**SUMMARY**

The report summarizes the analysis performed on Single-family loan data, provided by Freddie Mac(<http://www.freddiemac.com/news/finance/sf_loanlevel_dataset.html>). The data provided by Freddie Mac consists of the following details:

*Mortgages originated from January 1, 1999, through the “Origination Cutoff Date”, with monthly loan performance data through the “Performance Cutoff Date,” that were sold to Freddie Mac or back Freddie Mac Participation Certificates (PCs).*

* Fully amortizing 15-, 20-, and 30-year fixed-rate mortgages*

* Mortgages categorized as having verified or waived documentation.*

We then build predictive analytics models using the datasets. The problem presented is divided into 2 sections:

**Section 1: Data wrangling**

* Data Download and pre-processing
* Exploratory Data analysis

**Section 2: Building and evaluating models**

* **Prediction** using Linear Regression, Random Forest, Neural Network KNN Algorithms
* **Classification** using Logistic Regression, Random Forest, Neural Network, SVN Algorithms

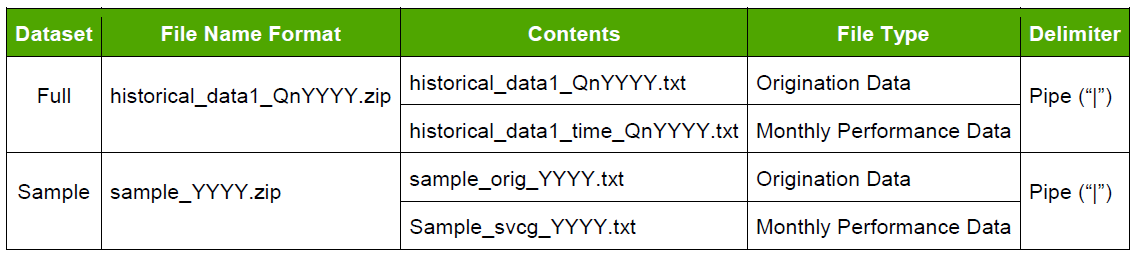
**1. Part I: Data Ingestion and Wrangling**

**1.1 THE DATA**

***Single Family Loan-Level Dataset***

For each calendar quarter, there is one file containing loan **origination data** and one file containing **monthly performance data** for each loan in the **origination data file**.

Freddie Mac has created a smaller dataset for those who may not require, or have the capability, to download the full Dataset. The sample dataset is a simple random sample6 of 50,000 loans selected from each full vintage year and a proportionate number of loans from each partial vintage year of the full Single-Family Loan-Level Dataset. Each vintage year has one origination data file and one corresponding monthly performance data file, containing the same loan-level data fields as those included in the full Dataset. Due to the size of the dataset, the data has been broken up and compressed as detailed below. The files are organized chronologically by year and quarter.



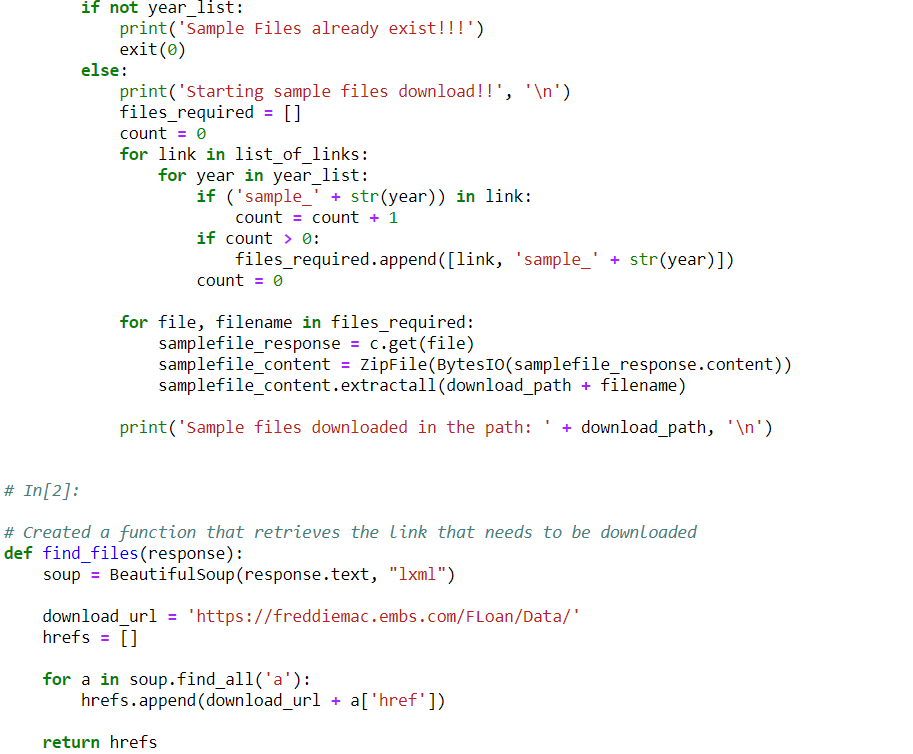
**Data Download and pre-processing:**

The very first challenge was to programmatically download the data from Freddie Mac website (<https://freddiemac.embs.com/FLoan/Data/download.php>) and download and preprocess the “SAMPLE” file both for origination and performance data.

To download the file programmatically, first the user should register him/herself by creating username and password. Once logged in, the user can download all the file required for analysis. We have used the python requests library for this purpose. To store the user credential, we need to store them in the request session so that user didn’t redirect back to the login page whenever he/she required to download a file from the Freddie Mac posted dataset.



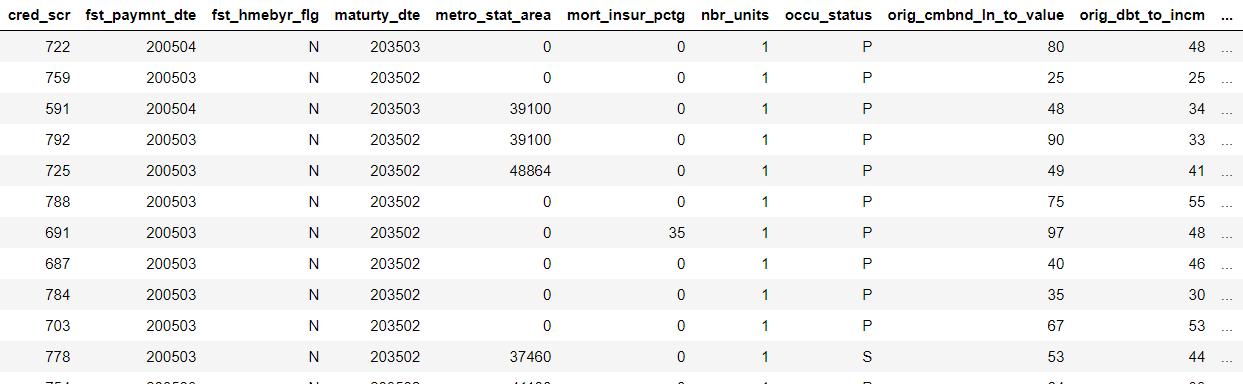
Once, the user is logged in to the website, we will be using request session to drive the further functionality. We will be using **Beautiful Soup** package, a powerful python package for data scrapping from the Freddie Mac Website and download all the “**Sample**” files for our analysis purpose.



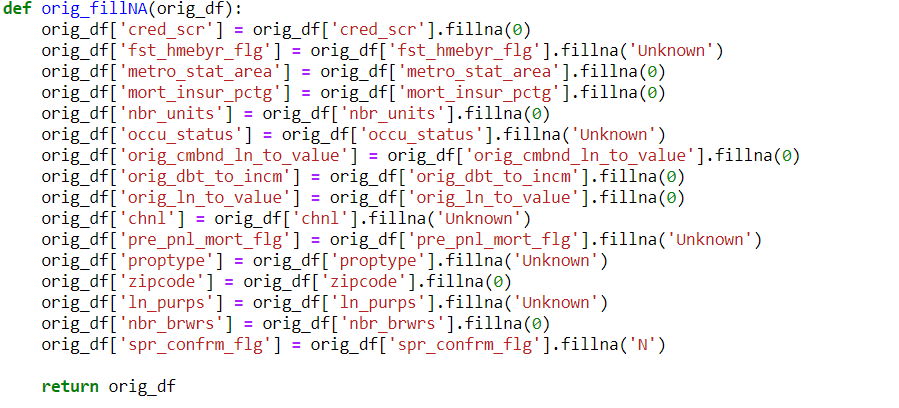
**1.2 Data Preprocessing and Cleaning**

Since, these files are big in size and consist of huge amount of data, we need to preprocess these files before getting saved in our drive. These Zip file consist of two files:

**Origination File:**

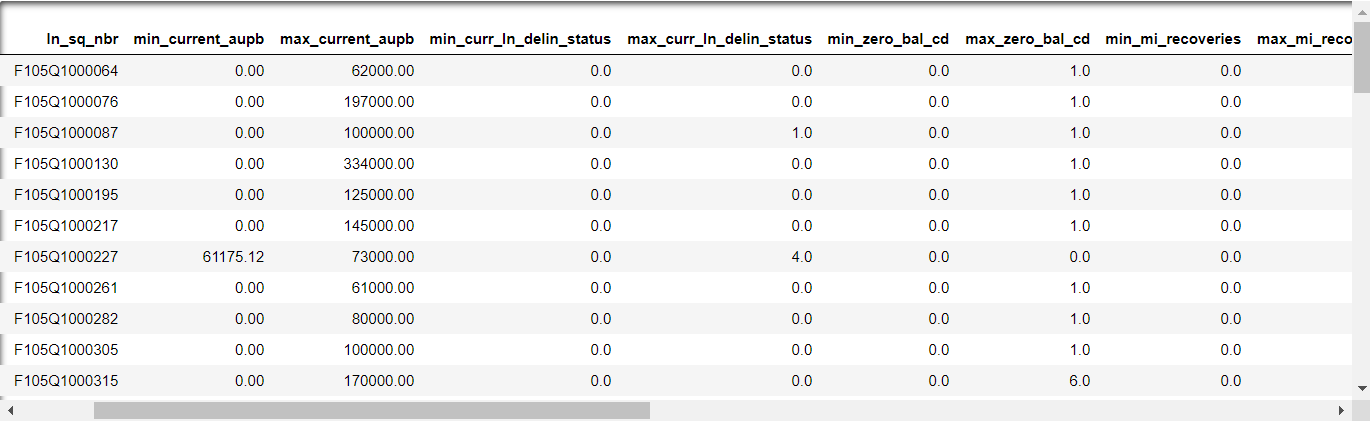
For origination file, we first analyze the file size for all the year. Using python pandas, we create a data frame where we append all the data from the sample file for all the year. Origination file consist of 27 columns which consist of various details associated with the loan originated in each year. Some of the major columns are defined below:

We have many Null values and spaces (an invalid values) which we need to handle before using these files to compute a summary report. We must make sure that our data is in proper format with same datatype. We have created following functions:

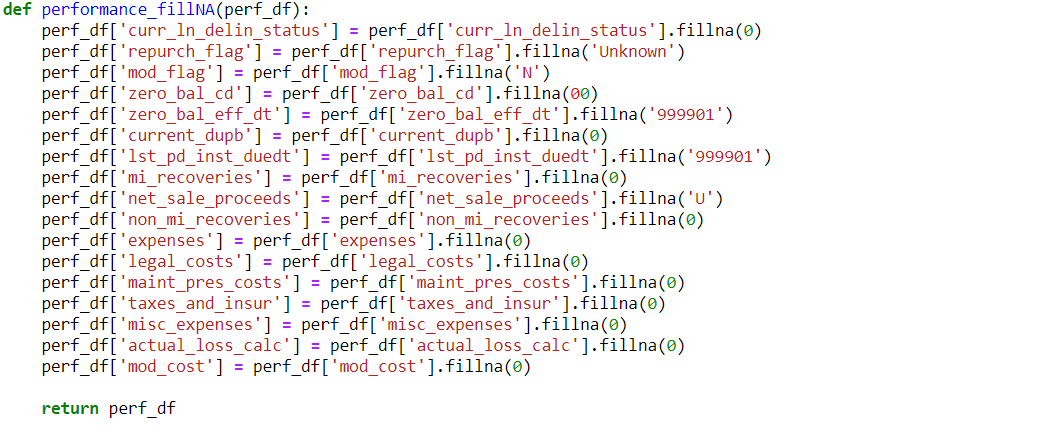


**Performance File:**

For performance file, we first analyze the file size for all the year. Using python pandas, we create a data frame where we append all the data for all the year. Performance file consist of 27 columns which consist of various information about the loan origination in a year. Since, the size of the file is very large, we decide to summarize the input file for all the year during its preprocessing. Some of the major columns are defined below:



As we have many empty column values in our origination and preprocessing file, we need to clean those to ensure that we don’t have any NAN/NA value in our data. Also, we need to take care of the data type of column. These columns will be required while creating the summary matrices.



Once, we are done with the cleaning of the performance file, we will create a summarized version of the file based on certain column which are important for us. For this step, we created a function which will get the Max/Min/Average value of the columns in our summarized performance file.

**1.3 Creating Summarized CSV (Output)**

Once the preprocessing and data cleaning steps are performed, we will have created our final output file, one for origination file named ‘OriginationCombined’ and for performance file named ‘PerformanceCombined. These final files will be used for our analysis performed in part 2.



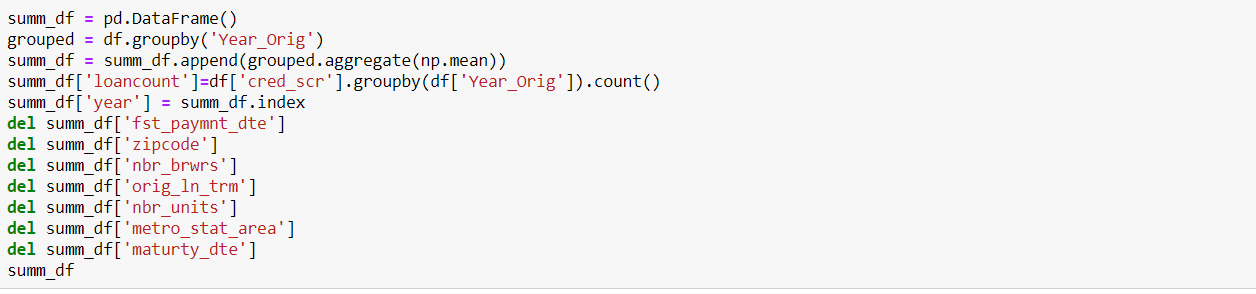
**NOTE:** As we are working on two different files, we need to know about the data and the relationship between the two files created. In origination file, we have **ln\_sq\_nbr** which is a unique loan sequence number with quarter and year of loan origination attached to it. In performance file, for a given year we have multiple rows associated with a loan number which depicts its performance. We don’t have any loan number duplicated in origination file with respect to year. It is always **Unique**.

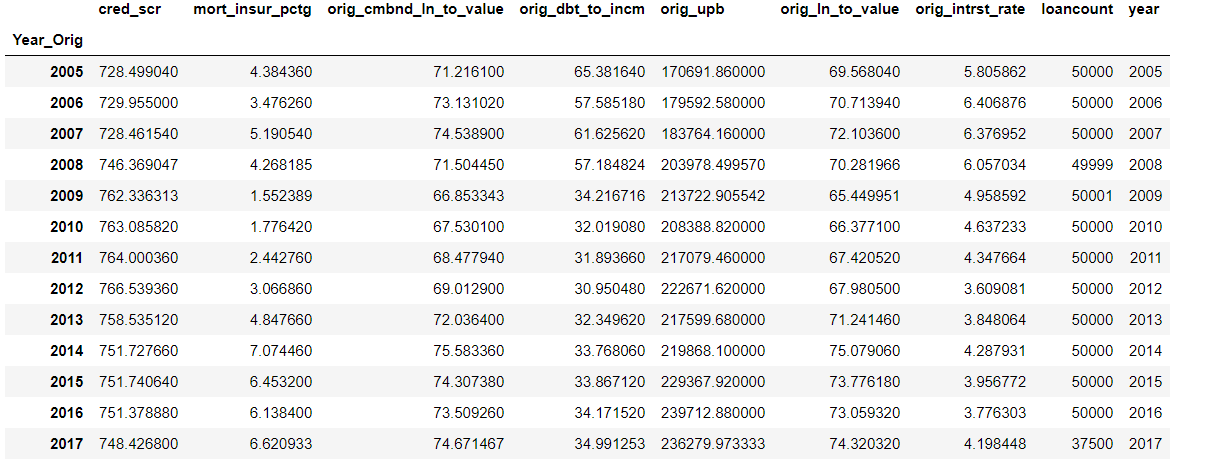
**PART 2: Exploratory Data Analysis**

**2.1 Analysis – Jupyter Notebook**

In Part 2, we were asked to write Jupyter notebook using R/Python to graphically represent different summaries of data and summarize our findings in this notebook.

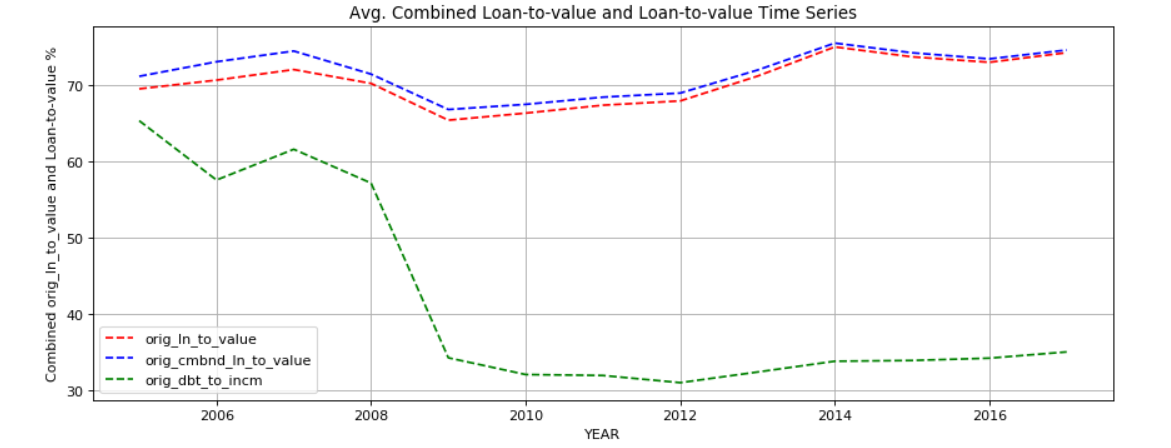
We first create a pandas’ data frame for the origination file and group the data on the year.

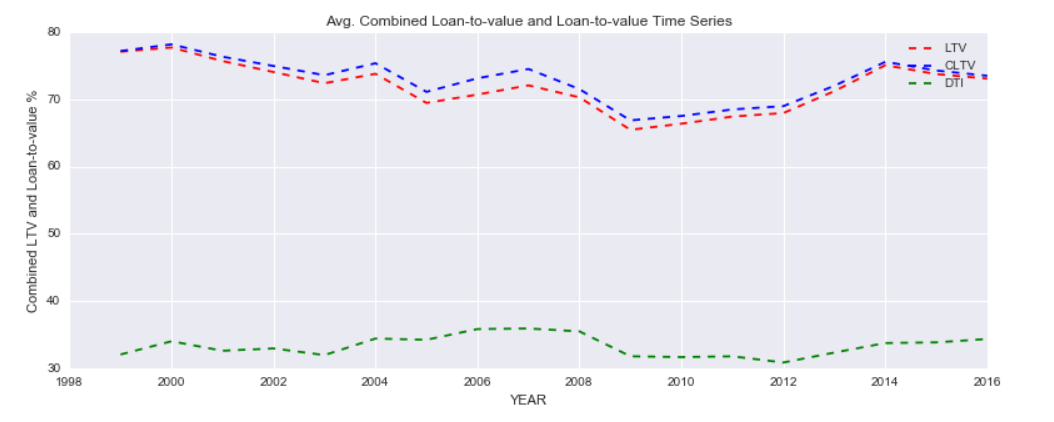




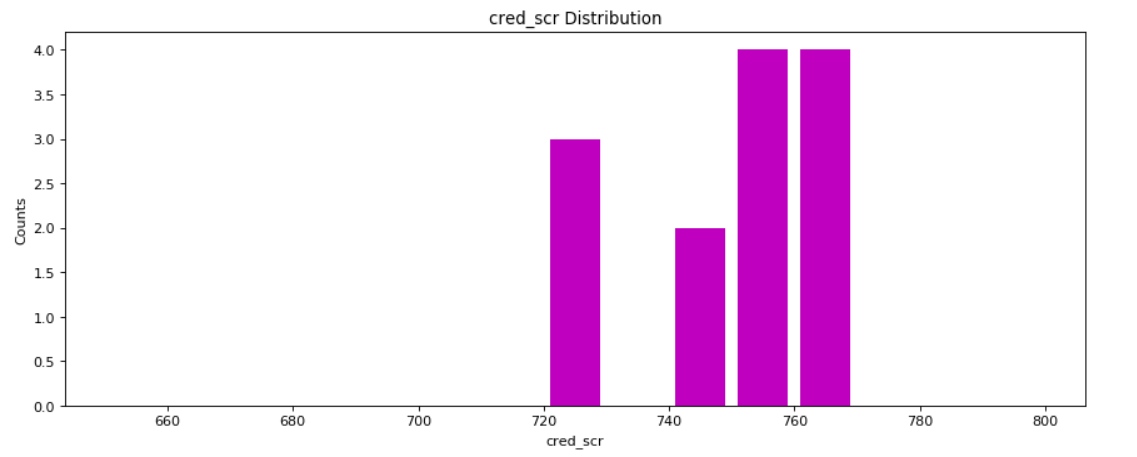
Here we show the various details associated with the origination file and the total loan count and mean of various important factor like credit score, interest rate based on year.

**CLTV – LTV & DTI Comparison based on Year**

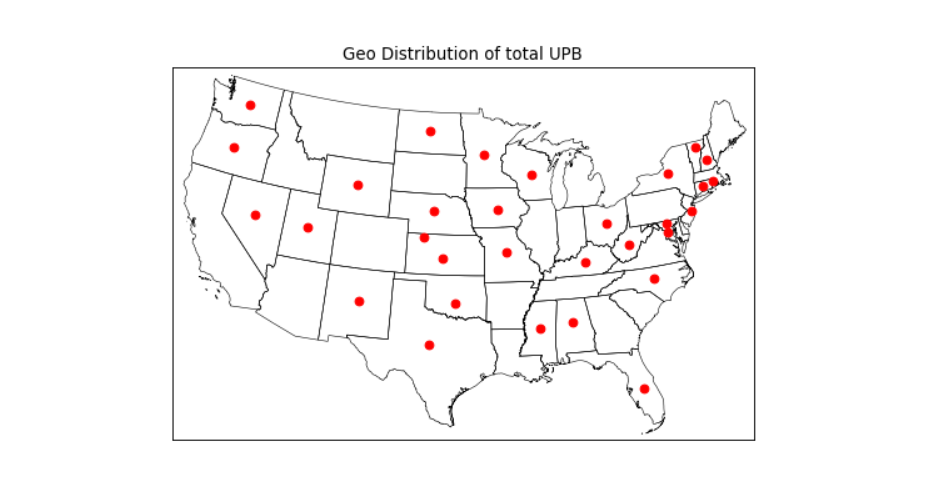




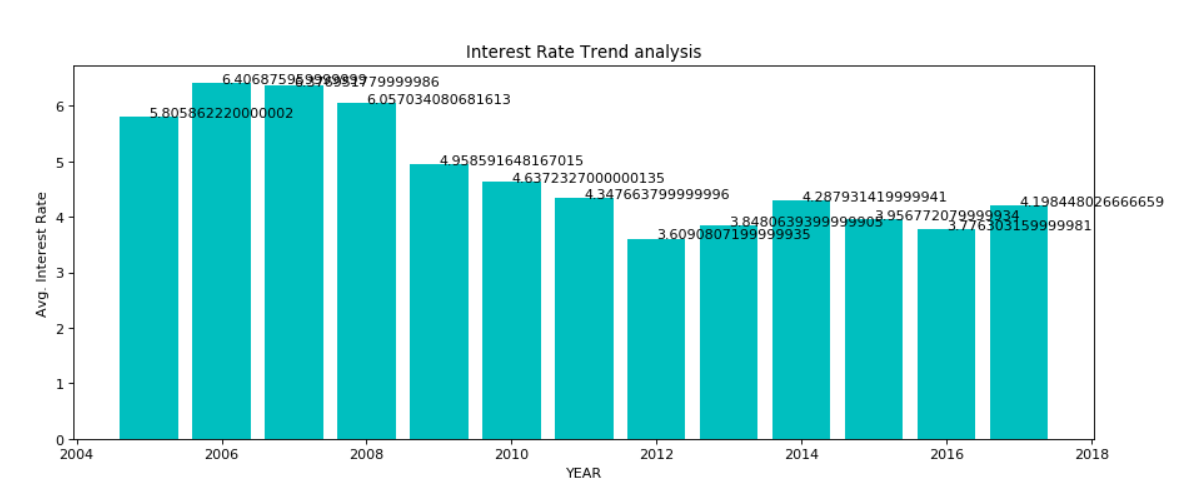
**FICO Based on Count**

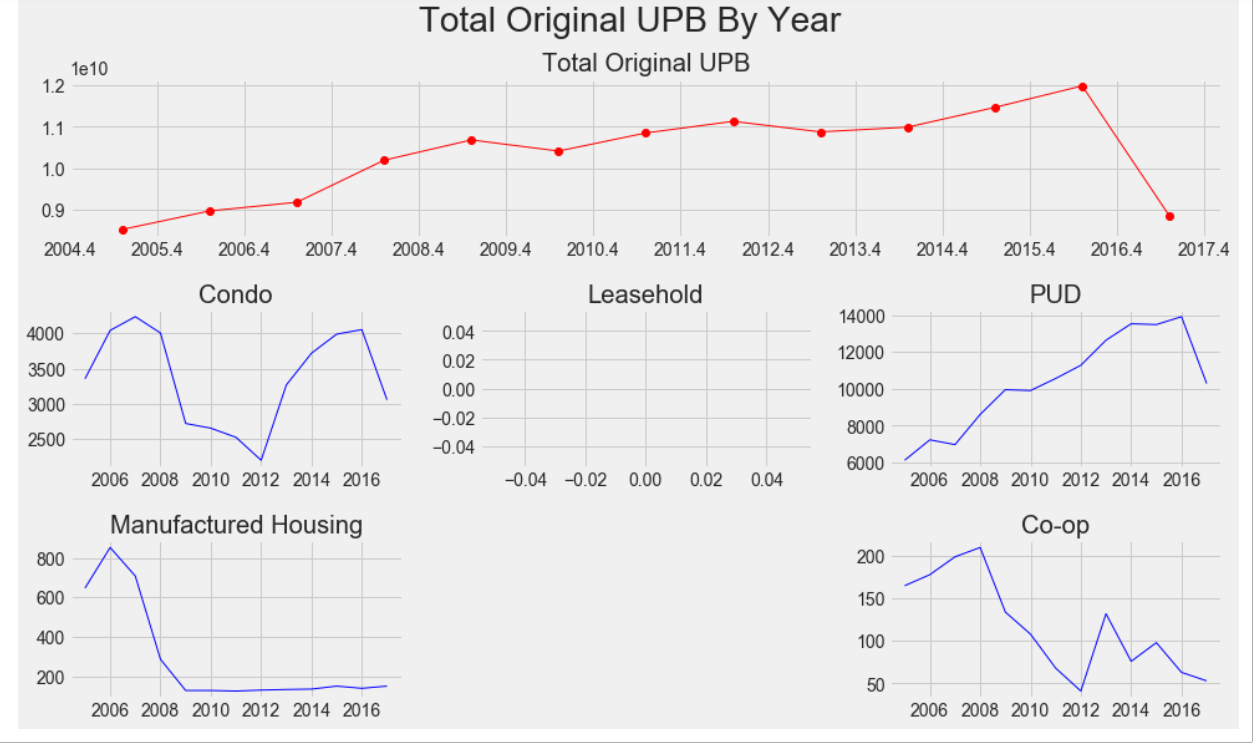


**Count of Loan based on Geographical Presence**

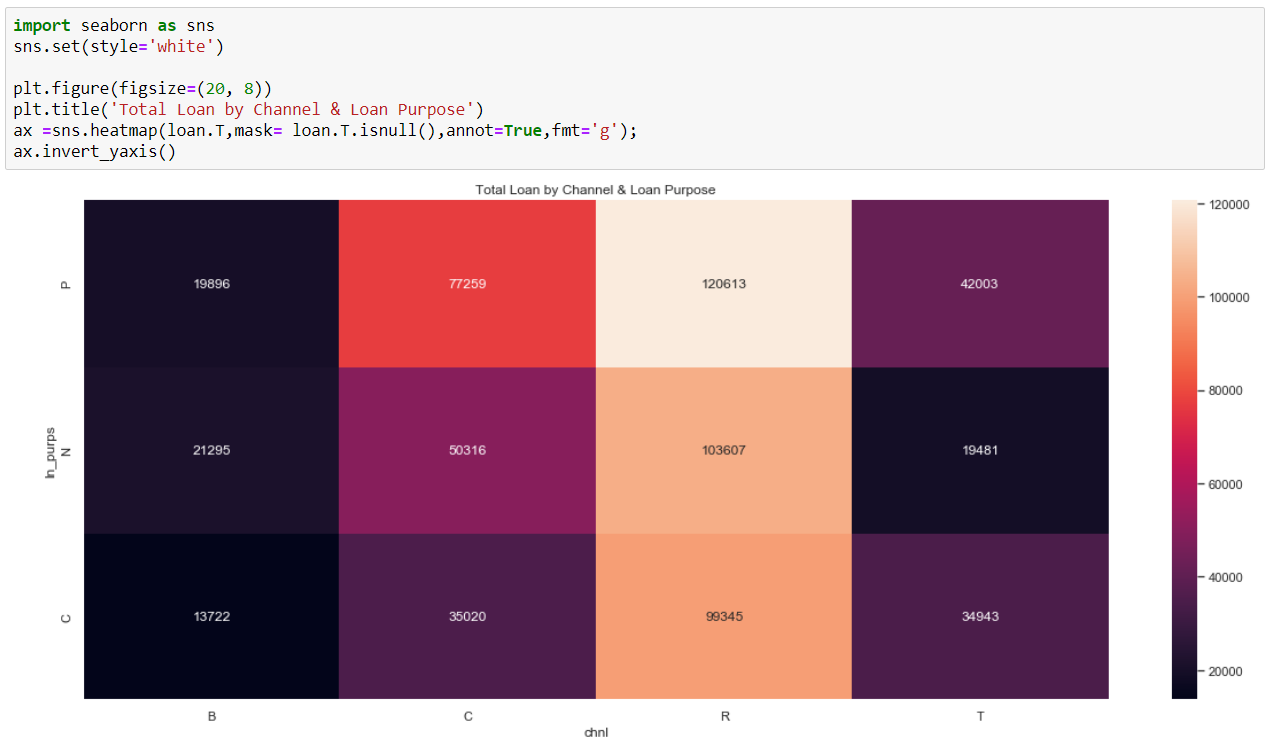


**Average Credit Score based on Year**

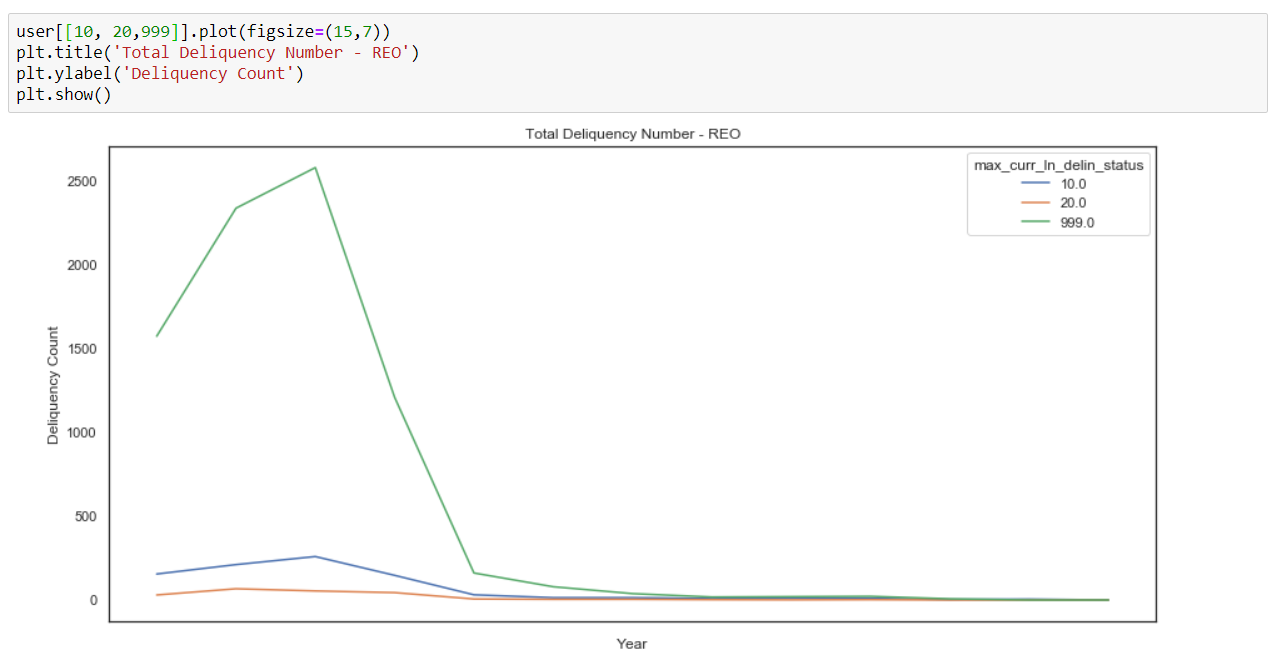


**Average UPB based on Year**

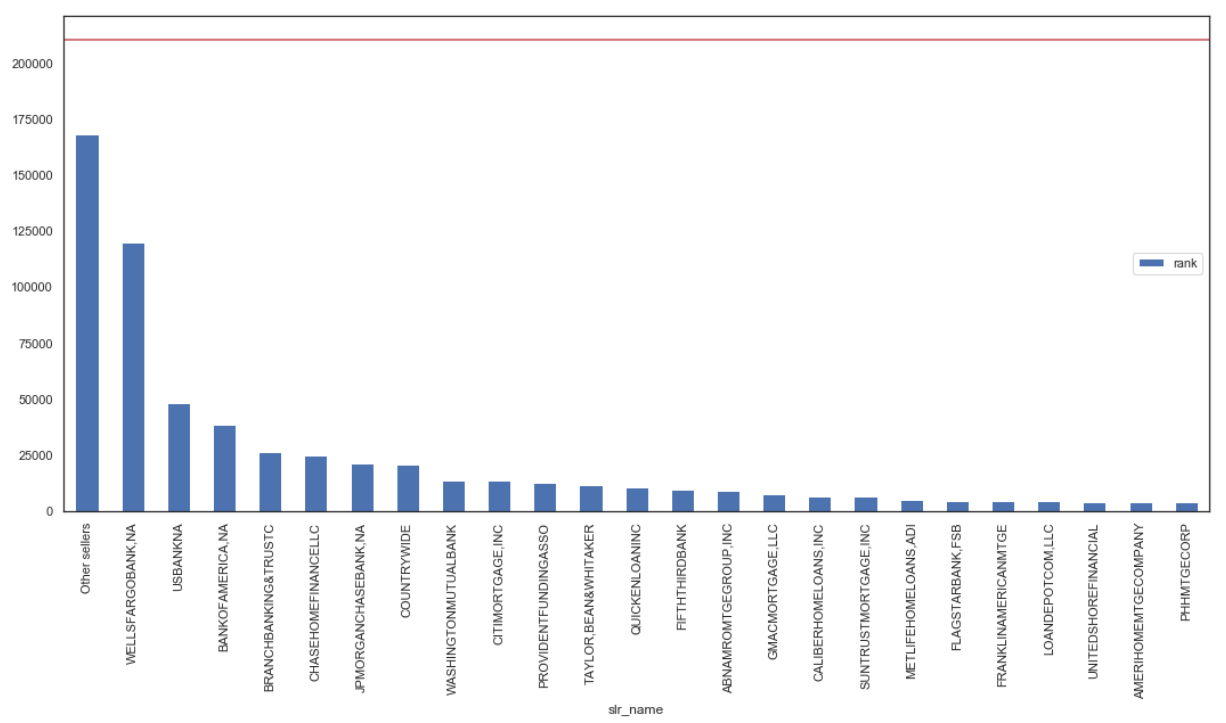
**Total Loan Count by Purpose**



**Total Deposition by Year**



**Top 25 Seller**



**2. Part II: Building and evaluating models**

**PART B- Classification**

In this part In this part we will explore different algorithms to classify loans as delinquent or non-delinquent based on the features existing in the historical performance dataset.

**Part 1- Data Download and cleansing**

For this part, we first begin with downloading the data. For downloading data, we have created a python script which accepts the train quarter as the input and downloads data for that quarter as well as the consecutive quarter.

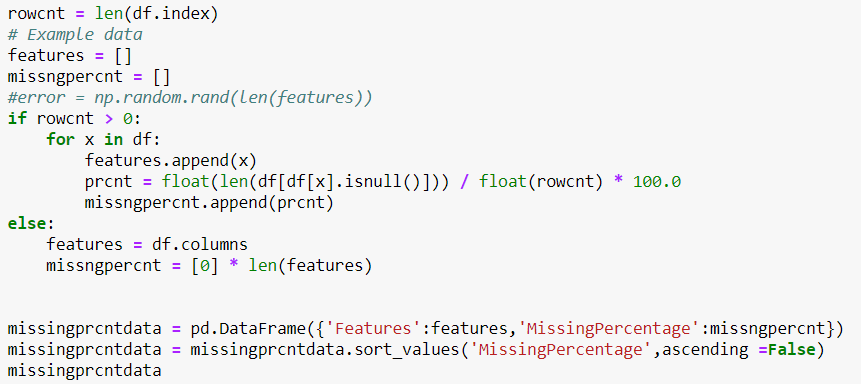
For downloading data first we run the script and save after unzipping the data.



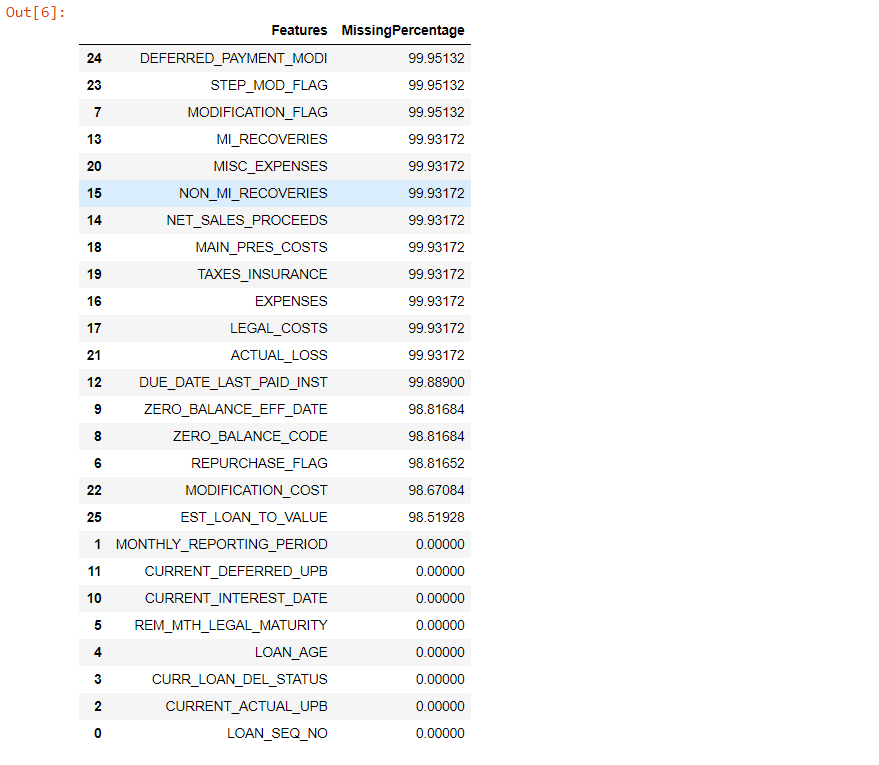
Since the data file is very big. We upload it in chunks in the pandas dataframe.

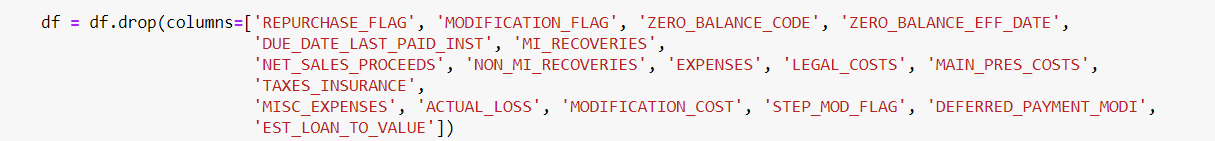


After downloading the data we check for missing values and get the values from following function



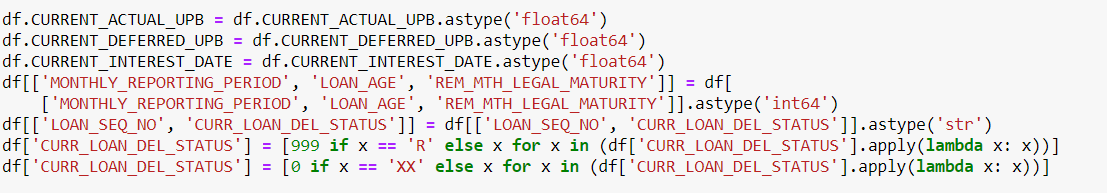
Depending on the following output, we get to know that for top 19 columns around 99 % data is missing. So, we can not use these columns for our classification algorithm.



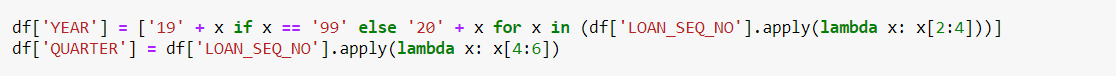


After dropping the unnecessary columns, we clean the remaining data.

First we change the data types of the columns depending upon the values.



Then we add two new columns for year and quarter for future use in classification matrix.



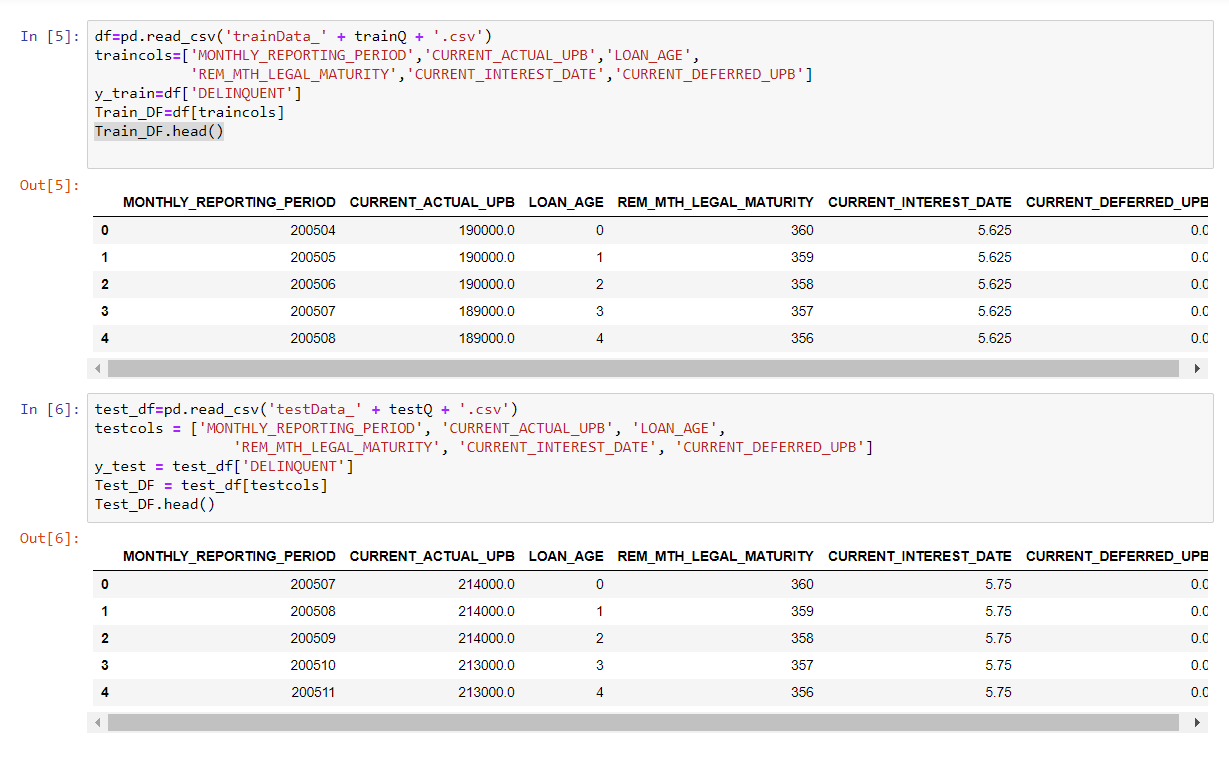
Then we add anew column as DELINQUENT depending upon the values from column - CURRENT LOAN DEL STATUS and then drop this column as this is not required in the classification data.



Now, we have clean data and we are ready for our classification algorithms.

We do the same for testing quarter data and save both in Train\_DF and Test\_DF.

We take column DELINQUENT as our Y variable.



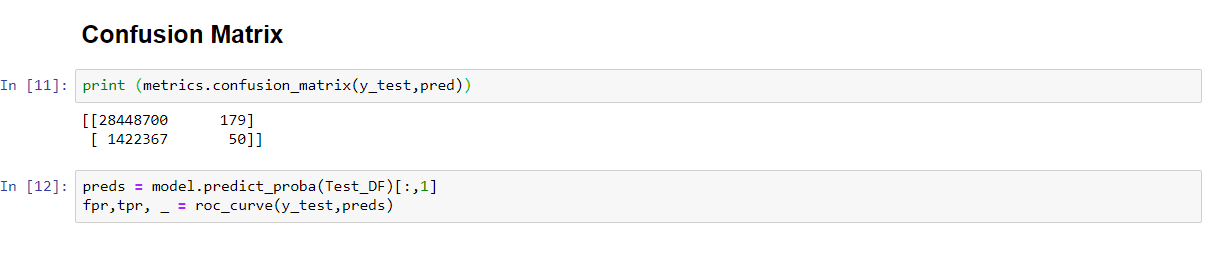
Now we run classification algorithms on these.

**Part 2- Running Algorithms**

**Logistic regression:**

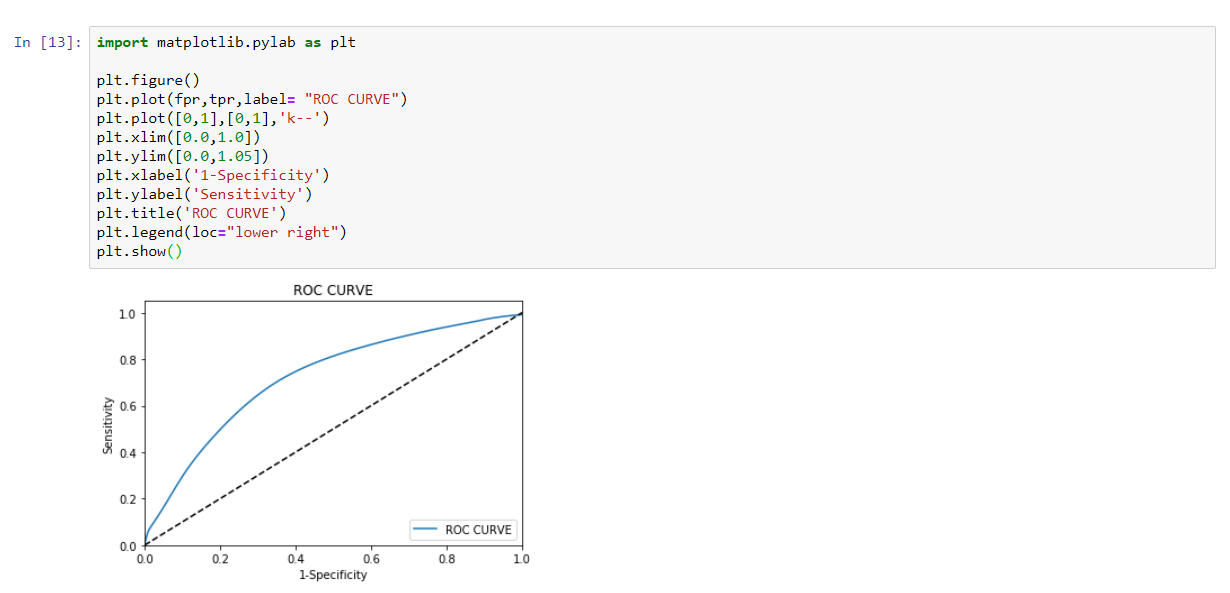
We did not select this model because although we got similar results for the accuracy, we got very different results for the confusion matrix. The number of true positives was lower by a significant amount.



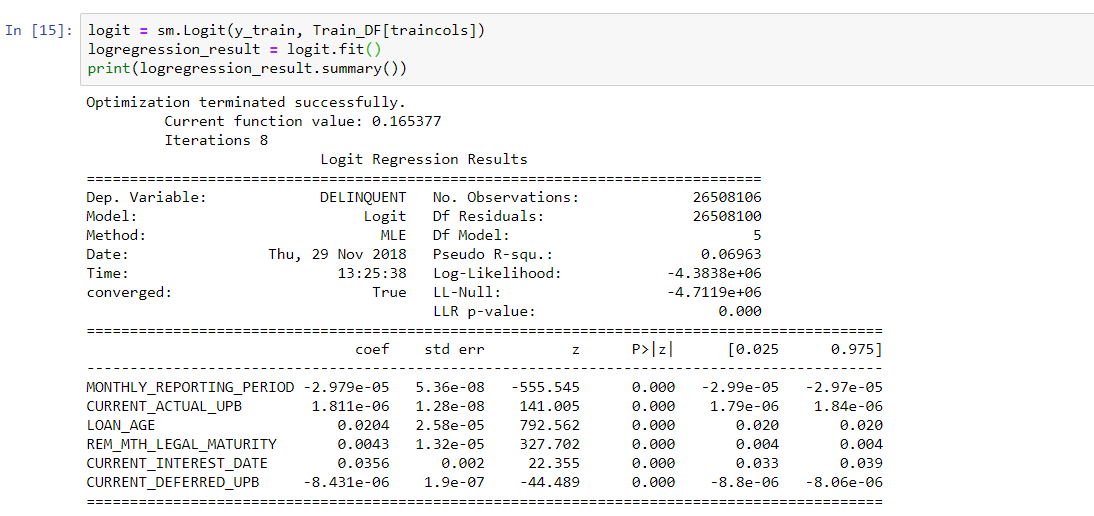


**ROC Curve**

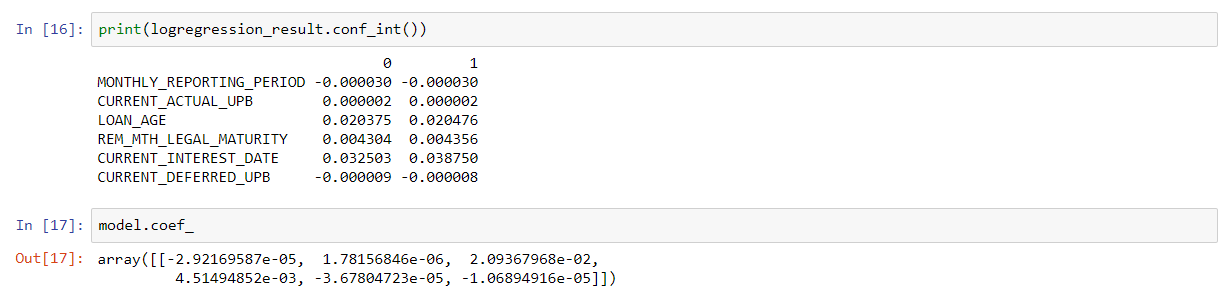
The ROC curve based on this algorithms is:



Also we used logit for detailed analysis.

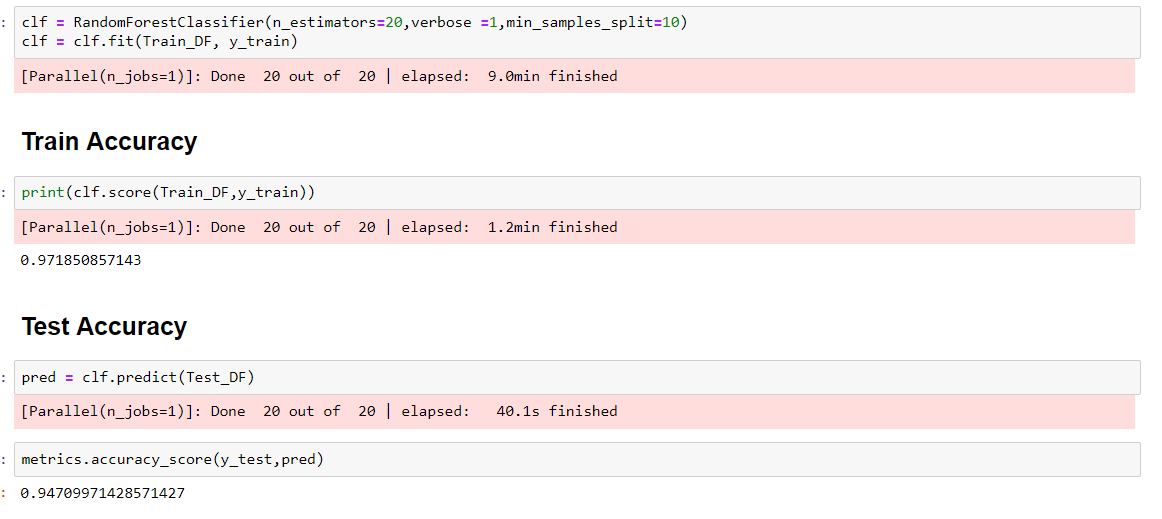


Finally, the values of other coefficients:

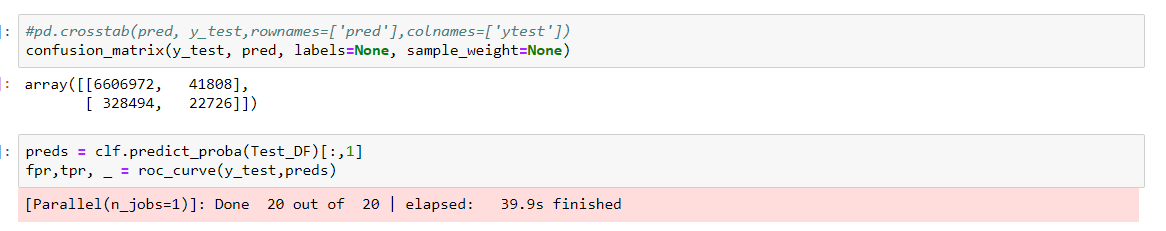


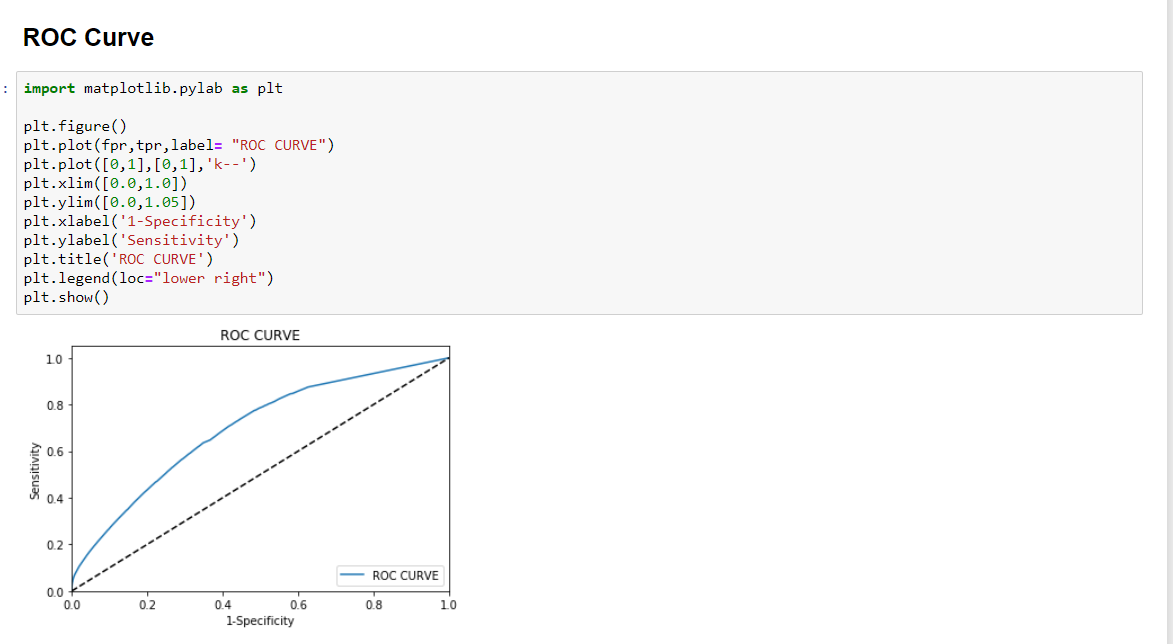
**Random Forest Algorithms:**

**We chose random forest as best algorithm due to its accuracy.**



**Confusion Matrix**





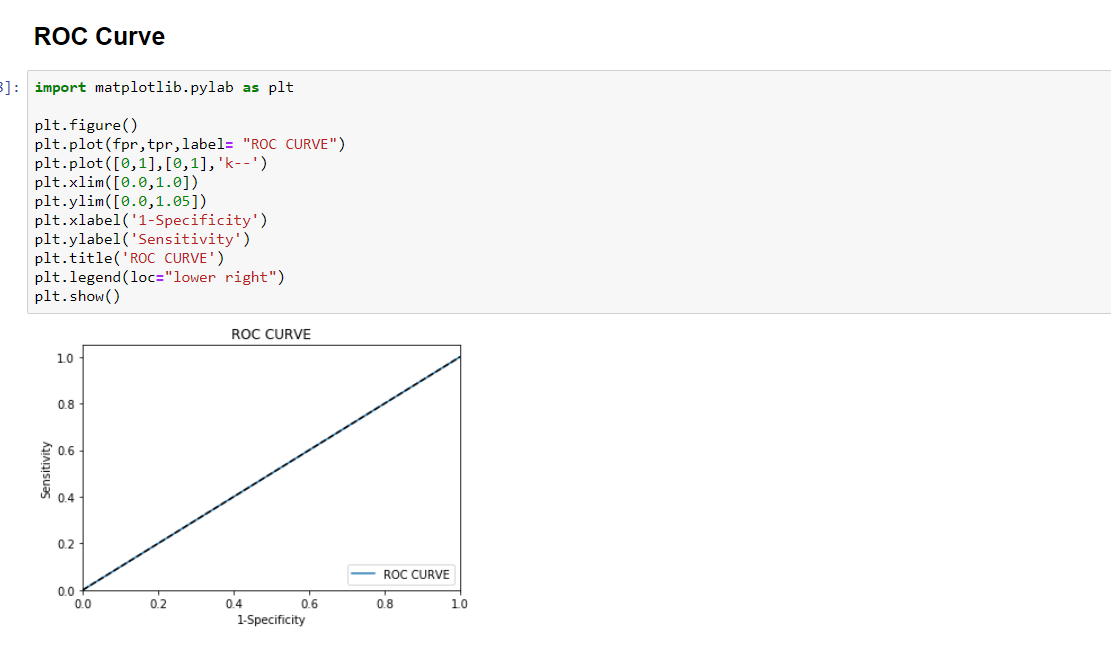
**Neural Networks:**

We tried running this algorithm with different tuning parameters which we tweaked to get results. We tried changing the number of hidden layers and the number of neurons with different learning rates.

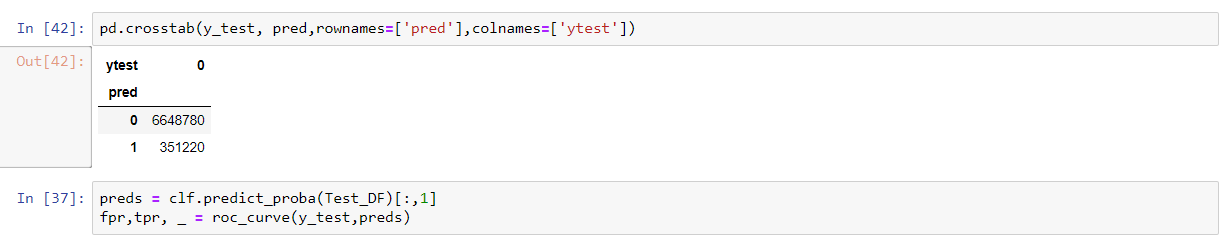
We are getting similar accuracy as the one we got from Random Forest and Logistic Regression, but the Confusion matrix shows very poor results.

Neural Network is not able to identify the delinquent loans and is predicting everything as non-delinquent. That is why we have discarded this algorithm.



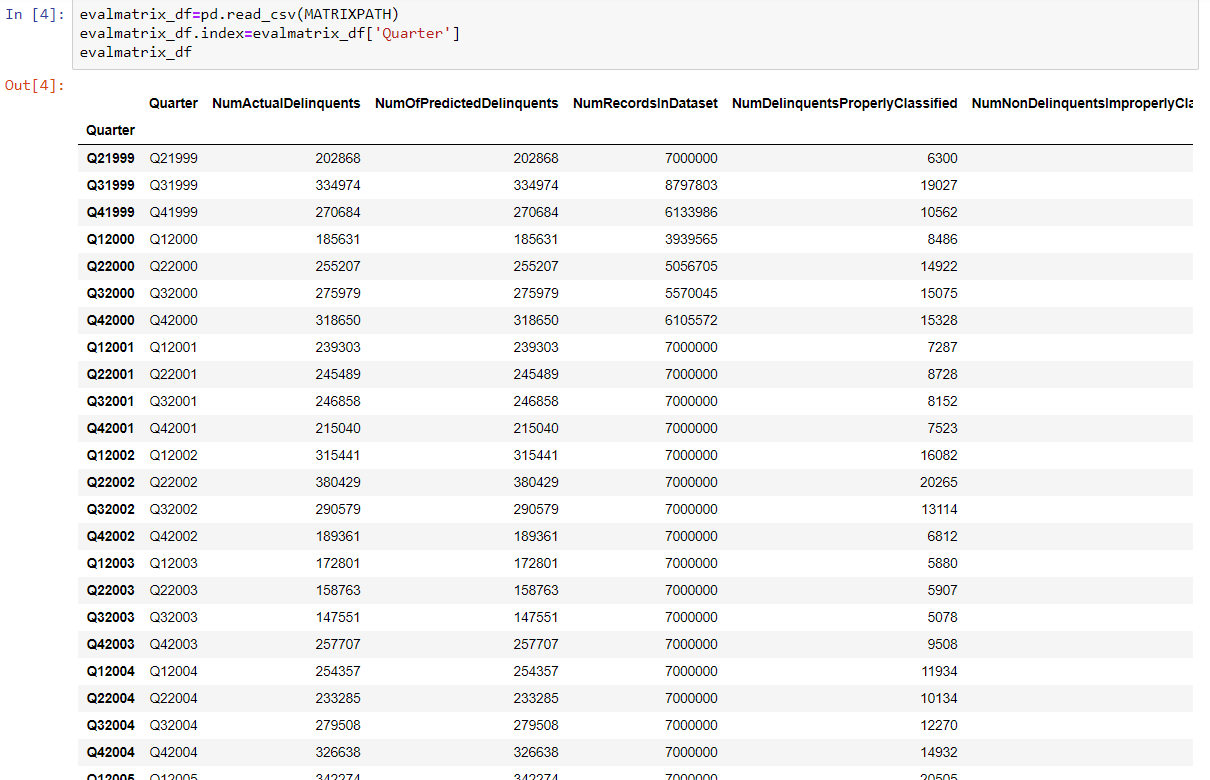


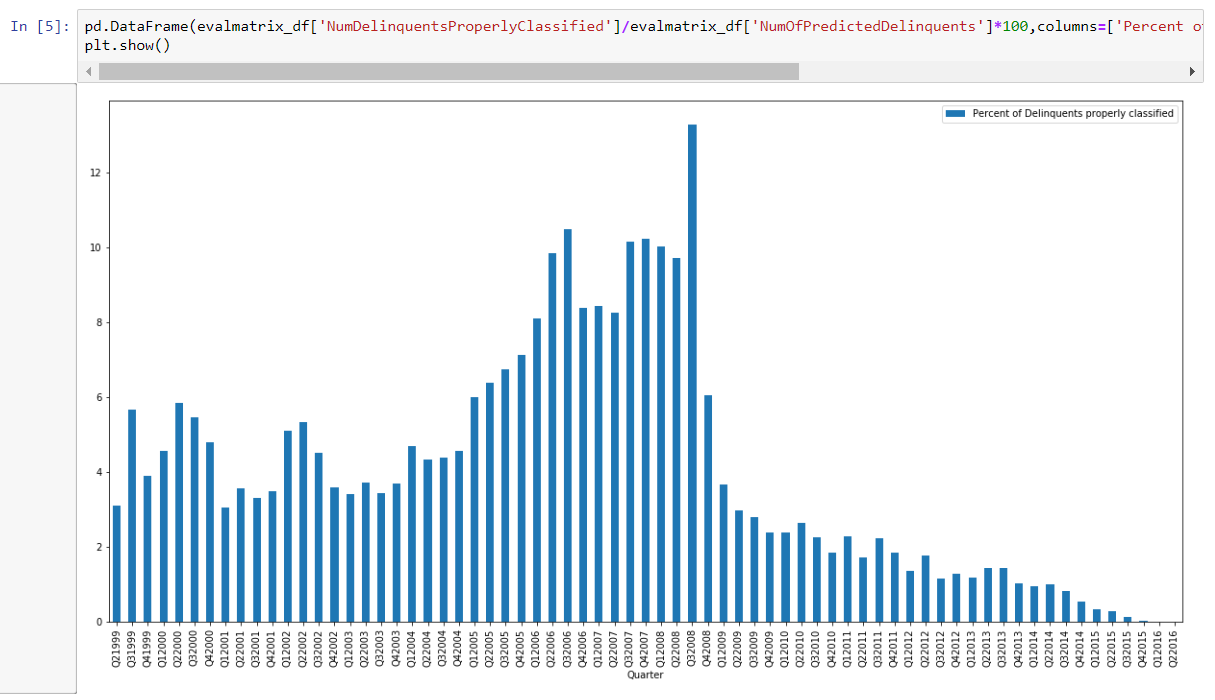
**Confusion Matrix**

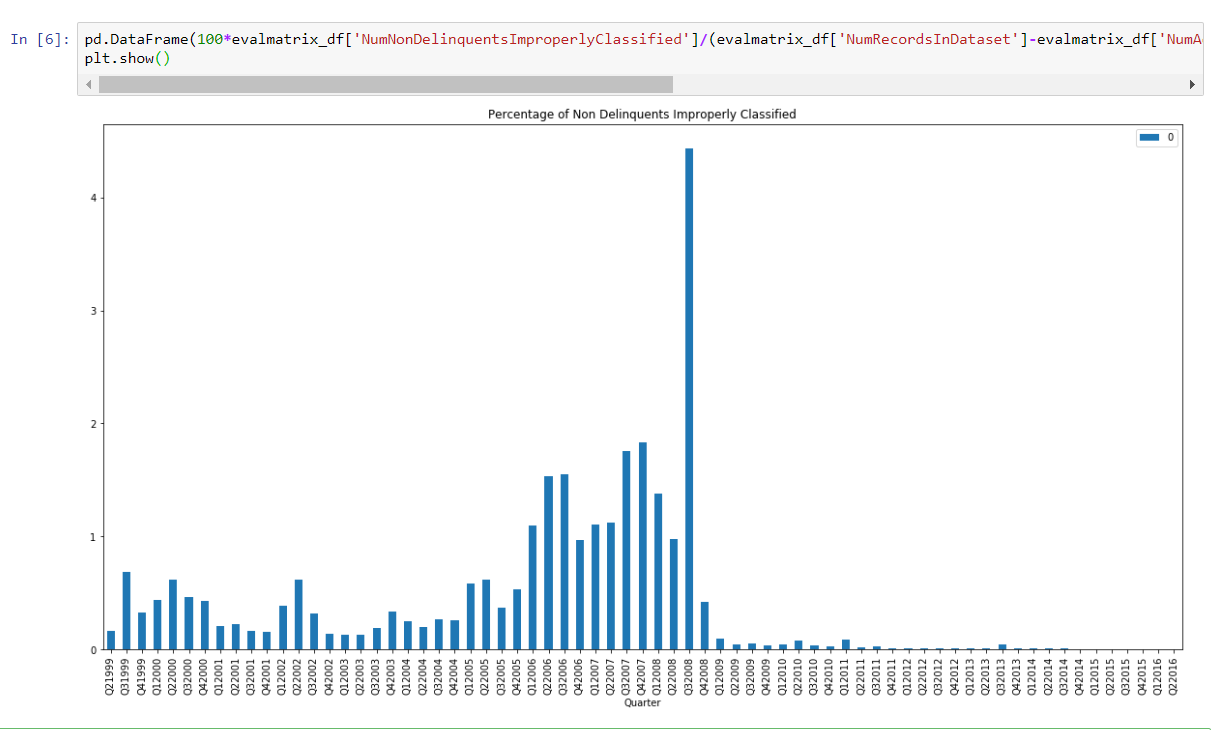


**Evaluation Matrix**

We downloaded the data from Q11999 to Q22016 and saved the evaluation matrix based on delinquents.







**Classification Summary**

**Justification on the chosen classification model**

• Similar score for all the algorithms was obtained.

• Random forest gives the best results in terms of the confusion matrix.

• Random Forest takes lesser time to train as compared to the Neural Network.

• Although the area under the ROC curve for Logistic regression is more, we have chosen the Random Forest algorithm because the number of True positives is significantly higher.

**What can be done better**

• Although we can see that the random forest provides reasonable outputs for this type of data. We can perform further analysis with more computational power to arrive at better results.

• The dataset is a biased one meaning that there are significantly more number of N values for delinquents as compared Y values. Because of which, the algorithm has a lesser True positive rate.

• We can see from the evaluation that, the algorithm performs better when trained with more delinquent data. We can re-develop this algorithm and fine tune the tree parameters for the Q12007-Q42008 period which might enable us to create an algorithm with higher accuracy as far as the confusion matrix is concerned.

• We can explore more algorithms for classification such as KNN or SVM to check the accuracy.

• There might be an optimum parameter for Neural Network that exists which we haven’t come across in our analysis.

**Summary**

We have analyzed the data given in the Freddie Mac Single Family Loans Dataset. We have seen different parameters related to the origination and performance files. These have been summarized to perform EDA on the data. Data has been automatically downloaded using HTML scraping techniques. We have performed Lasso Regression, recursive feature elimination, F regression and PCA for performing feature extraction and selection. We have analyzed different algorithms such as Regression, Random Forest and Neural Networks to perform Predictive and Classification modelling and tested the algorithm on Quarters from 1999-2016 in a rolling quarters manner. According to our analysis, Random Forest was the algorithm of choice for Prediction as well as Classification. We have done extensive analysis for the period 2007-2009 to analyze different aspects of the market crash