that quarter, and for two arguments, the script generates an array of quarters from starting quarter and end quarter.



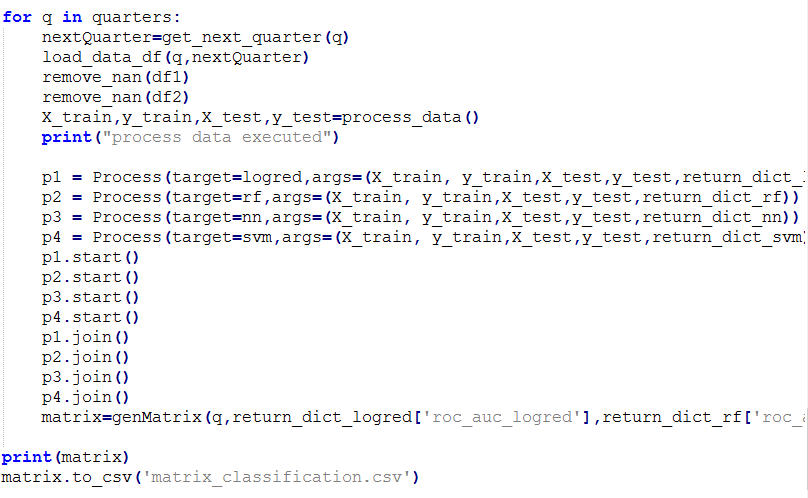


1. Download the quarterly zip files for the quarter data and extract them

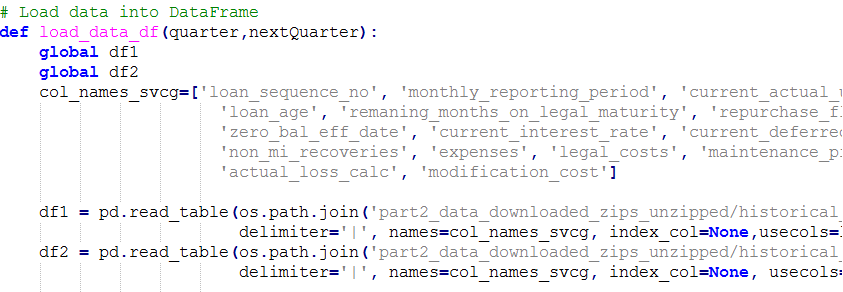




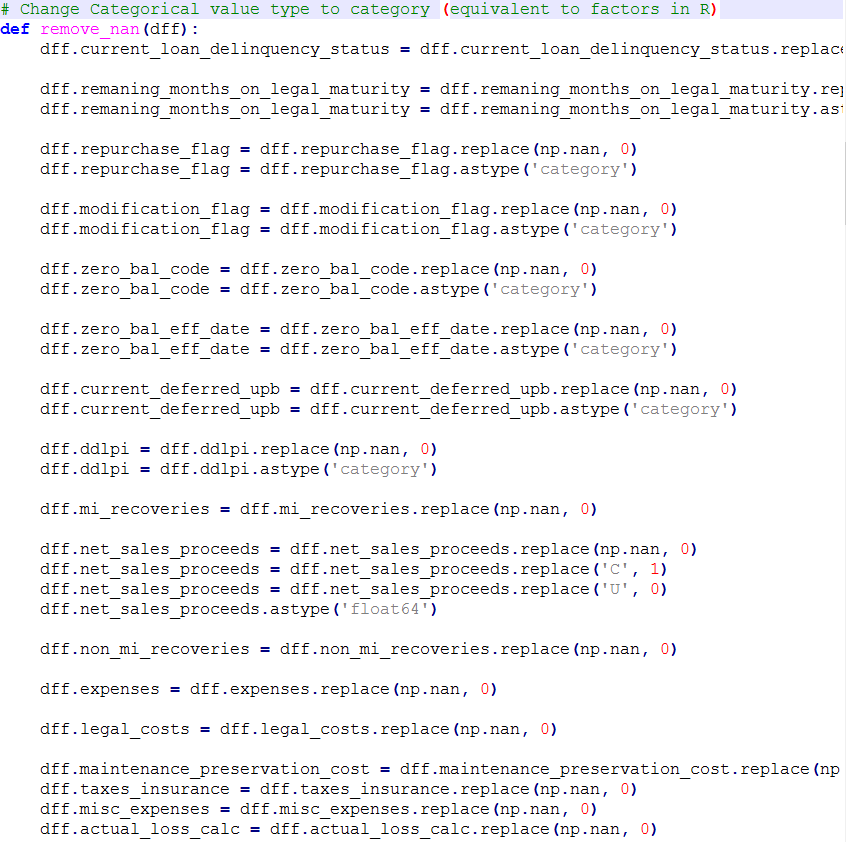
1. We have implemented Multiprocessing in our program. It runs for all the quarters. For each quarter, four processes are started in parallel, which generate their respective confusion matrix and roc\_curves. Once the processes have finished their work (Process.join()), the matrix is generated.



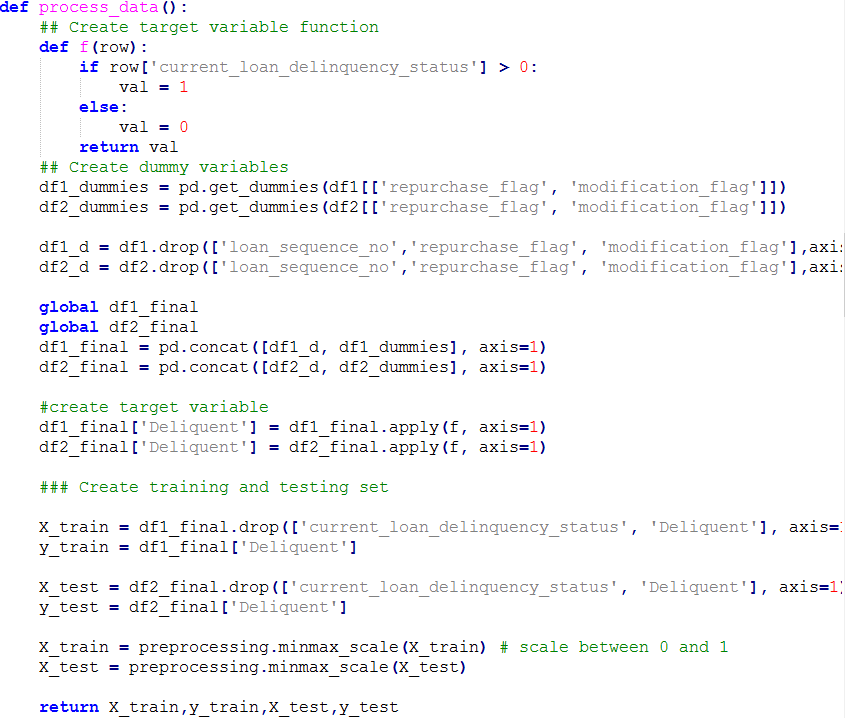
1. Load\_Data\_df() method loads the data in the data frame



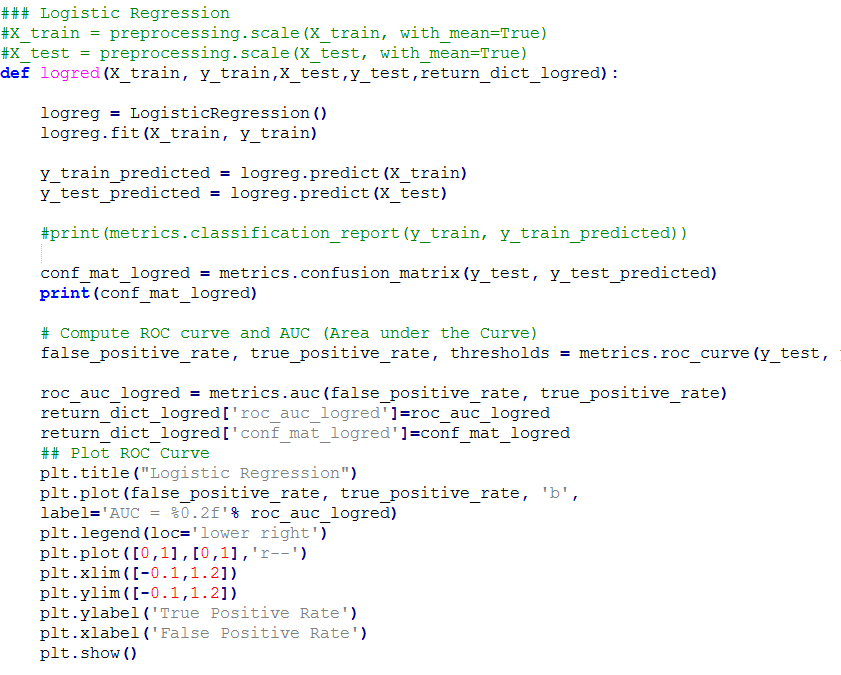
1. Remove\_nan() method replaces nan values and converted categorical variables into ‘Category’



1. Process\_data() method changes the numerical features to Dummy variables, creates the target variable, create Training and Testing datasets. The variables are scaled in the range of 0 to 1.



1. Logred() method generates model for Logistic Regression

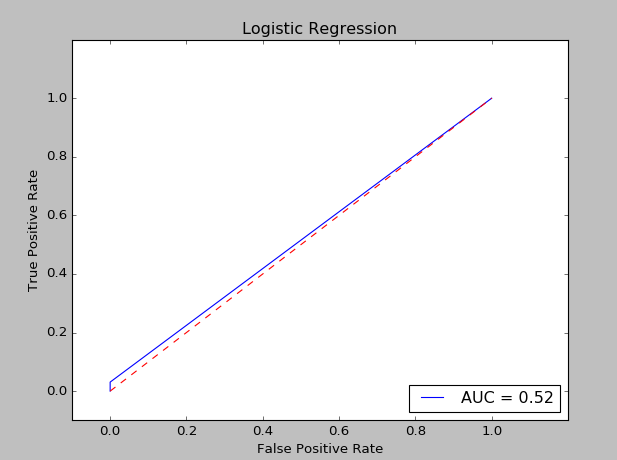


Confusion Matrix for Q12005

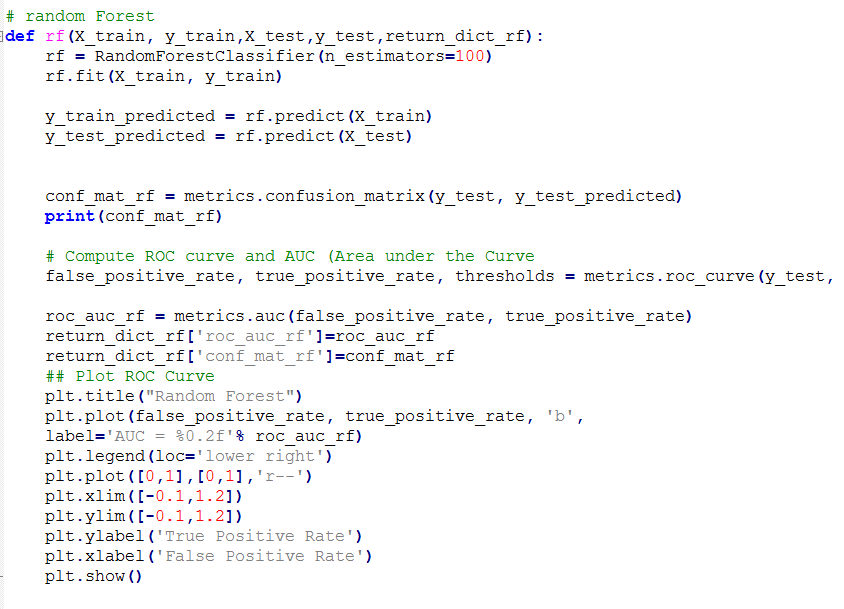
[[191670 2]

[ 8072 256]]

ROC curve for Q12005:



1. Rf() method generates model for Random Forest

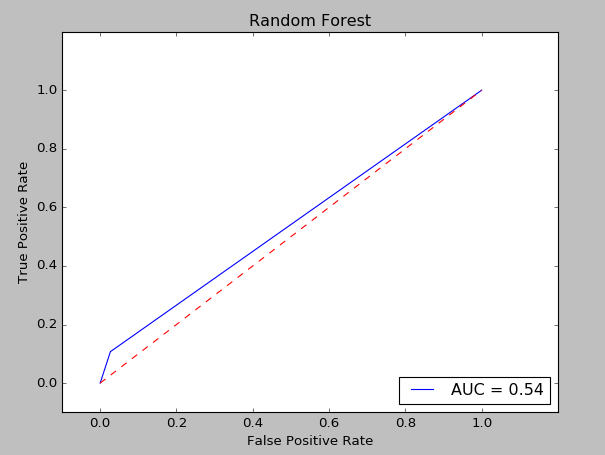


Confusion Matrix for Q12005

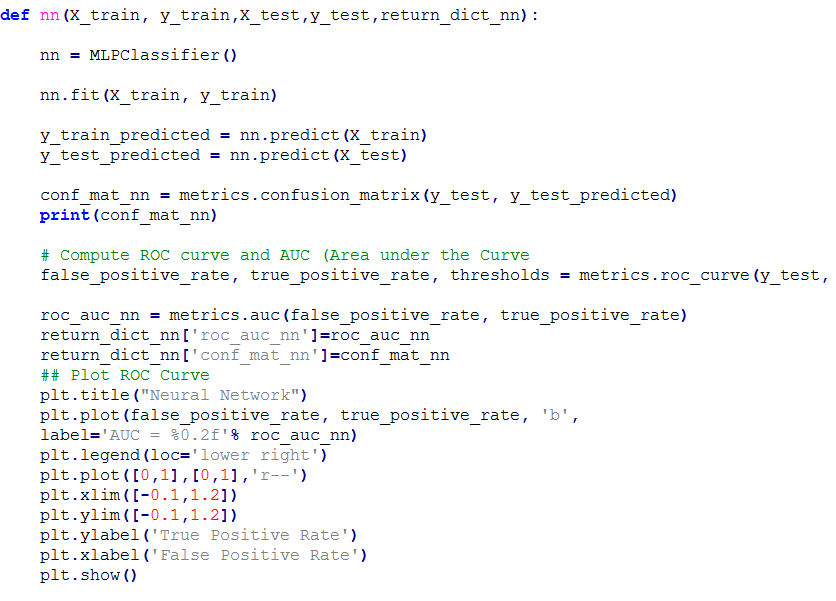
[[191621 51]

[ 8068 260]]

ROC curve for Q12005



1. Nn() method generates model for Neural Network MLPClassifier

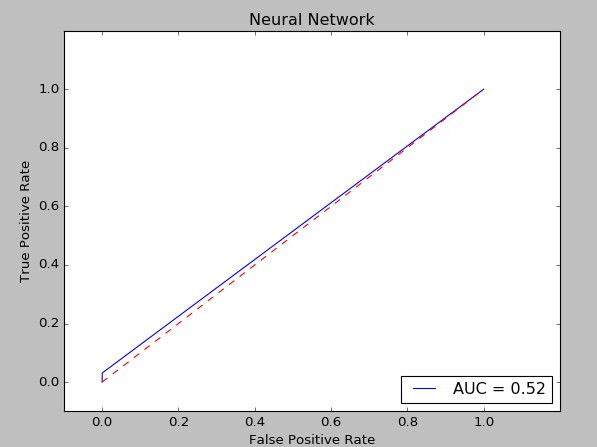


Confusion Matrix for Q12005

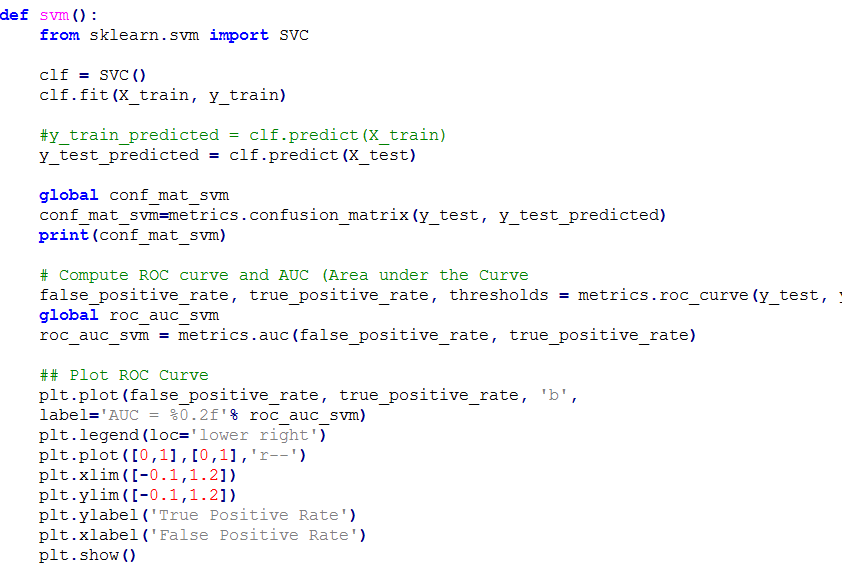
[[186518 5154]

[ 7434 894]]

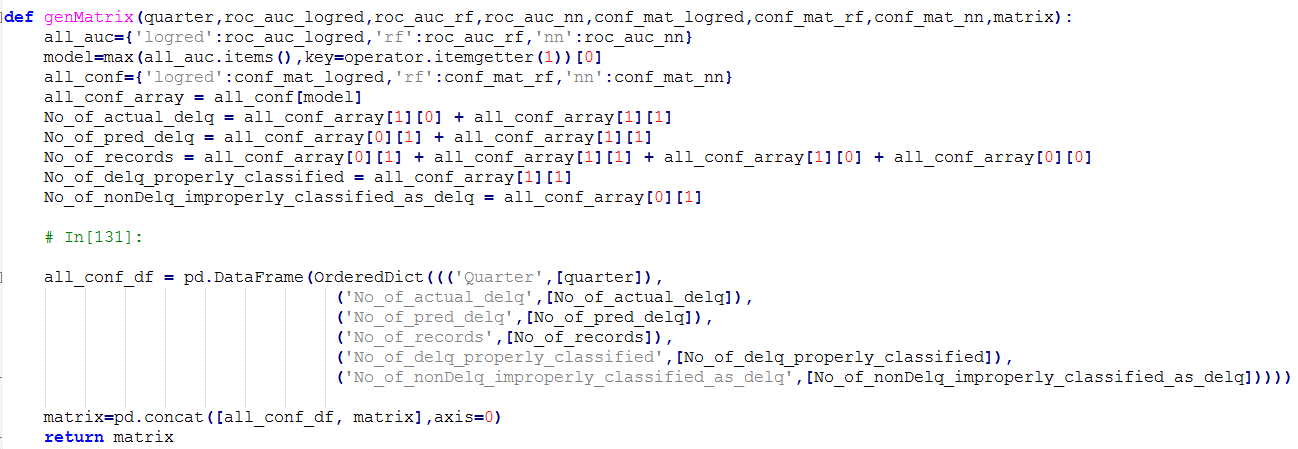
ROC curve for Q12005



1. Svm() method generates model for Support Vector Machine (SVM)



1. Generates the matrix by determining the best model among the four



Output for args: Q11999 Q41999 (Parallel Computation Usage)



* **Metrics for algorithms, for input quarter Q12016**

Logistic Regression:

[[39921 0]

[ 59 20]]

Logred precision recall f1-score support

0 1.00 1.00 1.00 39921

1 1.00 0.25 0.40 79

avg / total 1.00 1.00 1.00 40000

Random Forest:

[[39911 10]

[ 59 20]]

RF precision recall f1-score support

0 1.00 1.00 1.00 39921

1 0.67 0.25 0.37 79

avg / total 1.00 1.00 1.00 40000

Neural Network:

[[39921 0]

[ 59 20]]

NN precision recall f1-score support

0 1.00 1.00 1.00 39921

1 1.00 0.25 0.40 79

avg / total 1.00 1.00 1.00 40000

# **Running Various Scripts in Docker**

docker pull aashritandon/midtermteam9

**For Data Wrangling**:

docker run aashritandon/midtermteam9 python Part1.py 'USERNAME'

'PASSWORD'

**For Prediction**:

docker run aashritandon/midtermteam9 python Part2\_Prediction.py 'QnYYYY'

docker run aashritandon/midtermteam9 python Part2\_Prediction\_finanCrisis.py 'YEAR'

**For Classification**:

docker run aashritandon/midtermteam9 python Classification.py 'QnYYYY'

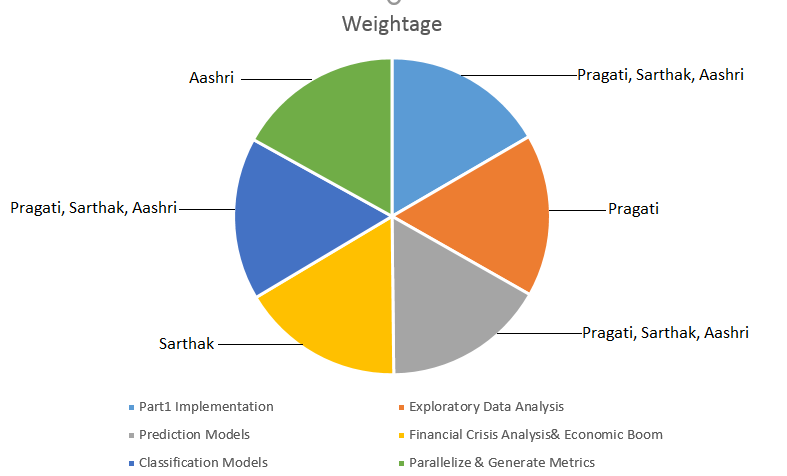
or

docker run aashritandon/midtermteam9 python Classification.py 'QnYYYY' 'QnYYYY'

**For Classification Metrics**:

docker run aashritandon/midtermteam9 python AlgorithmMetrics.py

# **Contributions**



# **References**

[https://portal.hud.gov/hudportal/HUD](%20https:/portal.hud.gov/hudportal/HUD%20)

[http://www.freddiemac.com/news/finance/sf\_loanlevel\_dataset.html](%20http:/www.freddiemac.com/news/finance/sf_loanlevel_dataset.html%20%20)

[https://www.stlouisfed.org/financial-crisis/full-timeline](https://www.stlouisfed.org/financial-crisis/full-timeline%20)

[https://www.thebalance.com/stock-market-returns-by-year- 2388543](https://www.thebalance.com/stock-market-returns-by-year-%202388543%20)

[http://www.davekuhlman.org/python\_multiprocessing\_01.html](http://www.davekuhlman.org/python_multiprocessing_01.html%20)

[https://en.wikipedia.org/wiki/Confusion\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix%20)