Advances in Data Science & Architecture

Mid Term Project Report

Sumit Deo

Priyanjna Sharma

Gautam Pawar

Contents

[1 Part1: Data Wrangling 3](#_Toc477558278)

[1.1 Running Part 1 from docker image 3](#_Toc477558279)

[1.2 Data Download and pre-processing 3](#_Toc477558280)

[1.2.1 Programmatically Downloading data 4](#_Toc477558281)

[1.2.2 Data Cleaning and Pre-Processing 4](#_Toc477558282)

[1.2.3 Parallel Processing for Summary Metrics 5](#_Toc477558283)

[1.2.4 Summary Metrics 7](#_Toc477558284)

[1.3 Exploratory Data Analysis 9](#_Toc477558285)

[1.3.1 Summary Plots 10](#_Toc477558286)

[1.3.2 Tableau Dashboards 17](#_Toc477558287)

[2 Part2: Building and Evaluating Models 27](#_Toc477558288)

[2.1 PREDICTION 27](#_Toc477558289)

[2.1.1 REGRESSION 27](#_Toc477558290)

[2.1.2 Exhaustive Search Variable Selection 27](#_Toc477558291)

[2.1.3 Forward Selection 30](#_Toc477558292)

[2.1.4 Backward Selection 30](#_Toc477558293)

[2.1.5 Backward regression 31](#_Toc477558294)

[2.1.6 Stepwise regression 31](#_Toc477558295)

[2.1.7 Regression Model 33](#_Toc477558296)

[2.1.8 KNN Algorithm 34](#_Toc477558297)

[2.1.9 Neural Network Model 35](#_Toc477558298)

[2.1.10 Random Forest Algorithm 36](#_Toc477558299)

[2.1.11 Summary: 38](#_Toc477558300)

[2.1.12 Financial Crisis 39](#_Toc477558301)

[2.2 Classification 39](#_Toc477558302)

[2.2.1 Data Cleaning and Preprocessing: 39](#_Toc477558303)

[2.2.2 Logistic Regression Model: 40](#_Toc477558304)

[2.2.3 Random Forest Model: 43](#_Toc477558305)

[1.1 Contribution 48](#_Toc477558306)

[48](#_Toc477558307)

[3 References 49](#_Toc477558308)

# Part1: Data Wrangling

## Running Part 1 from docker image

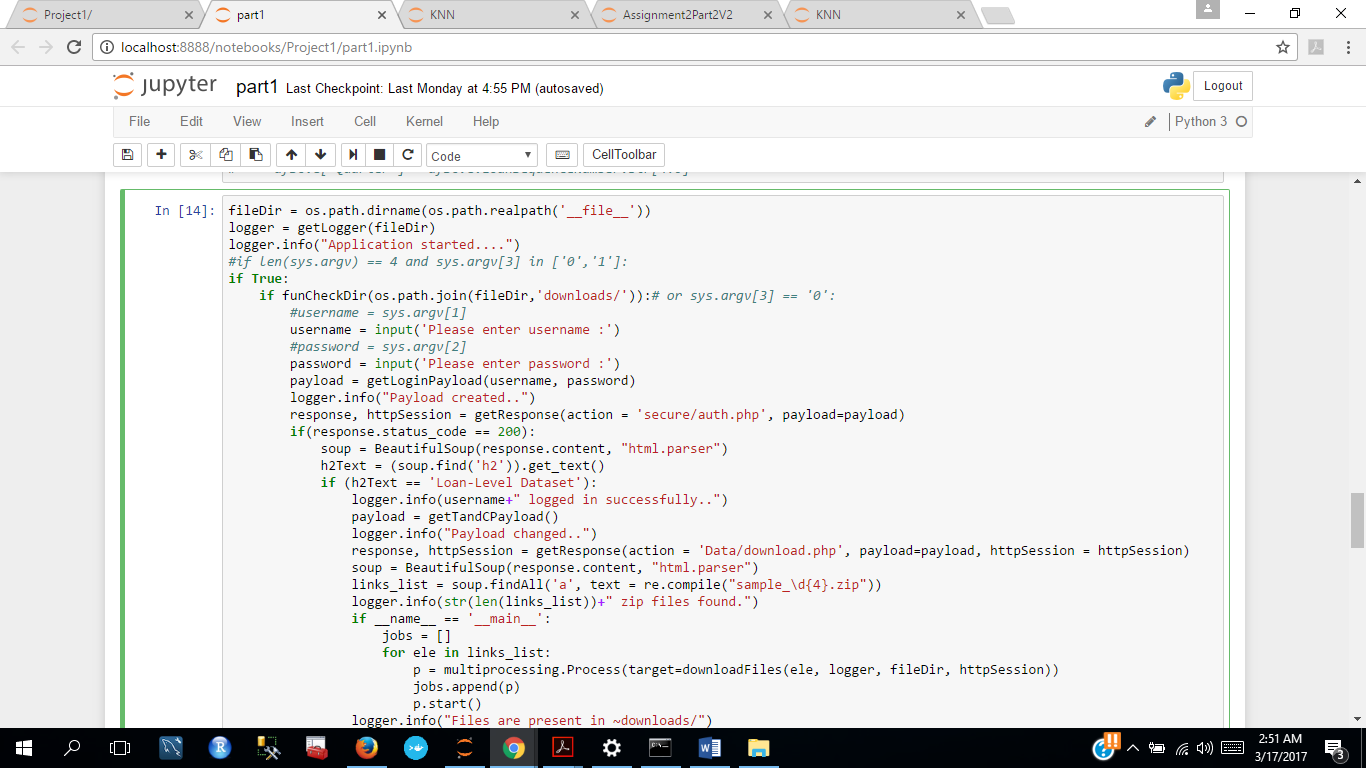
* To run docker image :
  + - docker run -d sumit91188/midterm
* To check docker container
  + - docker ps -a
* To execute our image
  + - docker exec -it dbd9833cd19a python part1.py <username>; <password>
* To go inside the container
  + - docker exec -it a6b36db9158c /bin/bash
    - cd downloads/

## Data Download and pre-processing

Goal:

* Programmatically download the data by getting past the login page using R/Python. Maintain session by saving and passing cookies.
* Scrape the data page and all ‘sample’ files 1999 onwards.
* Preprocess the data and summarize the key information to consolidate summaries into one file.

### Programmatically Downloading data

1. Enter username and password to get past the login page
2. Set session to retain the username and password and pass it in the response header to reach the next page
3. Check if we have reached the ‘Loan-Level Dataset’ page. If true, pass the username and password payload to in the next request to maintain session.
4. Finally, we have reached the data page. Now scrape the data page to find all ‘sample’ files 1999 onwards using the regular expression “sample\_\d{4}.zip"
5. Check if files are present in cache, If not then download the files and programmatically save it in downloads folder in the current directory using (os.path.realpath('\_\_file\_\_'))

### Data Cleaning and Pre-Processing

1. Replacing spaces in Credit Score by an invalid value : 300
2. Replacing spaces in Original Debt to Income Ratio (DTI) by an invalid value : 66
3. Replacing spaces in Original Loan To Value (LTV) ratio by an invalid value : 106
4. Replacing spaces in Original Loan To Value (LTV) ratio by an invalid value : 106
5. Replacing spaces in Super Conforming Flag by ‘N’
6. Filling all the missing values by Unknown

df['CreditScore'].replace({'^\s{1,}':'300'}, regex=True, inplace=True)

df['DTI'].replace({' ':'66'}, inplace=True)

df['OLTV'].replace({' ':'106'}, inplace=True)

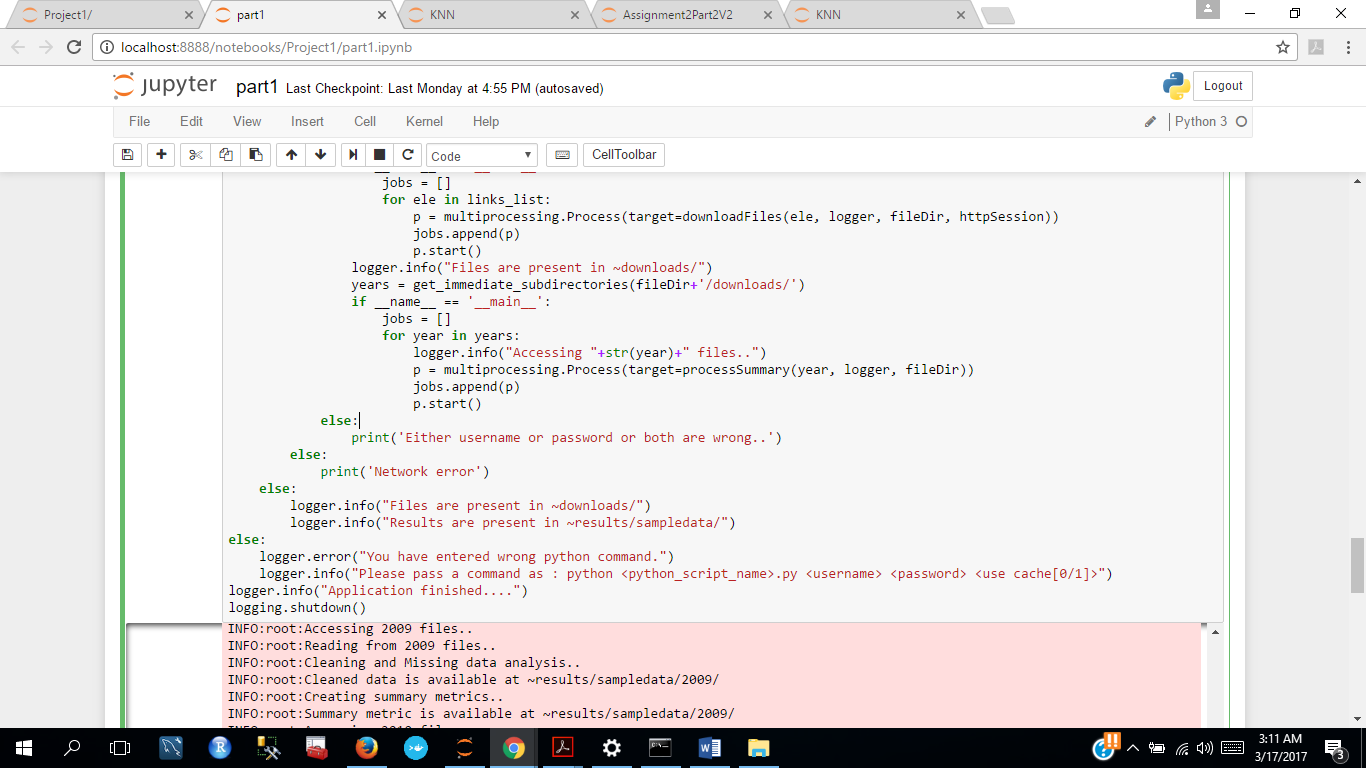
df['OCLTV'].replace({' ':'201'}, inplace=True)

df['SuperConformingFlag'].replace({' ':'N'}, inplace=True)

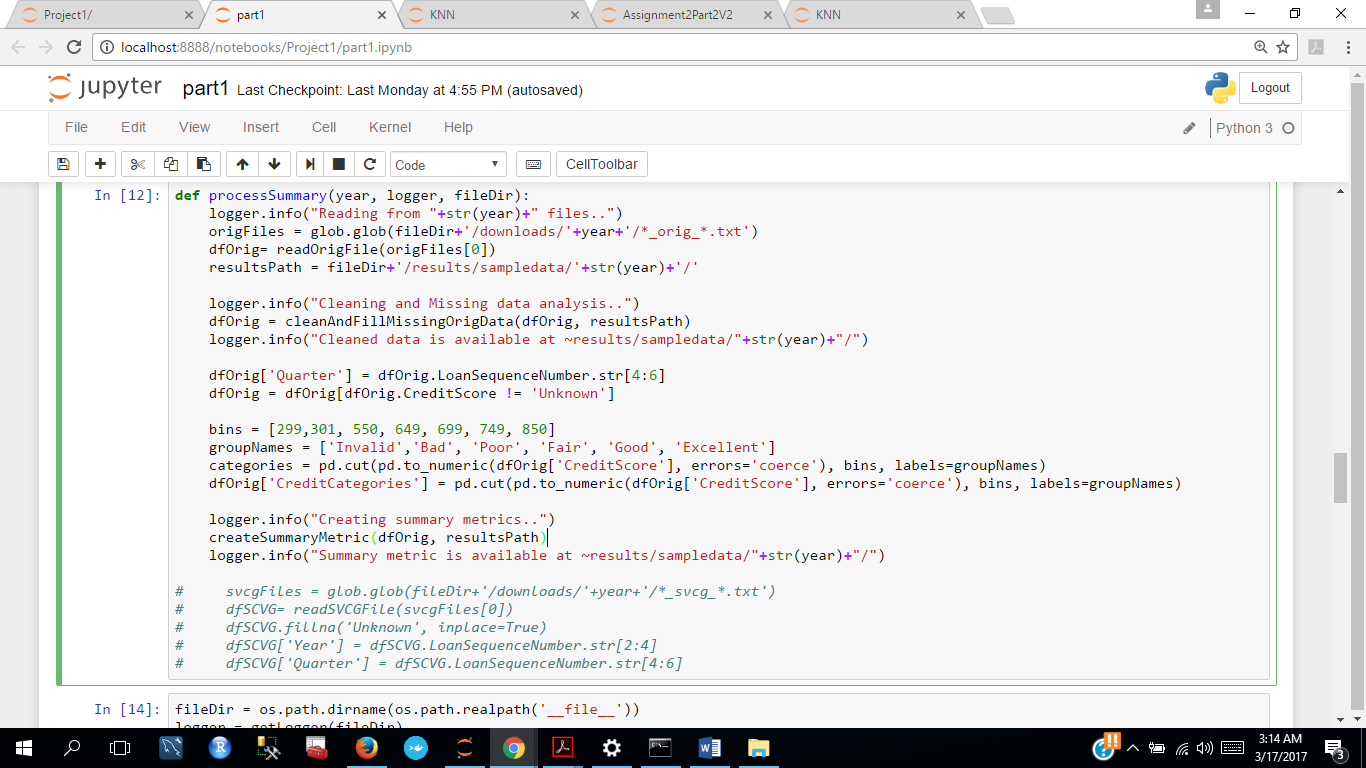
df.fillna('Unknown', inplace=True)

### Parallel Processing for Summary Metrics

1. Parallelly processing the data preprocessing and summary metrics tasks for each year from 1999.

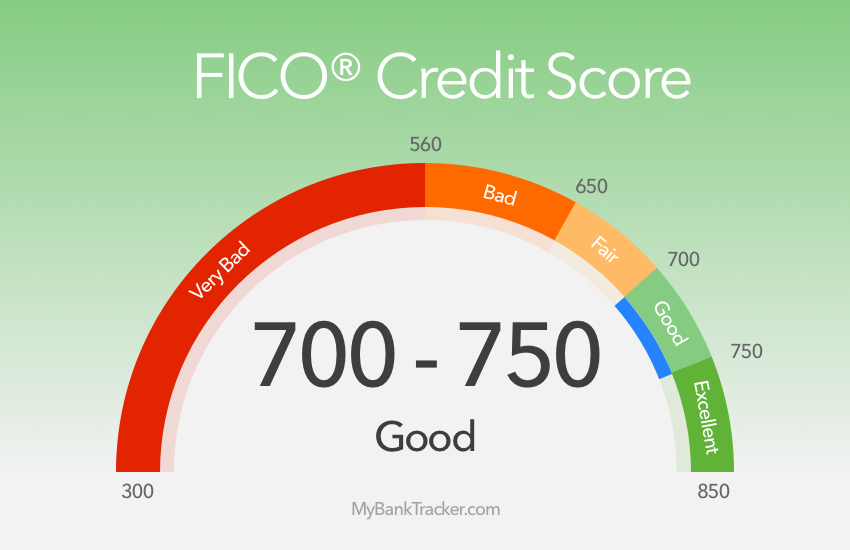


1. Call Function Process Summary that reads the files and calls the function cleanAndFillMissingData



1. Binning : Based on FICO scored for Credit Score we created bins for ranges of Credit Score.

|  |  |
| --- | --- |
| Label | Range |
| 299-301 | Invalid |
| 302-550 | Bad |
| 551-649 | Poor |
| 650-699 | Fair |
| 700-749 | Good |
| 750-850 | Excellent |

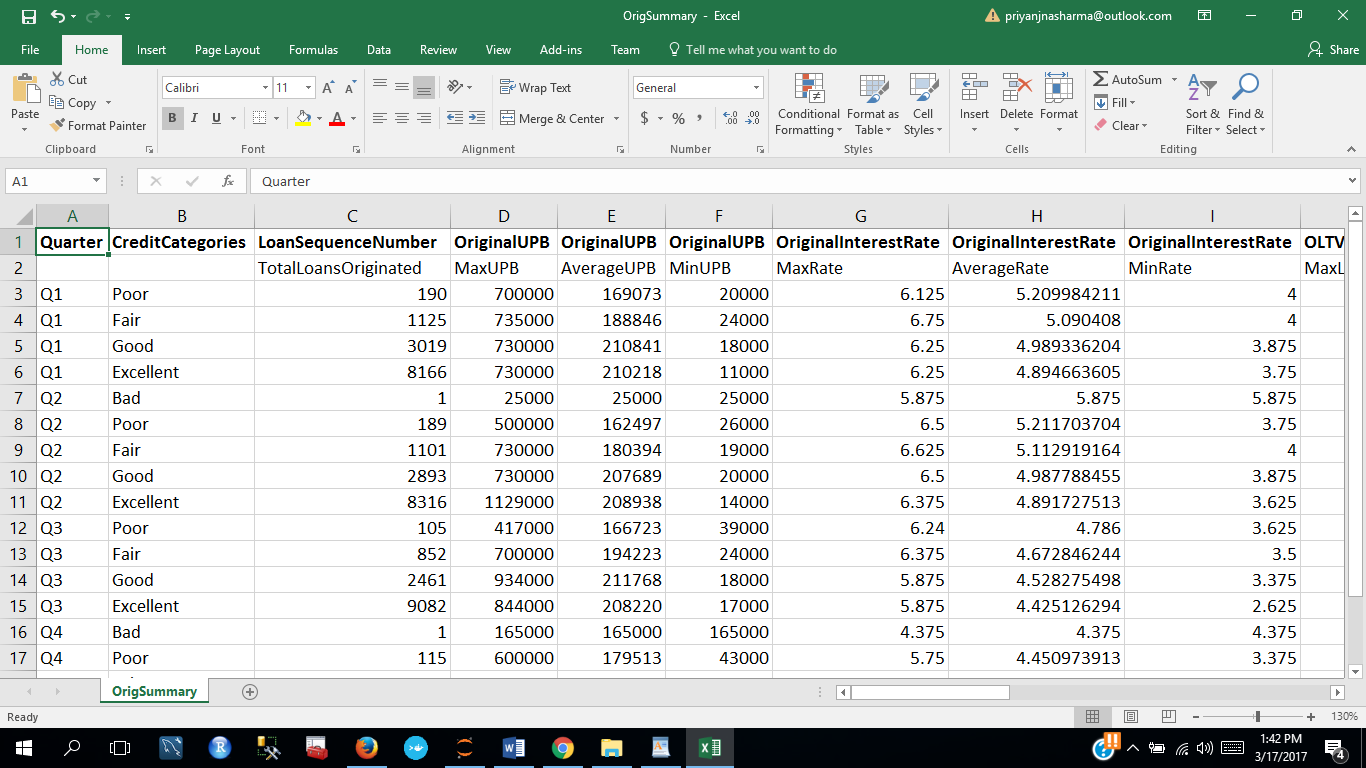


### Summary Metrics

1. Calculating summary metrics for the Origination file grouping it by columns Quarter and “CreditCategories”: Categories created from binning Credit Scores and saved it to a csv file.
2. Origination data:

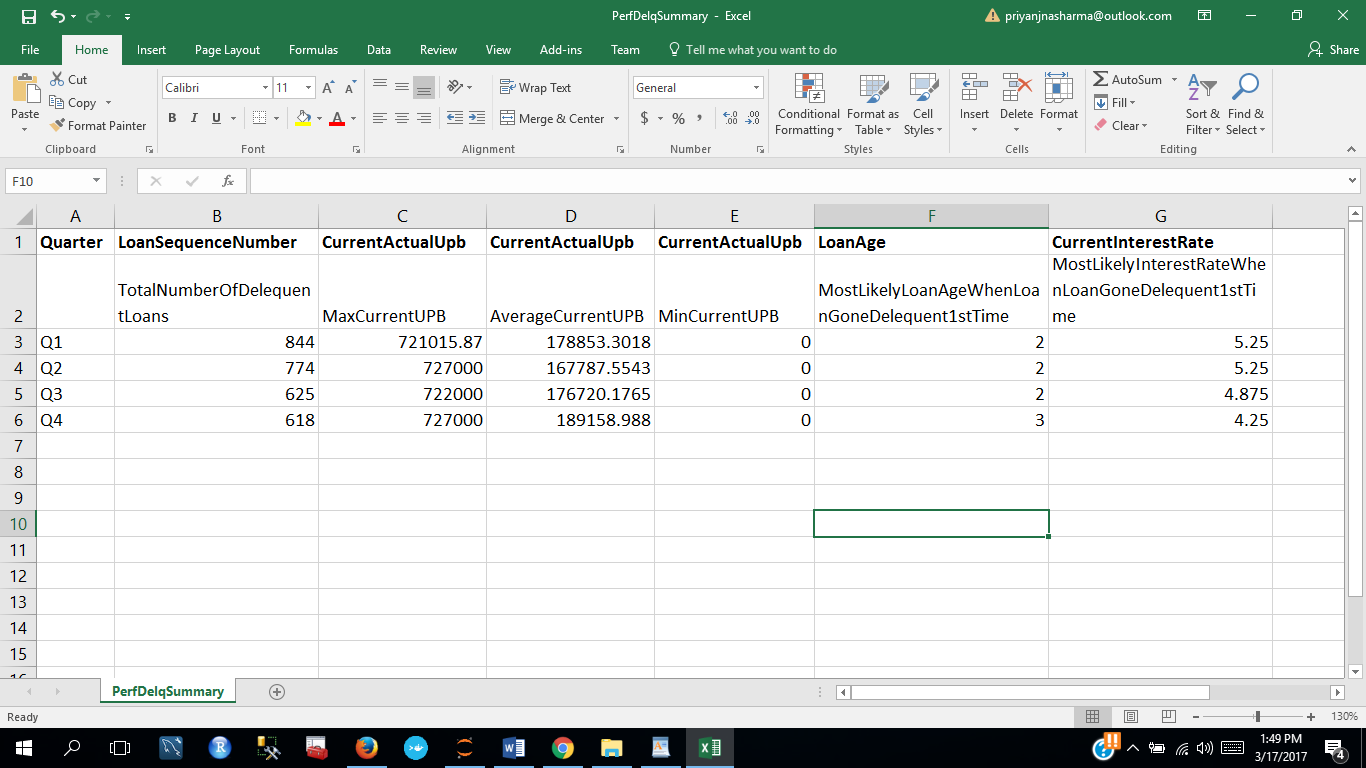
Defined the following summaries:

* Total Loans Originated
* Original Interest Rate: Maximum, Minimum and Average Interest rates
* Original -Loan to Value: Maximum and Minimum
* Original Loan Term: Maximum, Minimum and Average term of loan
* Occupancy Status: the most likely occupancy status to occur: mode
* First Time Home Buyer: total number of first time home buyers
* Prepayment Penalty Incurred: total prepayment penalty cases occurred
* Channel: the most likely channel to occur: mode
* Sample for year 2010



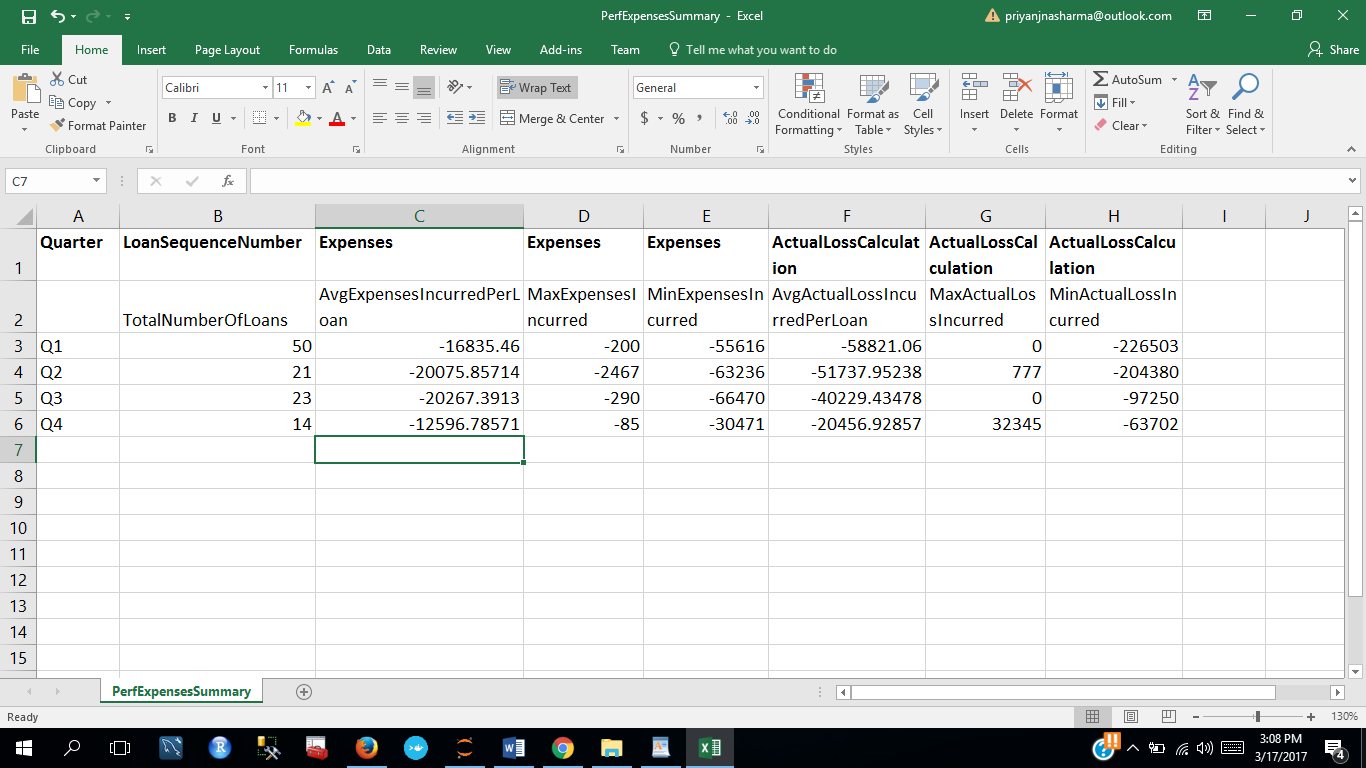
Performance data

* Summary 1: focuses on the delinquency trend and summarizes the total number of delinquent loans, Current Actual Unpaid balance, Loan Age and Interest Rate when the loans got delinquent for each quarter

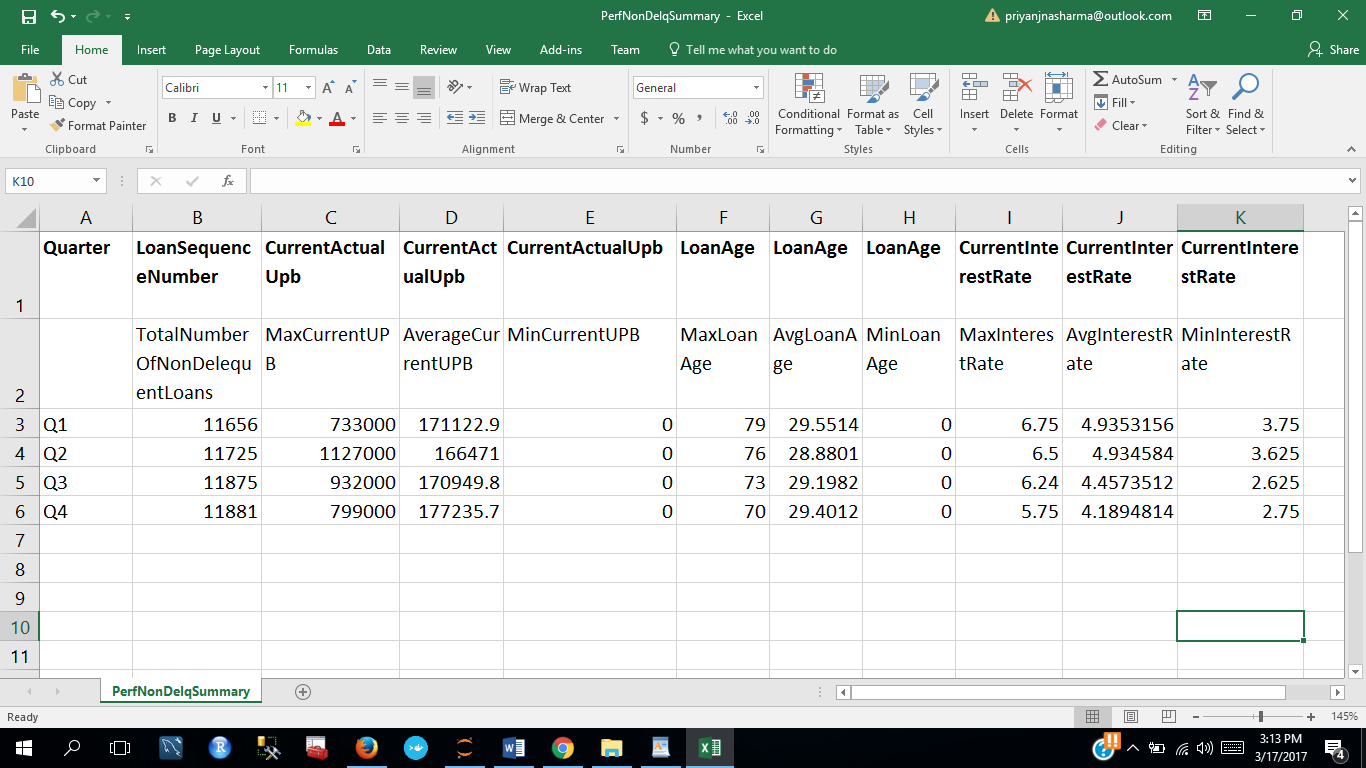


* Summary 2: Focuses on the expenses and loss calculations per quarter

Sample for year 2010



* Summary 3: Focuses on the non-delinquent loans and calculates summaries of interest rates, Unpaid balance and loan age for each quarter



## Exploratory Data Analysis

Goal:

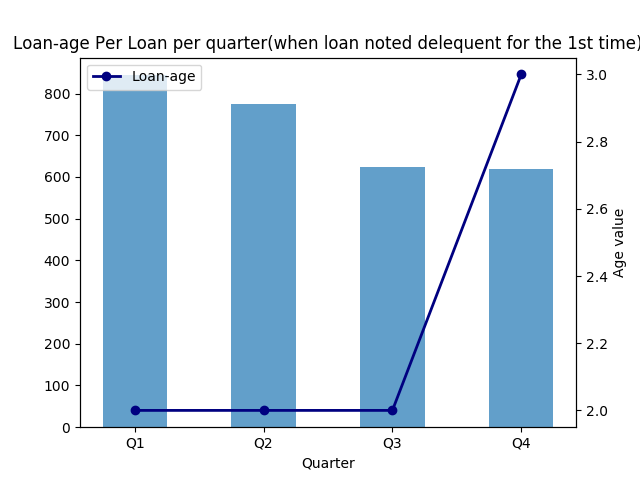
* Graphically represent summaries of data through python plots
* Create tableau dashboards to analyze Quarterly data 2005 onwards

### Summary Plots

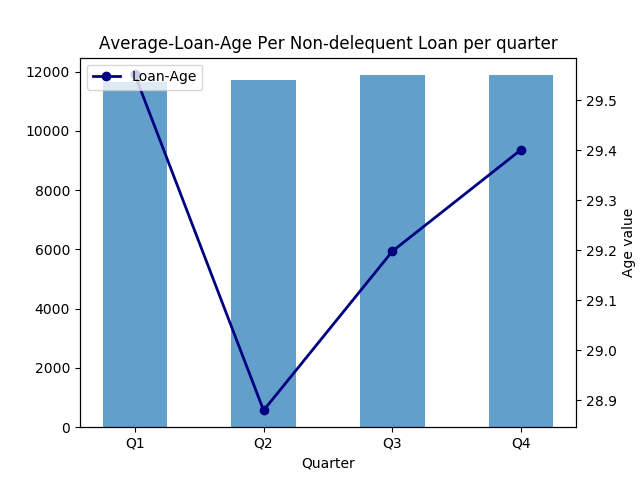
We used Python to plot summaries of data from 1999 onwards for Origination and Performance data

In this report, we are going to present the analysis and plots for the year 2010. To obtain the summary for other years, we can run the program for that year which will give us summary metrics in form of a csv file and plots for analysis.

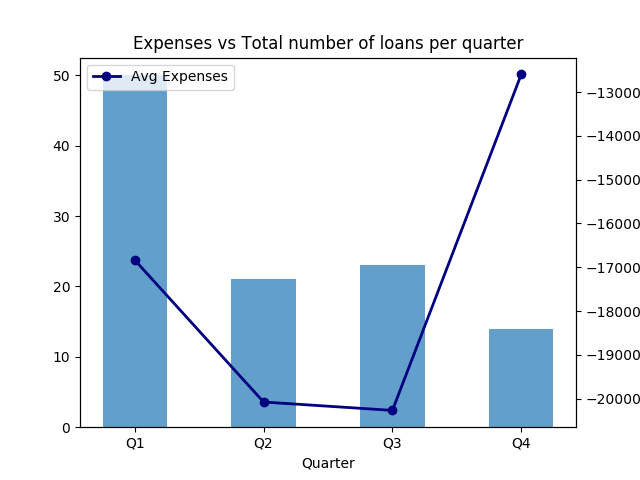
1. Loan Age versus Number of Delinquent loans per Quarter
   * This plot shows the total number of delinquent loans and the average loan – age per quarter
   * Here the histograms show the total number of loans for each quarter. The blue line depicts the average loan-age value with scale on the right side.
   * Observation: For year 2010, the average loan age when the loan got delinquent for the first time was highest for the fourth quarter and for all the other quarters it remained the same.



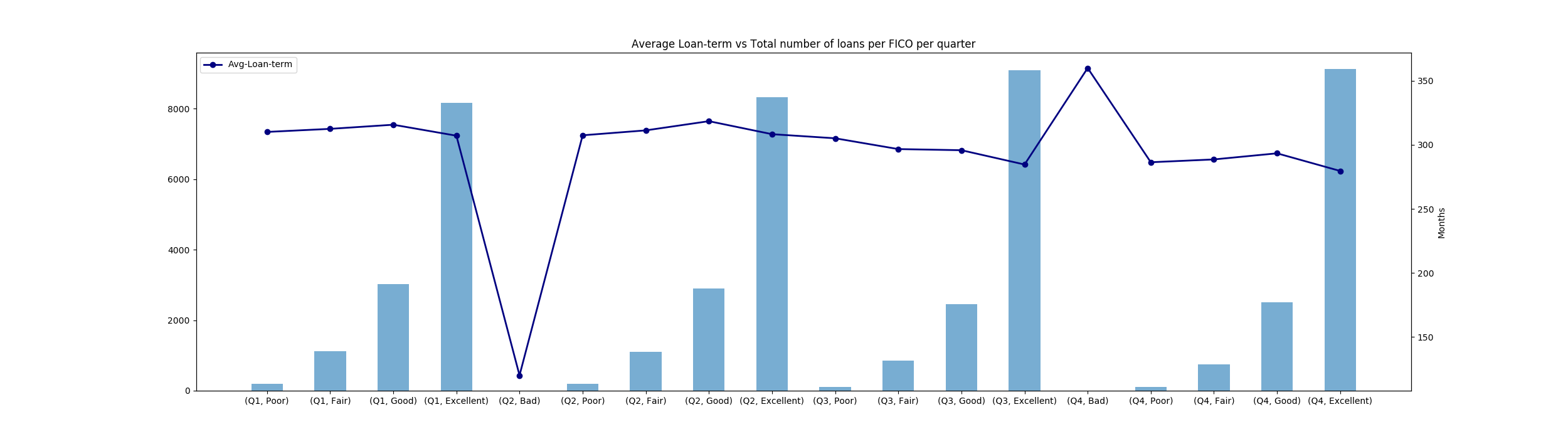
1. Average Loan Age versus Number of non-Delinquent loans per Quarter
   * This plot shows the total number of non-delinquent loans and the average loan – age per quarter
   * Here the histograms show the total number of non-delinquent loans for each quarter. The blue line depicts the average loan-age value with scale on the right side.
   * Observation: For year 2010, the average loan age for all non-delinquent loans was highest in Q1 and dropped greatly for Q2



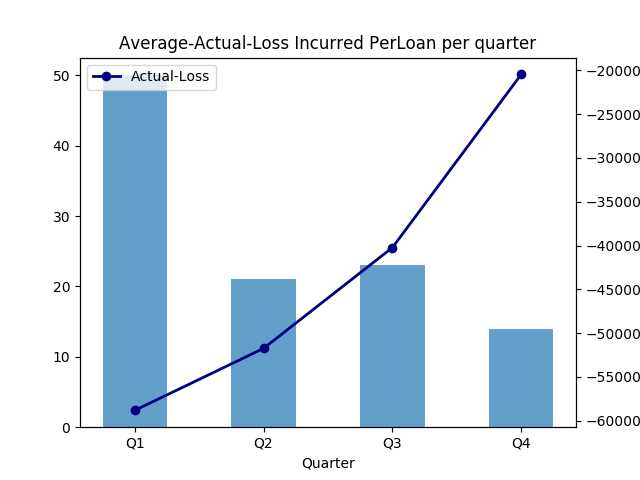
1. Expenses versus Number of loans per Quarter
   * This plot shows the total number of loans and the average loan – age per quarter
   * Here the histograms show the total number of loans for each quarter. The blue line depicts the average value of expenses with scale on the right side.



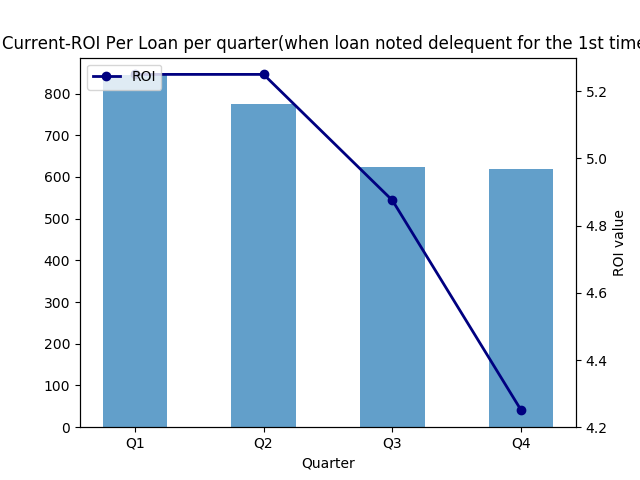
1. Average Loan term versus Number of loans per FICO Credit Score Range per Quarter
   * This plot shows the average loan term for total number of loans per FICO Credit Score range that we used to bin the data
   * Here the histograms show the total number of loans for each range of Credit score – Good, Bad, Fair etc. for each quarter. The blue line depicts the average value of loan term
   * Observation: For the year 2010, people with high credit scores got a high loan term as compared to those with lest credit score ranges



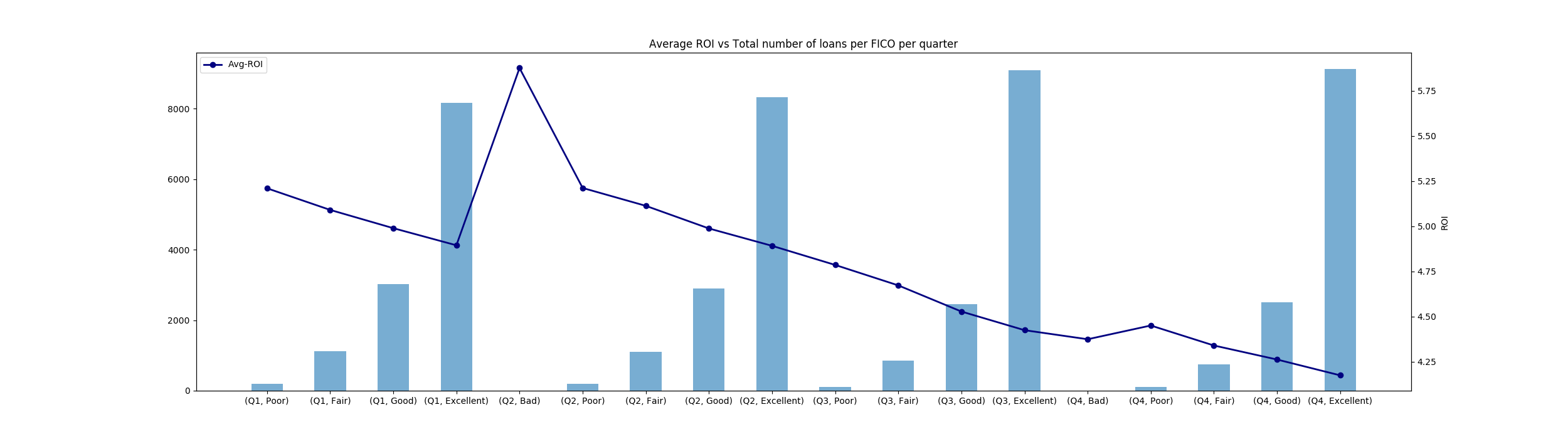
1. Average Actual Loss incurred versus Number of loans per Quarter
   * This plot shows the average actual loss incurred for total number of loans per quarter
   * Here the histograms show the total number of loans each quarter. The blue line depicts the average value of loss incurred
   * Observation: For the year 2010, average Loss was highest in Q4



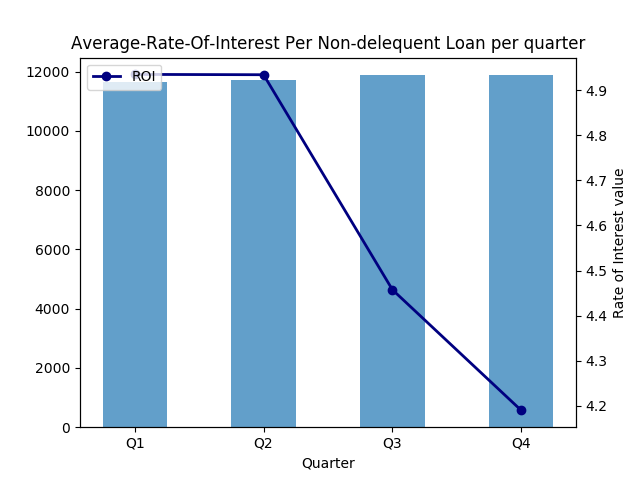
1. Current Rate of Interest incurred versus Number of loans per Quarter
   * This plot shows the average



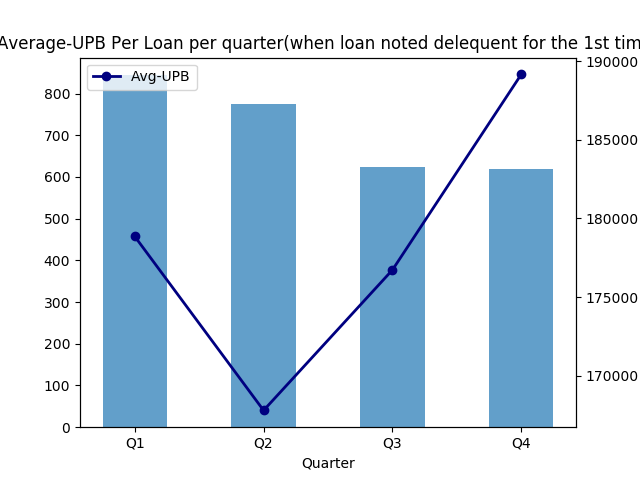
1. Average Rate of Interest versus Number of loans per FICO Credit Score Range per Quarter
   * This plot shows the average loan term for total number of loans per FICO Credit Score range that we used to bin the data
   * Here the histograms show the total number of loans for each range of Credit score – Good, Bad, Fair etc. for each quarter. The blue line depicts the average value of loan term
   * Observation: For the year 2010, For Q2 rate of interest was the highest for credit scores in the bad range. Interest rate gradually decreased for other quarters thereafter.



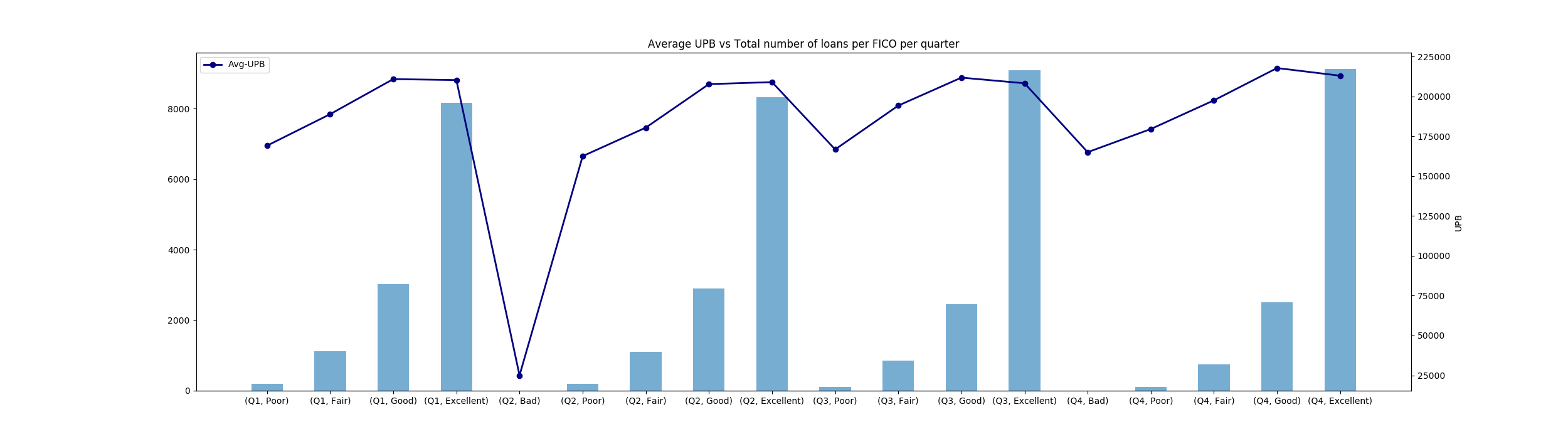
1. Average Rate of Interest versus Number Versus Non-delinquent loans per Quarter
   * This plot shows the average rate of interest for total number of non-delinquent loans per quarter
   * Here the histograms show the total number of non-delinquent loans for The blue line depicts the average rate of interest for each quarter.



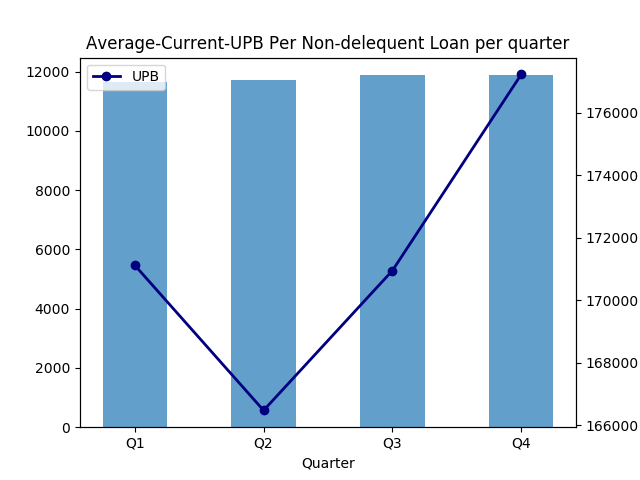
1. Average Unpaid Balance versus Number of delinquent loans per Quarter
   * This plot shows the average unpaid balance for total number of delinquent loans per quarter
   * Here the histograms show the total number of delinquent loans for each quarter and the blue line depicts the average Unpaid Balance
   * Observation: For the year 2010, For Q2 the average unpaid balance is lowest and highest for Q4



1. Average Unpaid Balance versus Number of loans per FICO credit score range for each quarter
   * This plot shows the average unpaid balance for total number of loans for each credit score range per quarter
   * Here the histograms show the total number of delinquent loans for each credit score range quarter and the blue line depicts the average Unpaid Balance
   * Observation: For the year 2010, For Q1 the average unpaid balance is lowest and highest for Poor range of credit scores and highest for credit scores in the good range. Similarly, we can check for other quarters as well.



1. Average Unpaid Balance versus Number of non-delinquent loans per Quarter
   * This plot shows the average unpaid balance for total number of non-delinquent loans per quarter
   * Here the histograms show the total number of non-delinquent loans for each quarter and the blue line depicts the average Unpaid Balance
   * Observation: For the year 2010, For Q2 the average unpaid balance is lowest and highest for Q4



### Tableau Dashboards

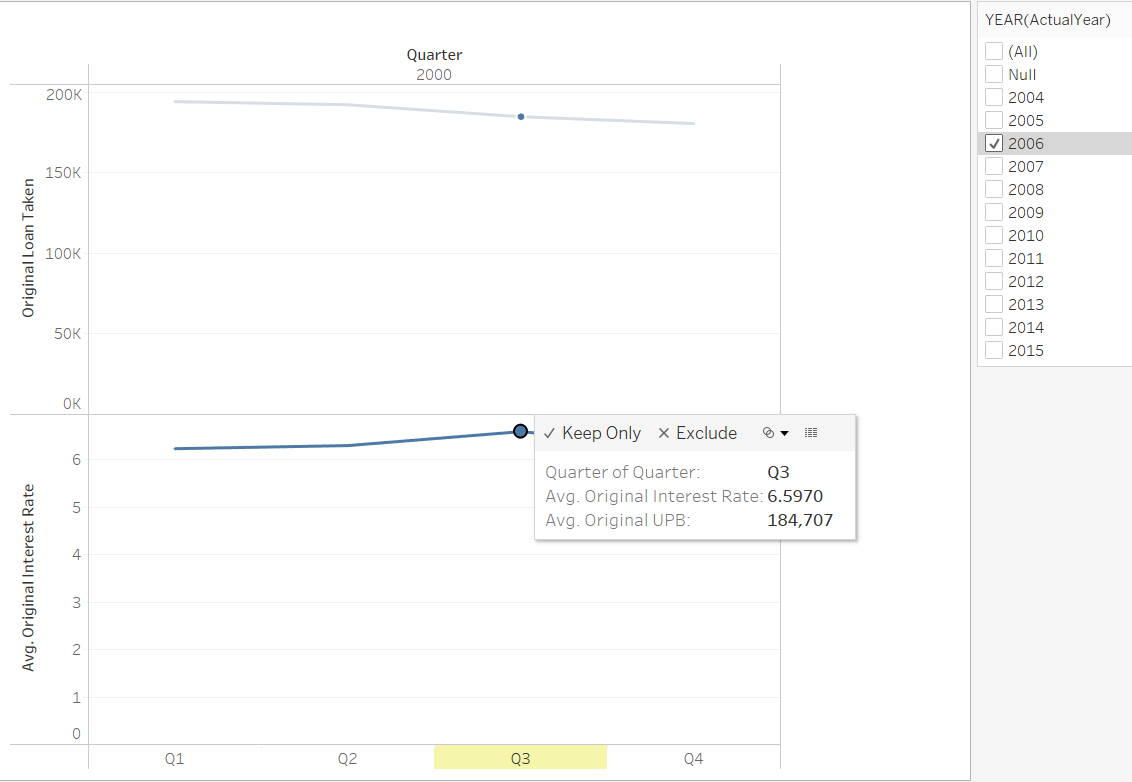
Goal:

* Create tableau dashboard to analyze quarterly data from 2005
* Analyze trends in Interest rates, delinquency trends over quarters, location specific insights.
* Tableau link: [https://public.tableau.com/profile/gautam6726#!/](https://public.tableau.com/profile/gautam6726#%21/)

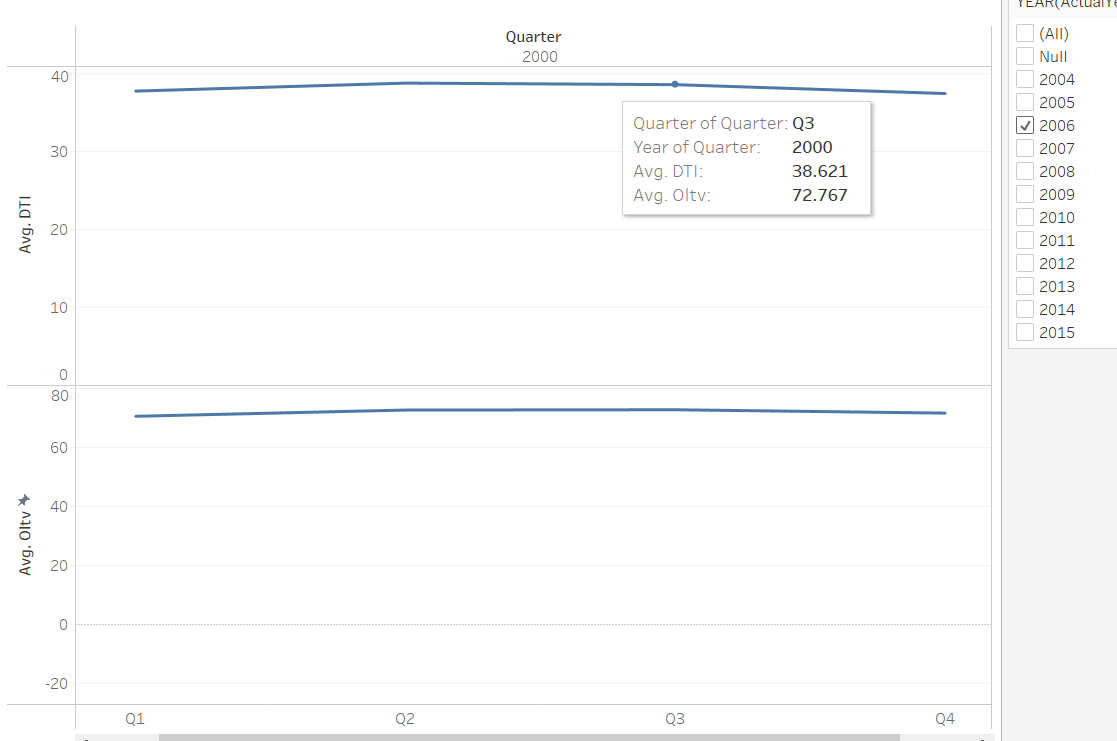
#### Quarterly Analysis:

We have analyzed the case:

1. Loan Amount taken per quarter on average interest throughout the years (2004-16). As you can see below, amount of 184,707 was taken at an average interest rate of 6.5970 for the year 2006, Quarter3.
2. For the same year and same quarter, the average Original Loan to Value ratio was 72.767 and average Debt to Income ratio was 38.621.



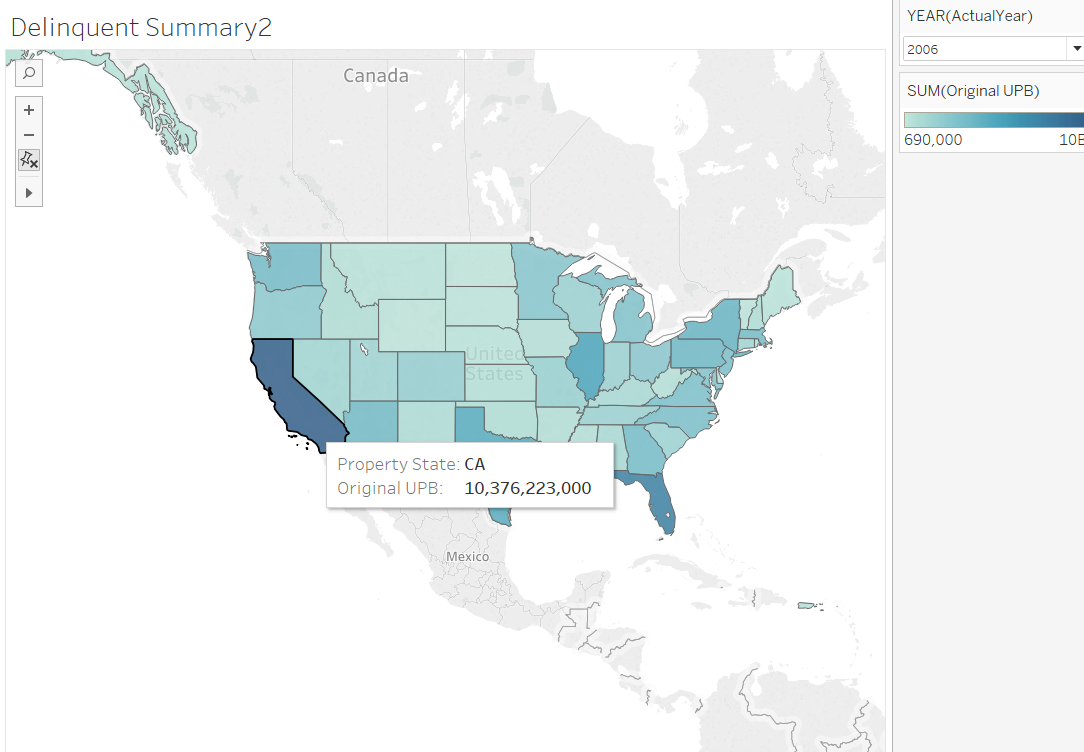
We have created a factor Delinquency Rate using formula:

 **Delinquency Rate = Distinct Count([Current Loan Delinquency Status]==1)/COUNTD([No of Loans)])\*100**

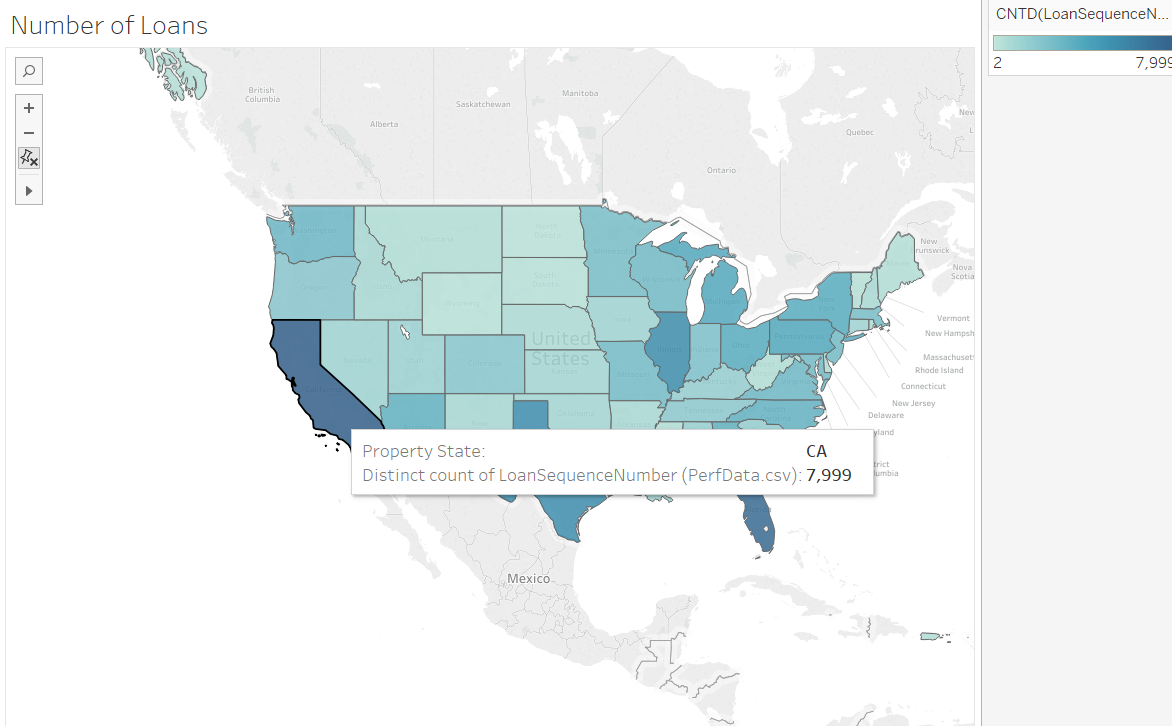
#### Geographical Analysis(State-wise):

We have performed state wise analysis to observe some patterns and trends based on the home loan key indicators of each state.

1. State wise total Loan amount taken.: As we can see below, for year 2006 California state leads other states with a total loan amount of $10,376,223,000.



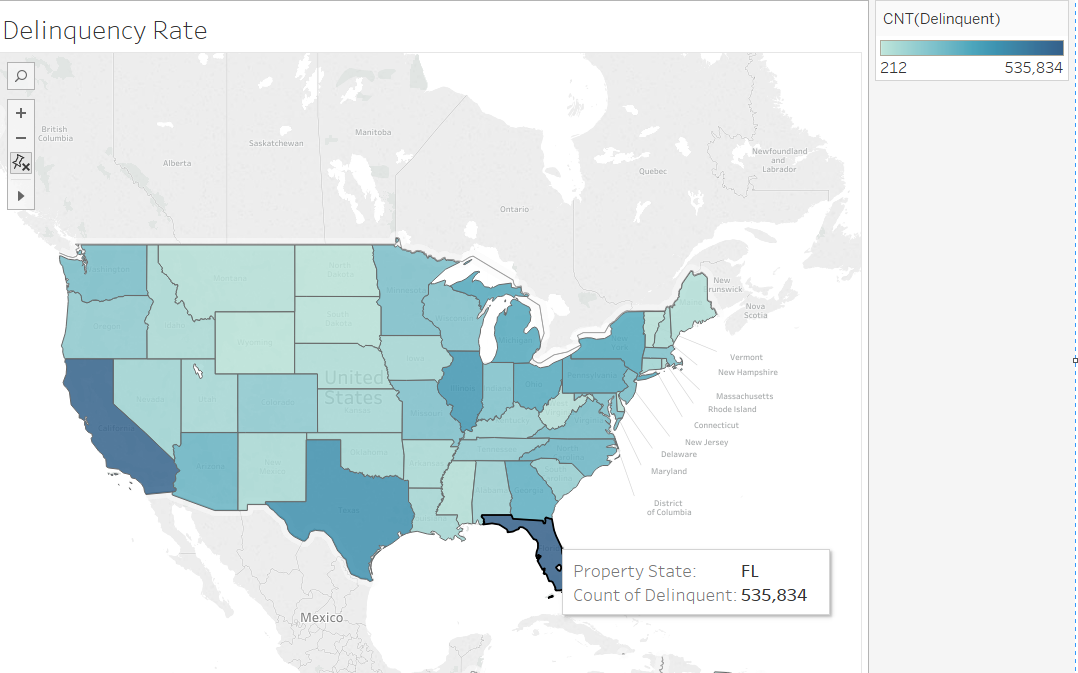
1. Count of Loan: California leads with 8,000 home loans for the year 2016 with florida on 2 with 570 loans.



1. Interest and Delinquency Over States: Delaware and some other states were leading with an average interest rate of 6.350 with a minimum of 6.00 in some states.

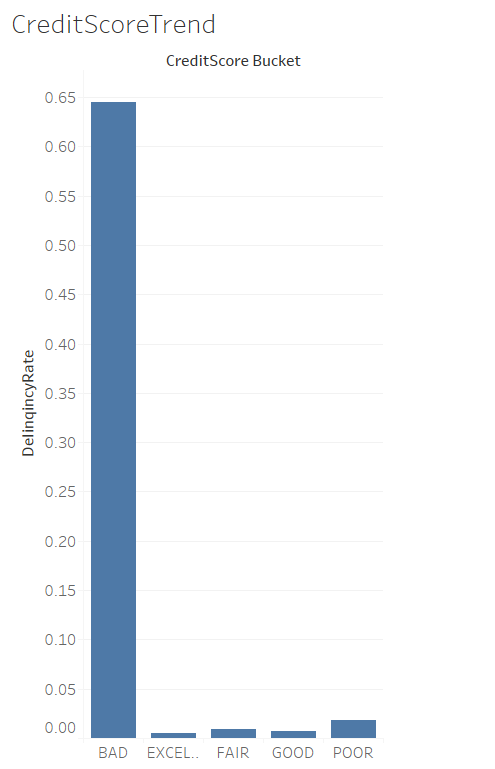


1. **Delinquent States: Florida** was ranked one with over 500k delinquent loan cases with Virginia as the lowest number of delinquent cases.

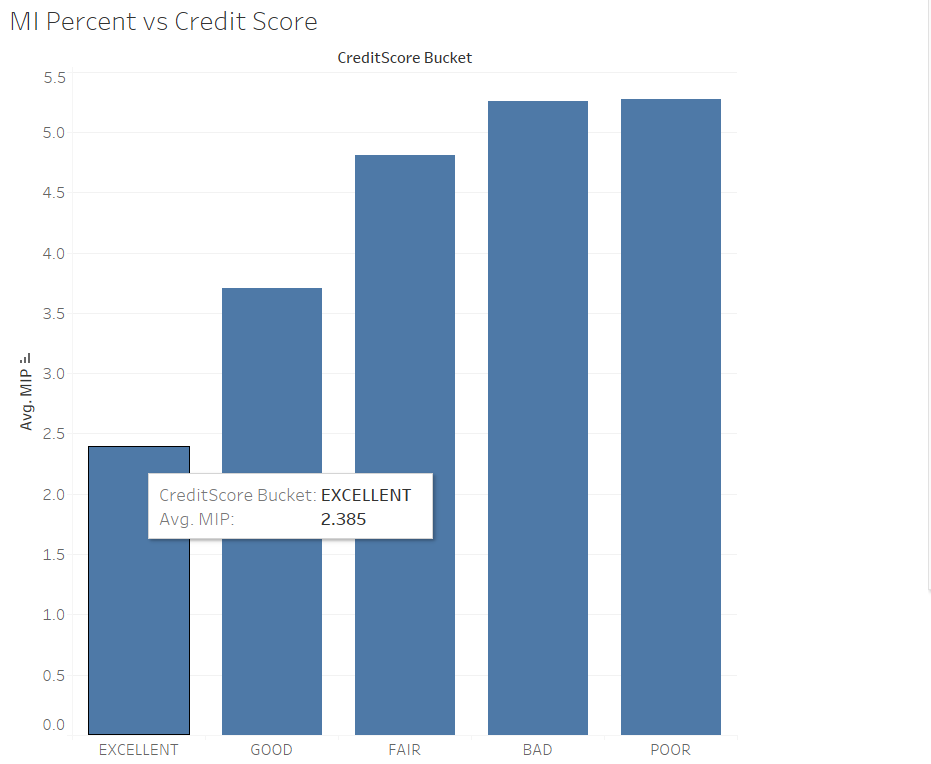


#### Credit Based Analysis: Created bins for credit scores marked as Good, Bad etc. based on ranges and performed analysis on them.

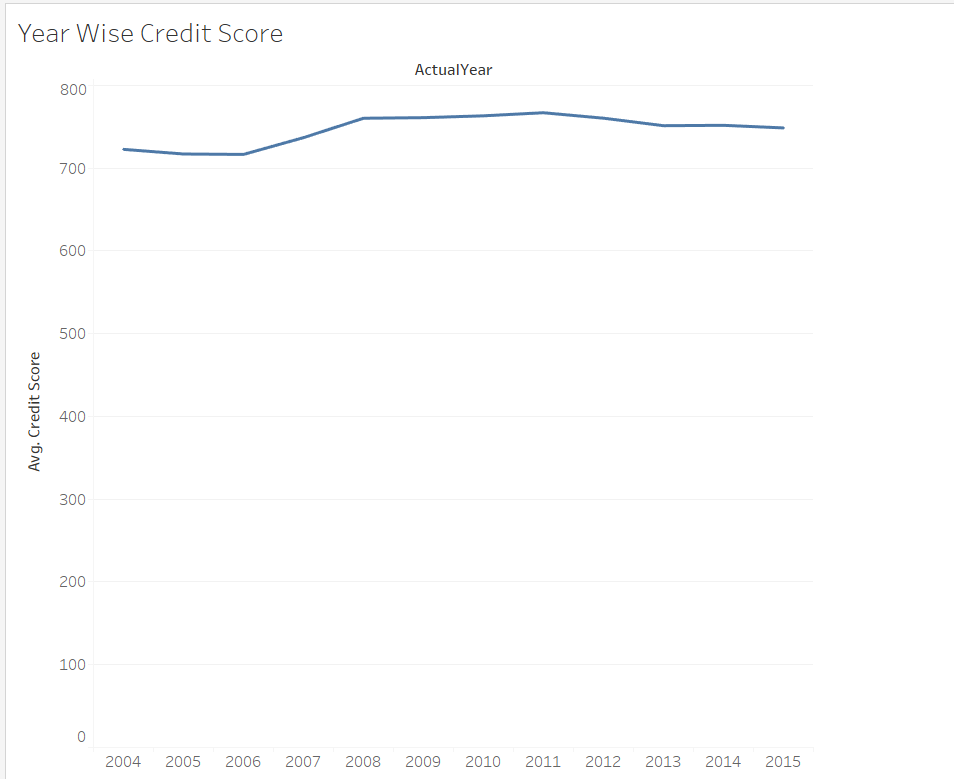
1. As shown, credit range bucket of Excellent(above 750) have been least delinquent with Bad credit ranges have been tending to most delinquent.



1. **Loan applications with higher credit score range tend to pay less mortgage insurance as seen.**



1. Credit score analysis over year shows a decline in the credit scores from 2011 to 2015.



1. Relation between Credit scores and LTV and DTI. Higher credit score will be availing higher Loan to Value and Debt to income.



#### Channel Based Analysis:

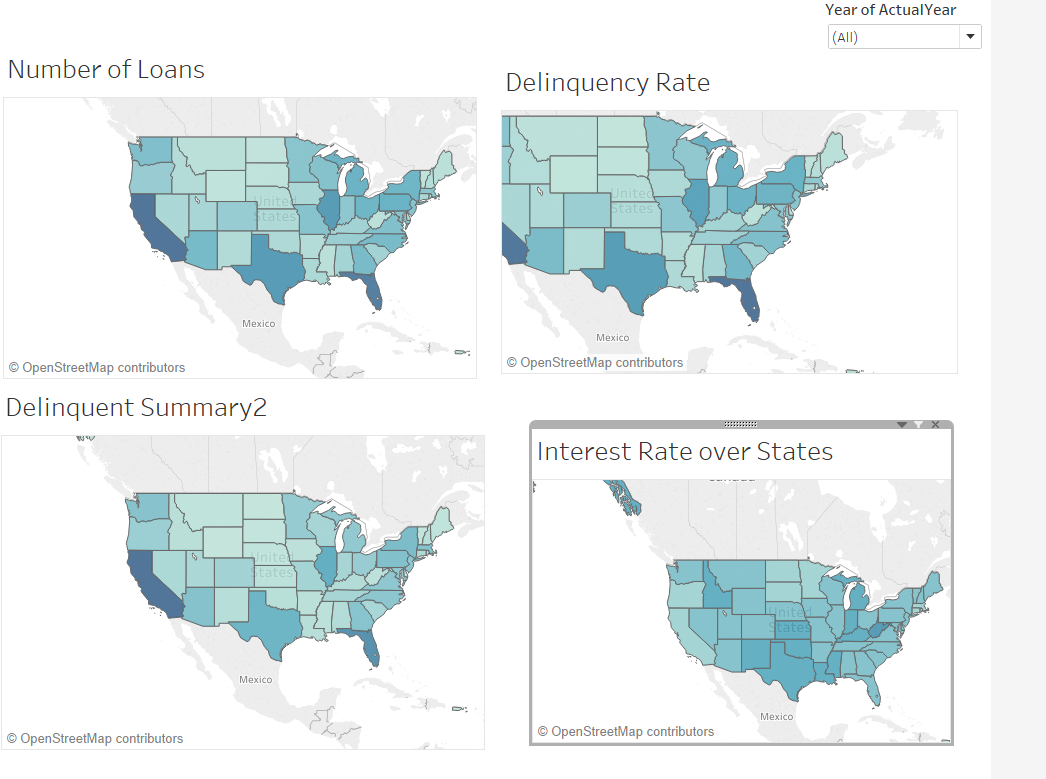
This is based on number of loans provided through each channel over the year and how it is connected to that loan being delinquent and the interest trend by Channel medium i.e. Broker, Retail etc.

Below, we can see most number of loans were given through Retail but least interest were paid through correspondents and TPO were most delinquents.

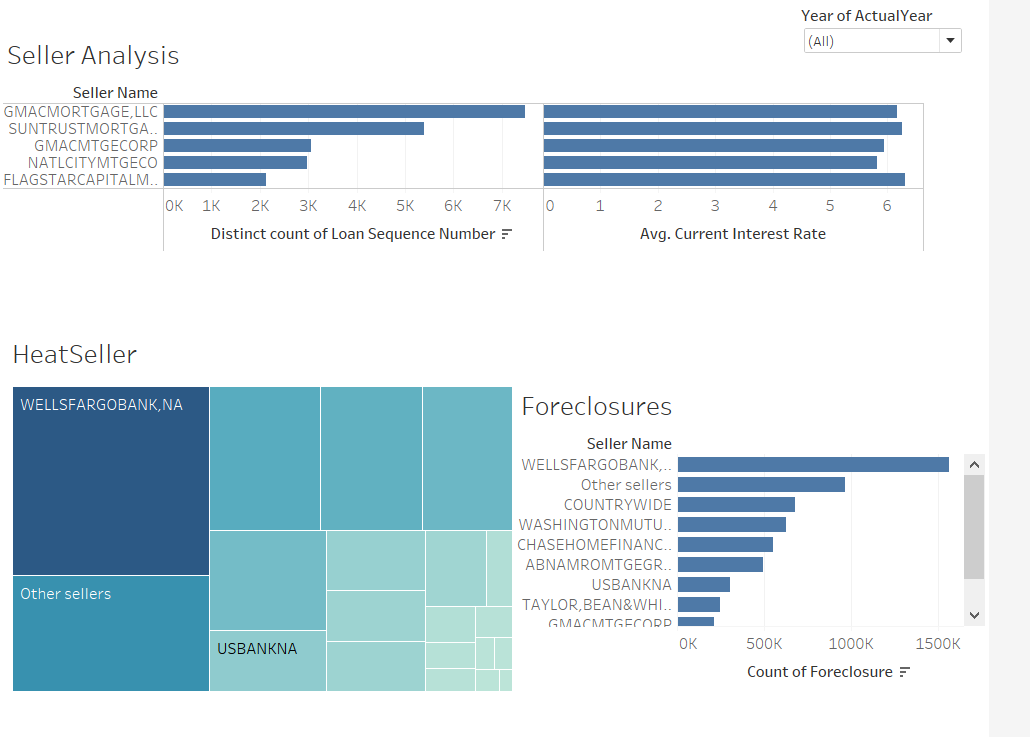


#### Interactive dashboards:

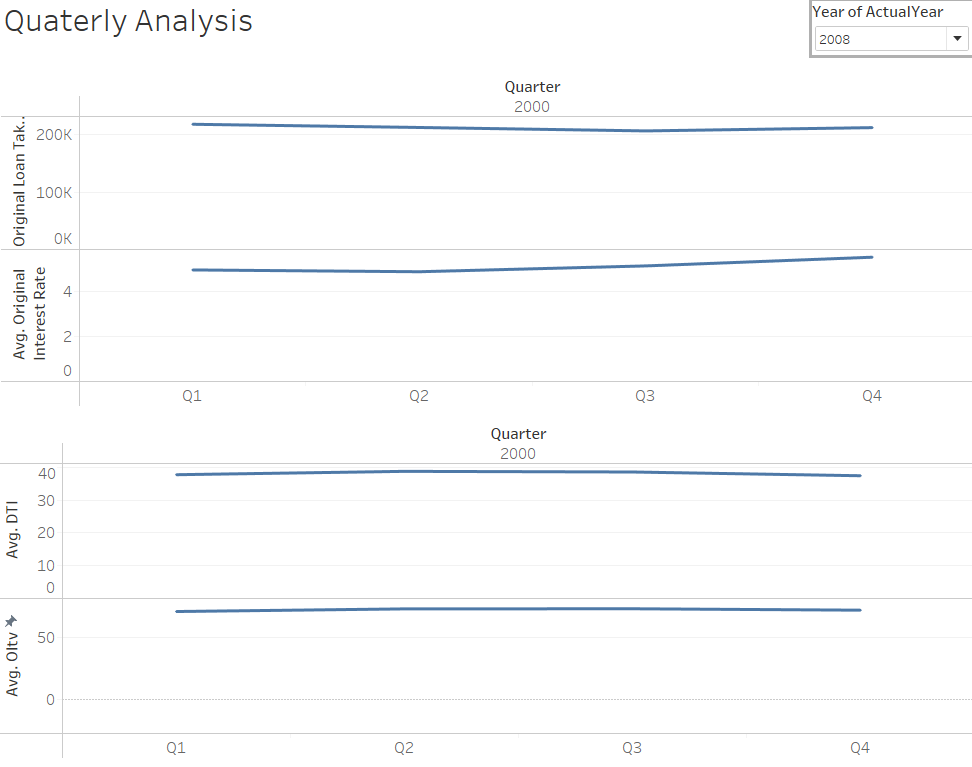
1. **Year wise Geographical Analysis:**



1. **Seller Analysis (yearly and quarterly):**



1. **Quarterly Analysis:**



# Part2: Building and Evaluating Models

## PREDICTION

### REGRESSION

Regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent predictors. More specifically, regression analysis helps one understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed.

Variable Selection

Selection techniques of features that will best fit the model.

### Exhaustive Search Variable Selection

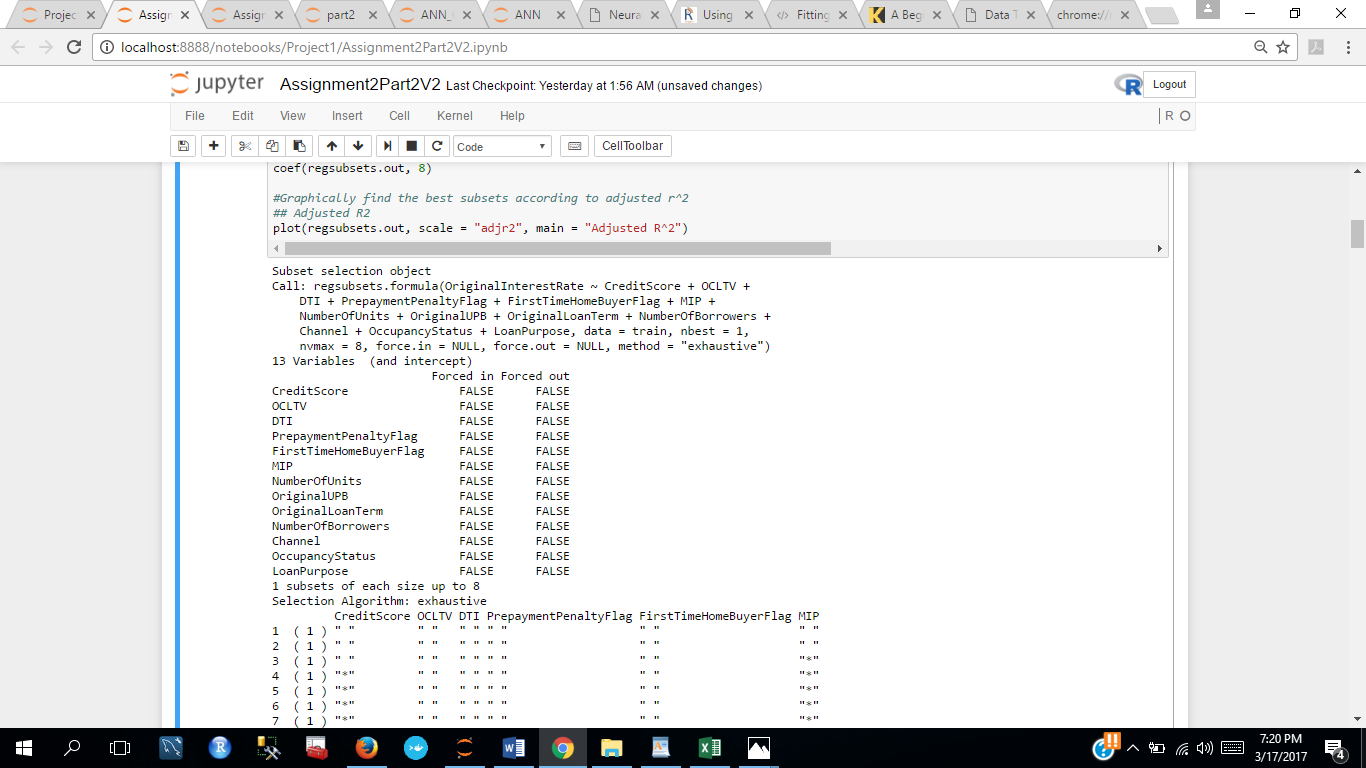
This technique finds subsets of the all the possible predictors for a regression mode. This the same as brute force search. An exhaustive search will build regression models with every possible combination of parameters and recommend the one which has the best adjusted-R^2 and most statistical significance based on t-value.

Process:

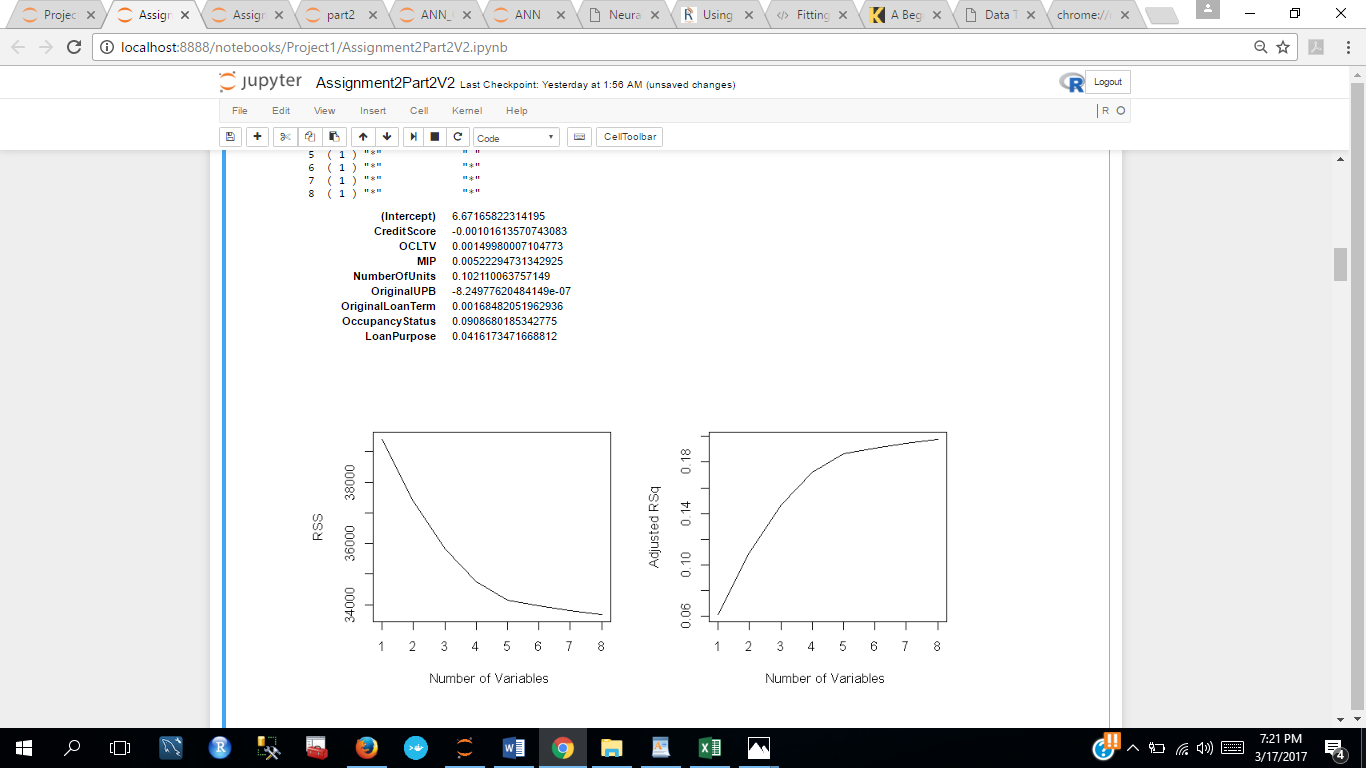
1. We performed Regression on all subsets of variables and chose best model for each number of predictors up to 8.

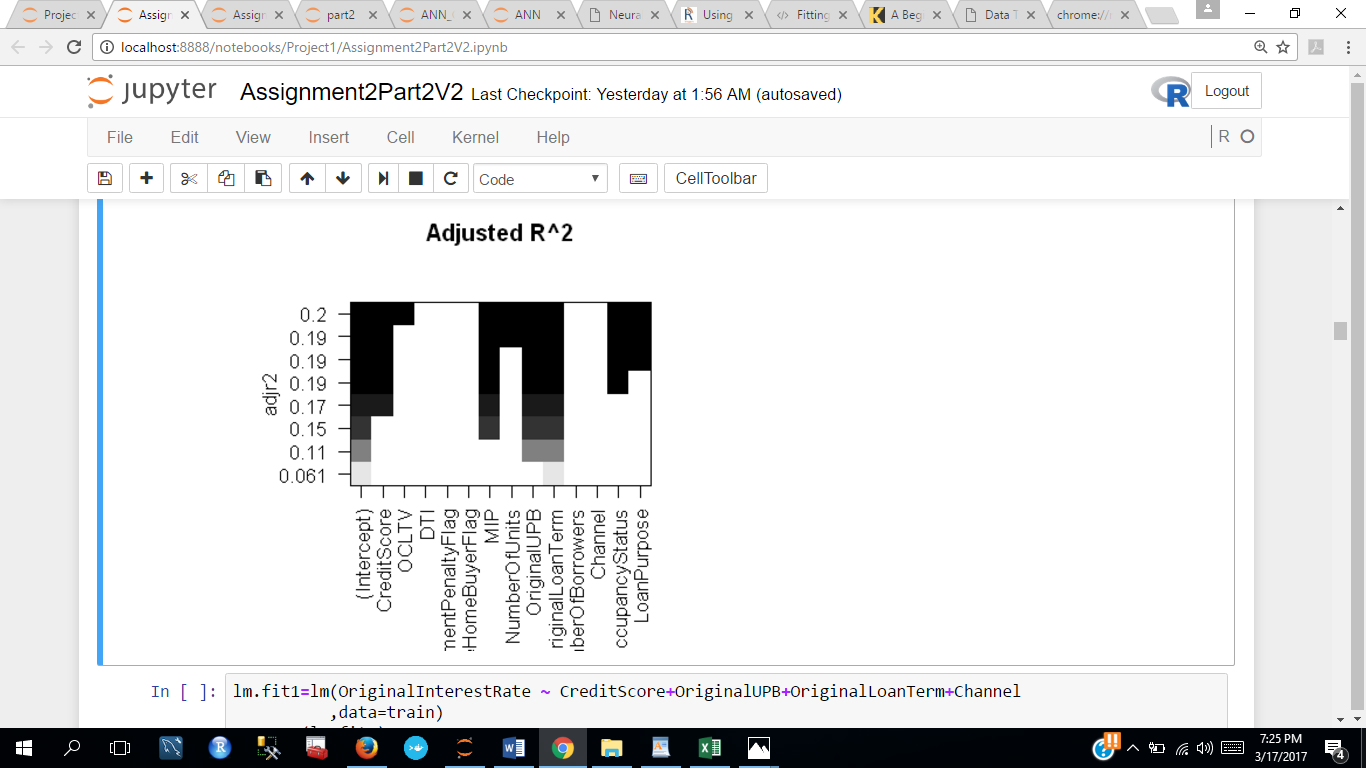
regsubsets.out <- regsubsets(OriginalInterestRate ~ CreditScore+OCLTV+DTI+PrepaymentPenaltyFlag+FirstTimeHomeBuyerFlag+MIP+NumberOfUnits+OriginalUPB+OriginalLoanTerm+NumberOfBorrowers+Channel+OccupancyStatus+LoanPurpose, data = train,  
nbest = 1, # 1 best model for each number of predictors  
nvmax = 8, # NULL for no limit on number of variables force.in = NULL, force.out = NULL, method = "exhaustive") regsubsets.out

1. Output:



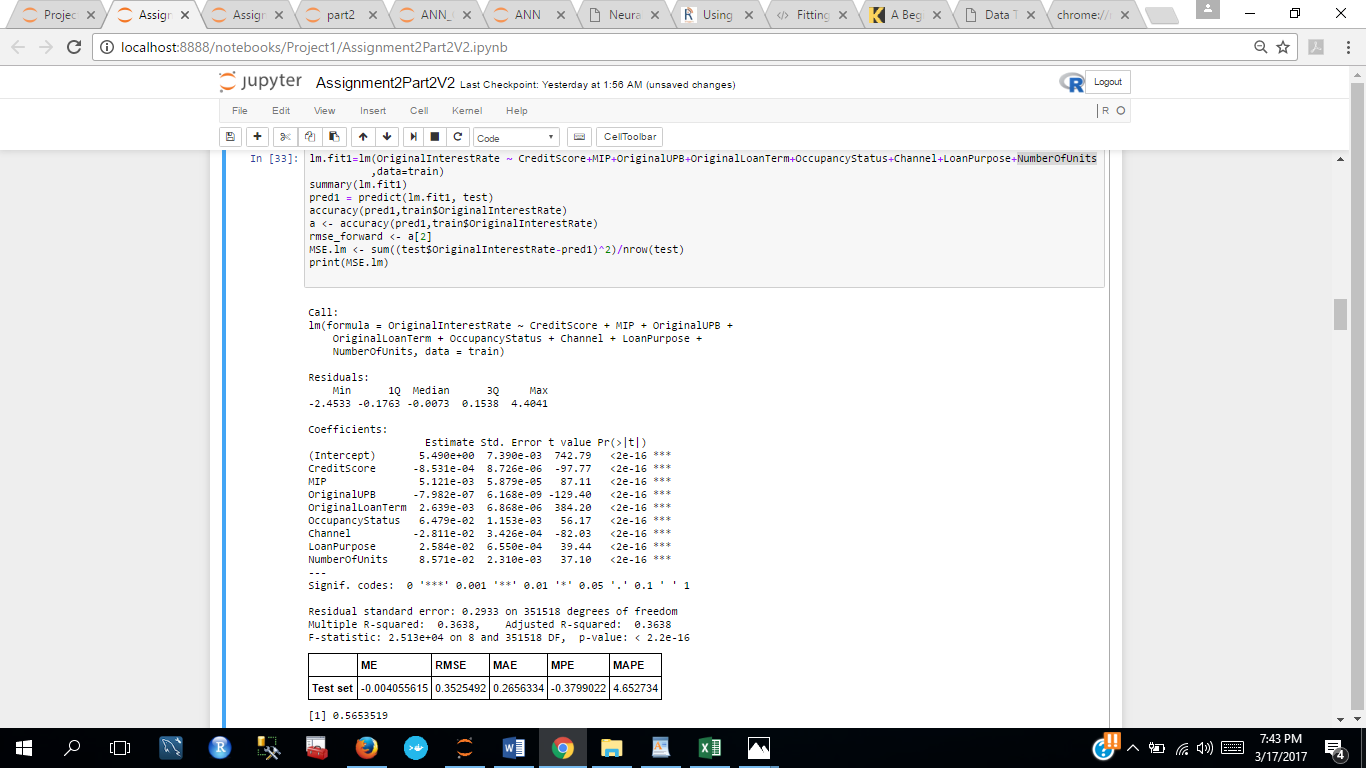
1. Created a plot of variables and their adjusted r^2 , RSS to choose the best variables
2. We see that after 7 variables the adjusted r^2 does not vary much, Hence we select top 7 variables





1. We considered the variables CreditScore, OCLTV, MIP, NumberOfUnits, OriginalUPB, OriginalLoanTerm, OccupancyStatus, LoanPurpose and ran a fit to the regression model with these variables.

Output:



1. This gave us an adjusted r^2 value of 0.3638 and an RMSE of ~35 %.
2. We again did an exhaustive search for nvmax a= 5,6,7. The best result we got was from 5 variables CreditScore, MIP, Original UPB, OriginalLoanTerm and Channel which gave us an adjusted r^2 value of 0.353 and an RMSE of ~34%

### **Forward Selection**

The only difference between this and Stepwise regression is that none of the variables are dropped out. Once a predictor enters a model, it is never deleted, and new variables will continue to be added if the t-value increases.

#forward selection

regfit.fwd=regsubsets(OriginalInterestRate ~.,data=train,nvmax=6, method="forward")

F=summary(regfit.fwd)

## Plotting and choosing the subset

par(mfrow=c(2,2))

plot(F$rss ,xlab="Number of Variables ",ylab="RSS", type="l")

plot(F$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq",type="l")

coef(regfit.fwd, 5)

#best variables

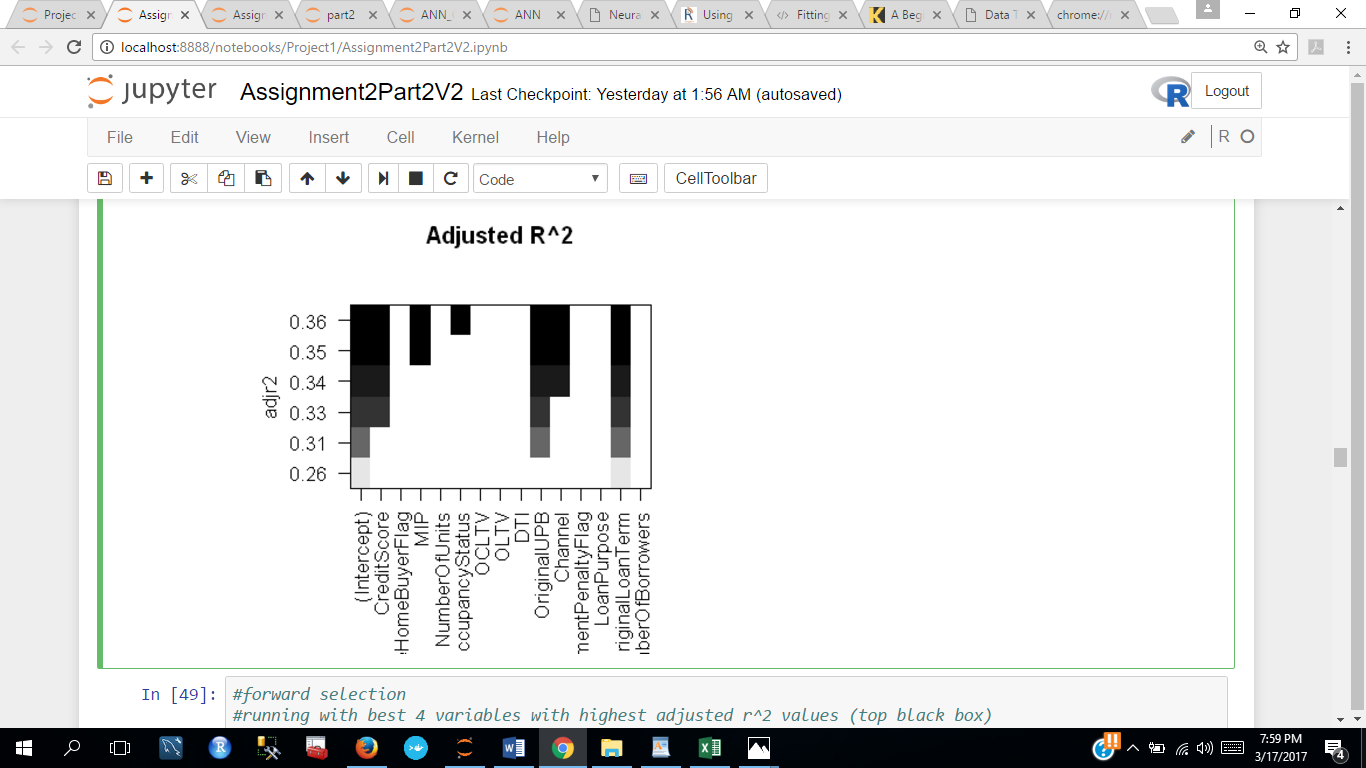
plot(regfit.fwd, scale = "adjr2", main = "Adjusted R^2")

### Backward Selection

Backward elimination: In backward elimination, we start with a "full" model that includes

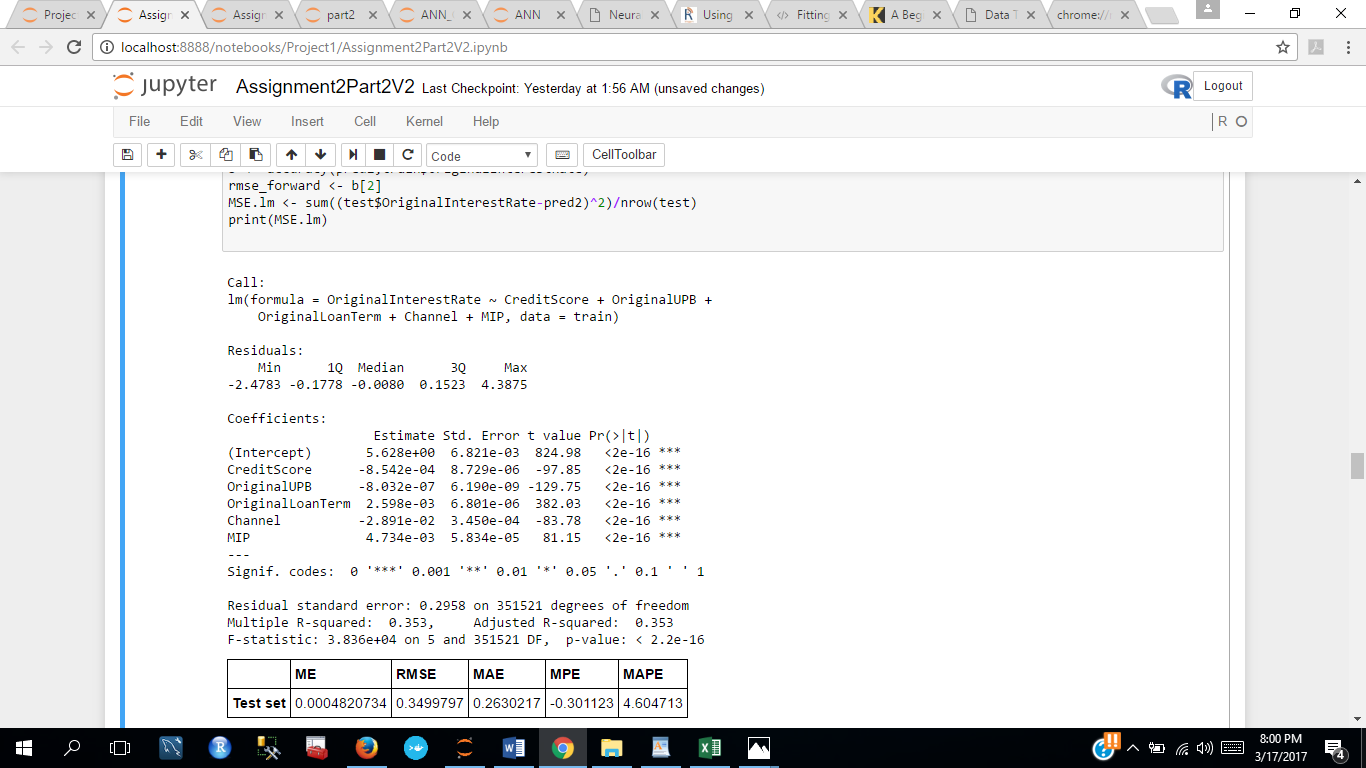
all variables. Independent variables with nonsignificant t-values (which is established a priori) are dropped off and a new model is built with the remaining variables. If all variables have significant t-values, then the procedure stops.

1. We ran forward search for 4,5,6,7,8 variables and got the best result with 5 variables which were the same as exhaustive search



1. We ran our model for these 5 variables which have the highest adjusted r^2 value

And got the adjusted r^2 as 0.353 and RMSE as 0.349



### Backward regression

In this technique, we start fitting the regression model with full variables and keep on deleting them until we find the variables are no more affecting the t-value.

#### Backward selection

regfit.bwd=regsubsets(OriginalInterestRate ~.,data=train,nvmax=6 , method="backward")

B=summary(regfit.bwd)

## Plotting and choosing the subset

par(mfrow=c(2,2))

plot(B$rss ,xlab="Number of Variables ",ylab="RSS", type="l")

plot(B$adjr2 ,xlab="Number of Variables ", ylab="Adjusted RSq",type="l")

coef(regfit.bwd, 6)

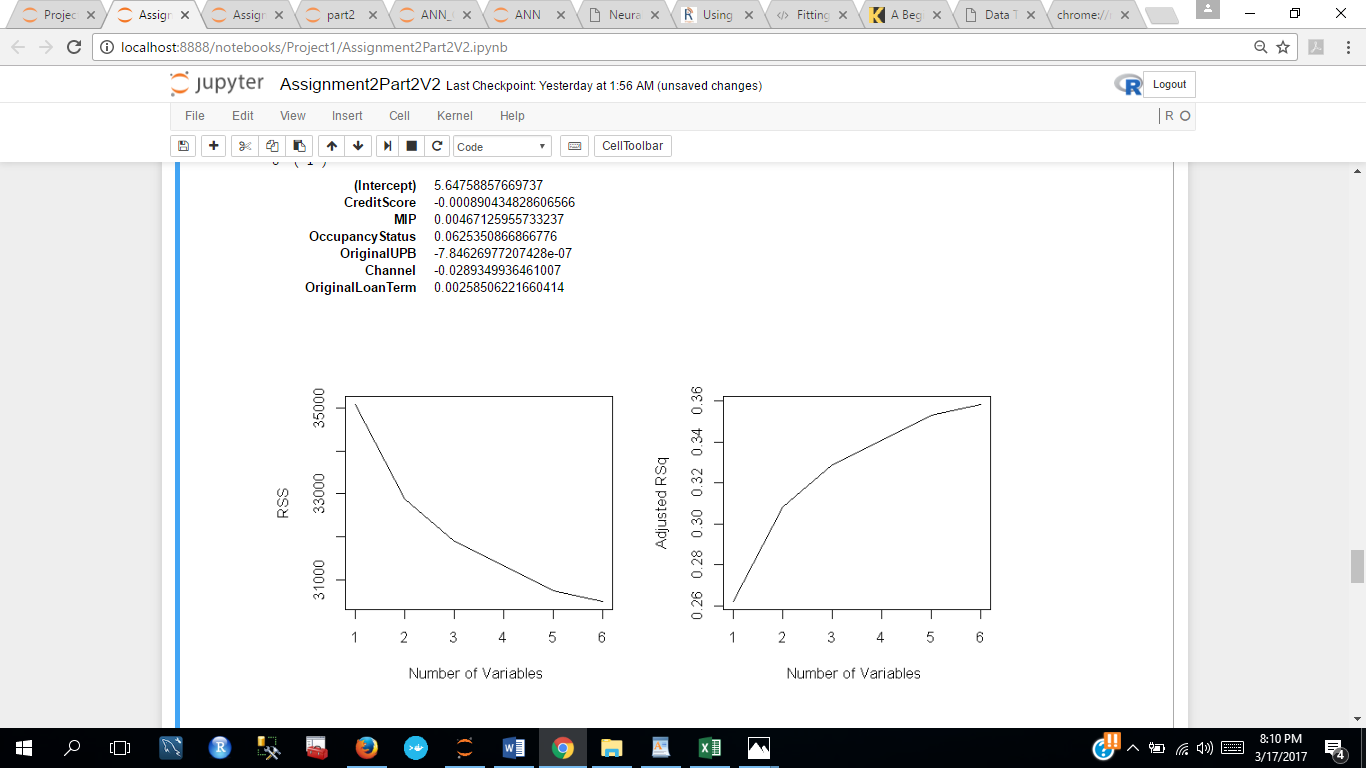
#best variables

plot(regfit.bwd, scale = "adjr2", main = "Adjusted R^2")

### Stepwise regression

In this technique, we start fitting the regression model with one single independent variable that has the largest t value. In the next step, we add a second variable and if the t-value is better than the previous, we keep the variable else we remove it and add a new variable. This process continues until the best combination of predictors are selected. This method is a combination of both forward and backward.

1. We ran backward search for 5,6,7,8 variables and got the best result with 5 variables which were the same as exhaustive search and forward search
2. For 6 variables, we got the following output:



1. We ran our model for these 6 variables which have the highest adjusted r^2 value

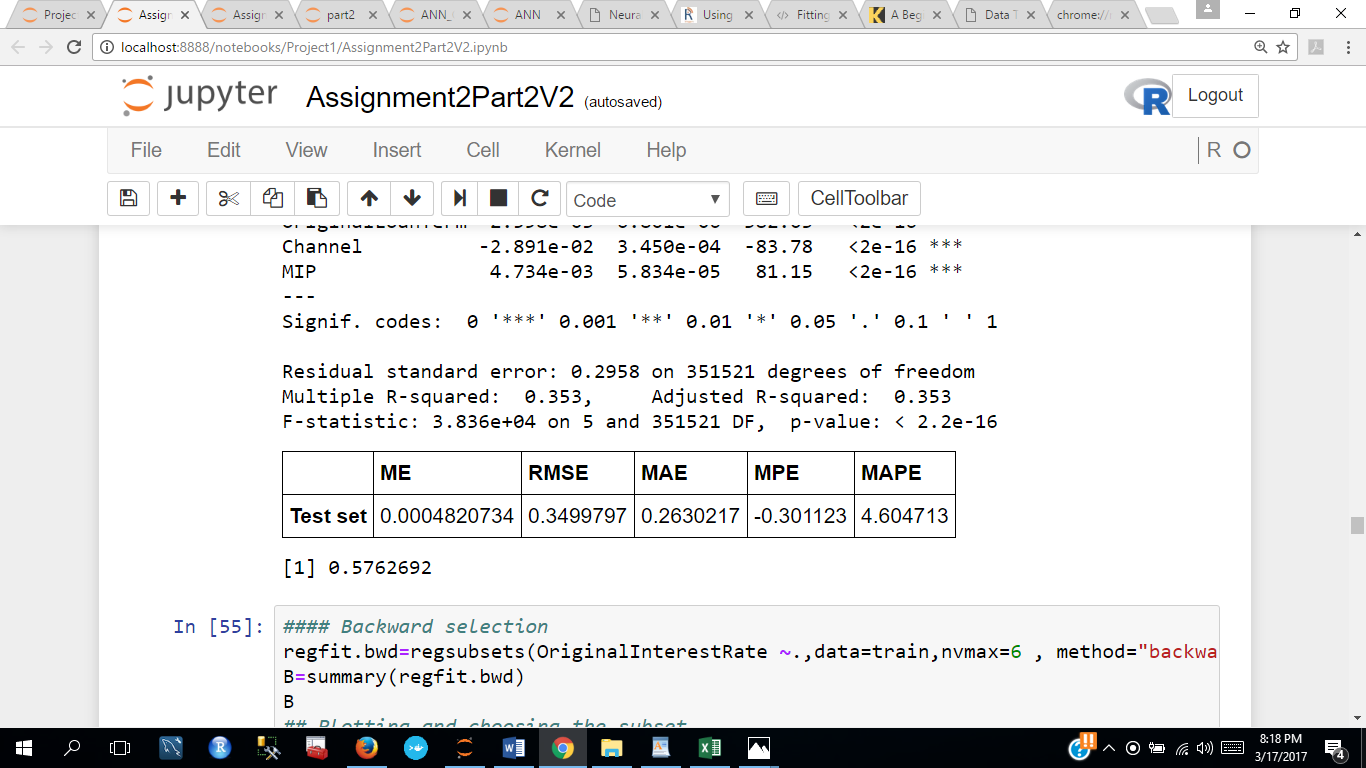
And got the adjusted r^2 as 0.3585 and RMSE as 0.35

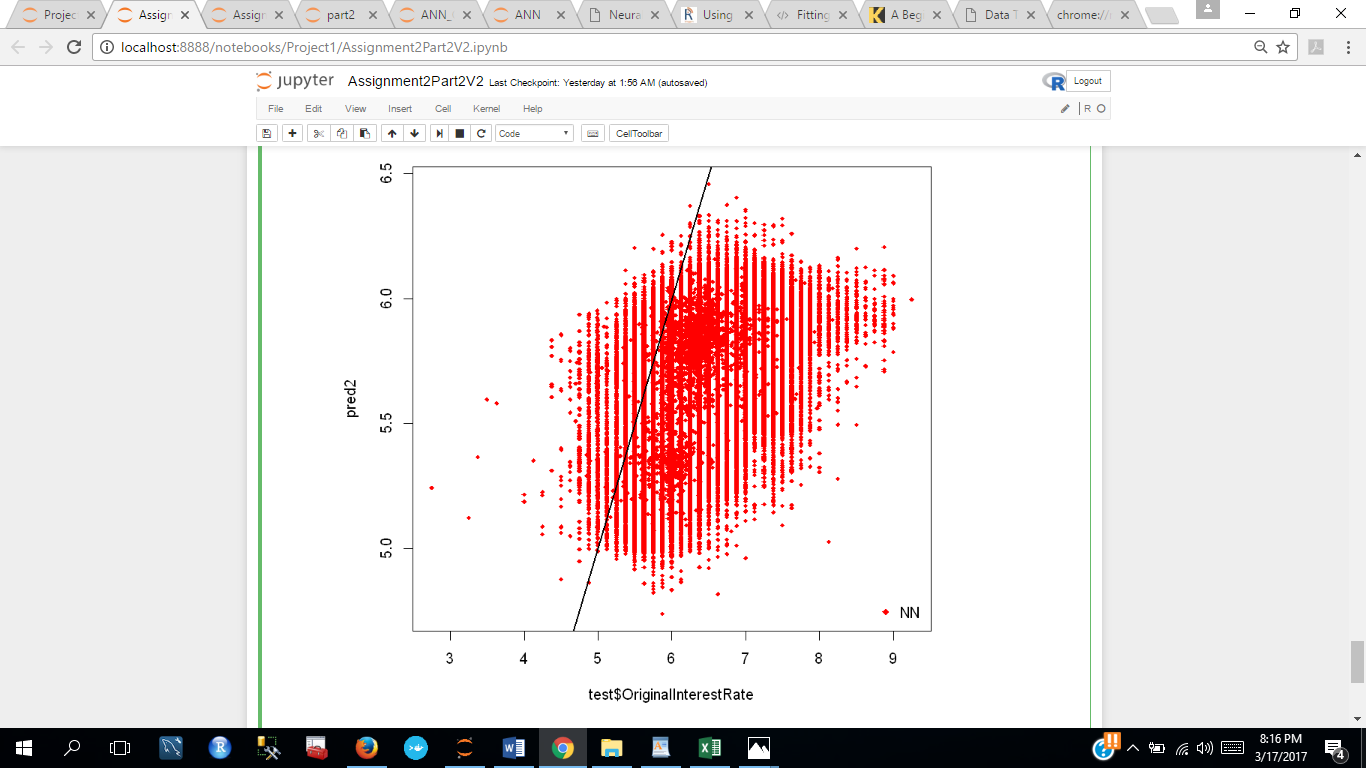
### Regression Model

* We selected the following 5 features after carefully considering all the variable selection techniques:
* Credit Score
* Original Unpaid Balance
* Original Loan Term
* Channel MIP

This is the output for the final regression model after considering the above 5 variables.

MSE: 0.576269170692854 (calculated)





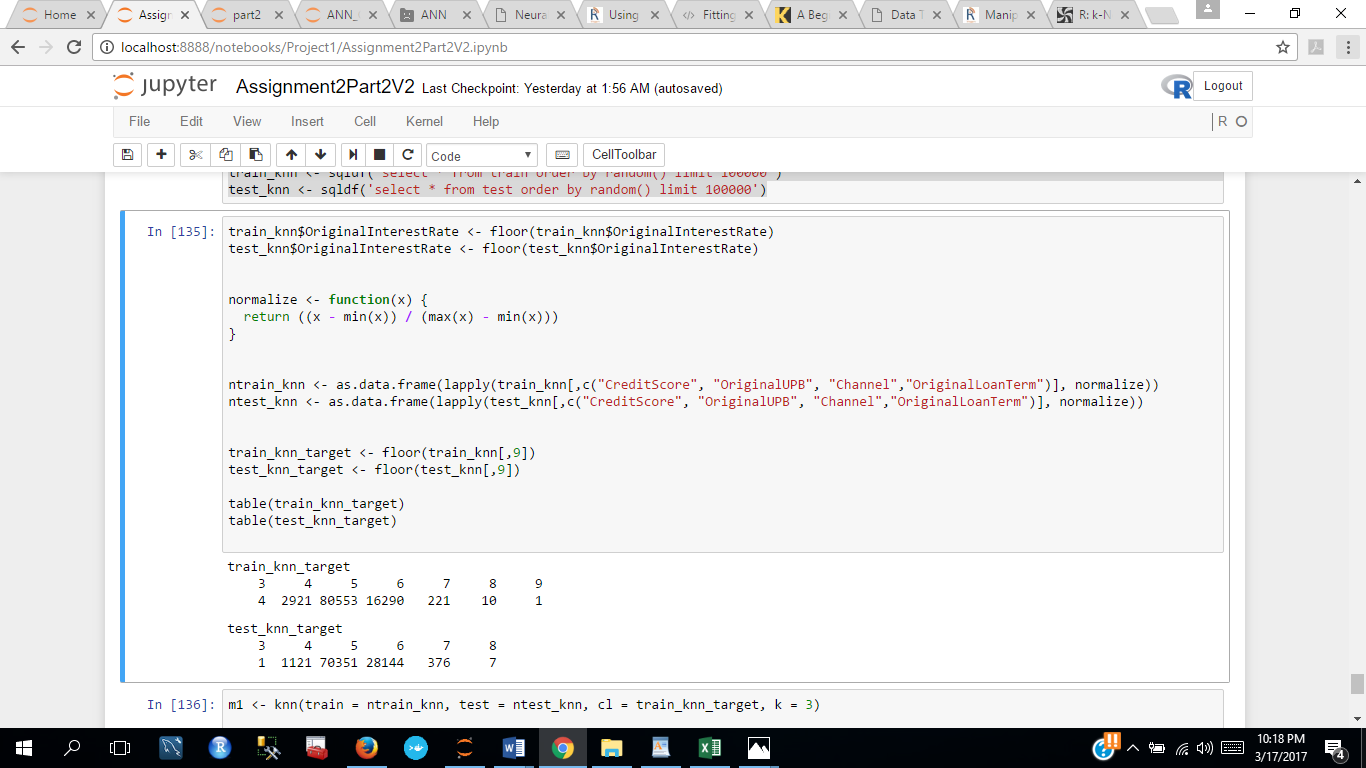
### KNN Algorithm

We ran the KNN algorithm for the selected features as below.

* Normalize the values to get all the values in the range of 0-1

normalize <- function(x) {return ((x - min(x)) / (max(x) - min(x)))}

* Our accuracy came out to be 0.26907



#KNN Algorithm

m1 <- knn(train = ntrain\_knn, test = ntest\_knn, cl = train\_knn\_target, k = 3)

(table(test\_knn\_target, m1))

sum(diag(t))/sum(t)

Accuracy : 0.26907

### Neural Network Model

* We built neural network model on the 5 variables chosen by variable selection as follows:

#Aritificial Neural Network Algorithm

train\_ann = train[,c(1,3,9,10,11,14)]

test\_ann = test[,c(1,3,9,10,11,14)]

n <- names(train\_ann)

f <- as.formula(paste("OriginalInterestRate ~", paste(n[!n %in% "OriginalInterestRate"], collapse = " + ")))

nn <- neuralnet(formula = f,data=train\_ann,hidden=c(4,2),linear.output=T)

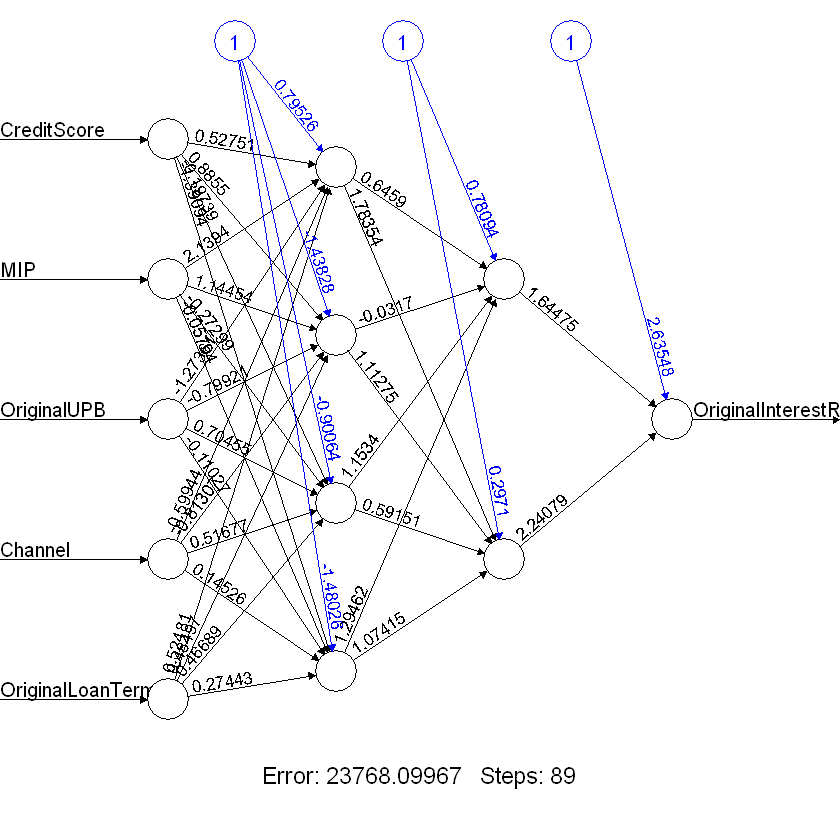
#nn$result.matrix

predict\_ann <- compute(nn,test\_ann[,-2])

plot(nn,rep="best")

MSE.nn <- sum((test\_ann$OriginalInterestRate - predict\_ann$net.result)^2)/nrow(test\_ann)

Output: MSE: Mean squared error value -> 0.681734390867288



### Random Forest Algorithm

* We applied random forest algorithm to the variables selected by variable selection above.
* We selected 100000 rows at random for training and testing to decrease the algorithm running time

#Random Forest Algorithm

#selecting 100000 rows randomly from train and test files for train and test

train\_for <- sqldf('select \* from train order by random() limit 100000')

test\_for <- sqldf('select \* from test order by random() limit 100000')

train\_for <- train\_for[,c(1,3,9,10,11,14)]

test\_for <- test\_for[,c(1,3,9,10,11,14)]

train\_for$OriginalInterestRate <- floor(train\_for$OriginalInterestRate)

test\_for$OriginalInterestRate <- floor(test\_for$OriginalInterestRate)

cols <- c("Channel", "OriginalInterestRate")

for(i in cols){

train\_for[,i] = as.factor(train\_for[,i])

}

for(i in cols){

test\_for[,i] = as.factor(test\_for[,i])

}

model\_random <- randomForest(OriginalInterestRate ~ ., data = train\_for, ntree = 20)

importance(model\_random)

varImpPlot(model\_random)

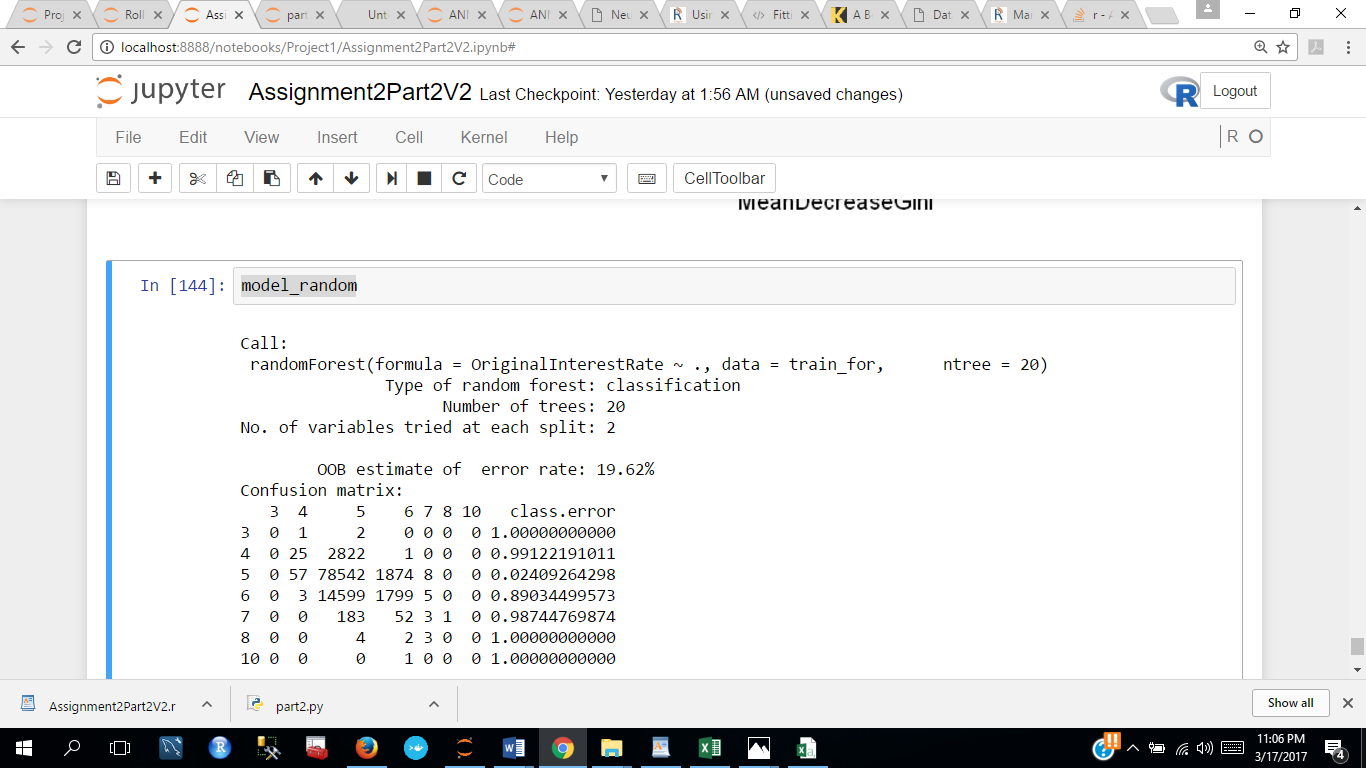
predictionwithclass <- predict(model\_random, test\_for, type = 'class')

t <- table(predictions=predictionwithclass, actual=test\_for$OriginalInterestRate)

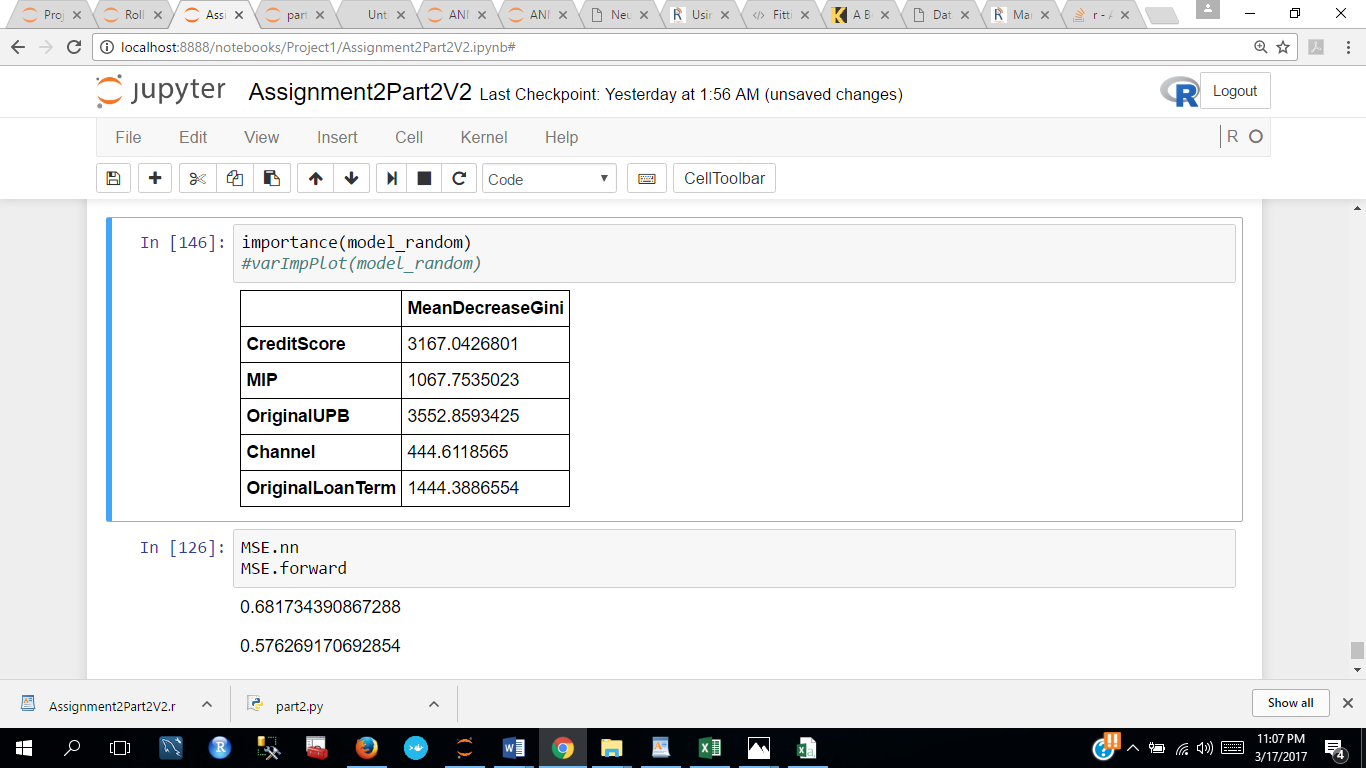
t

sum(diag(t))/sum(t)

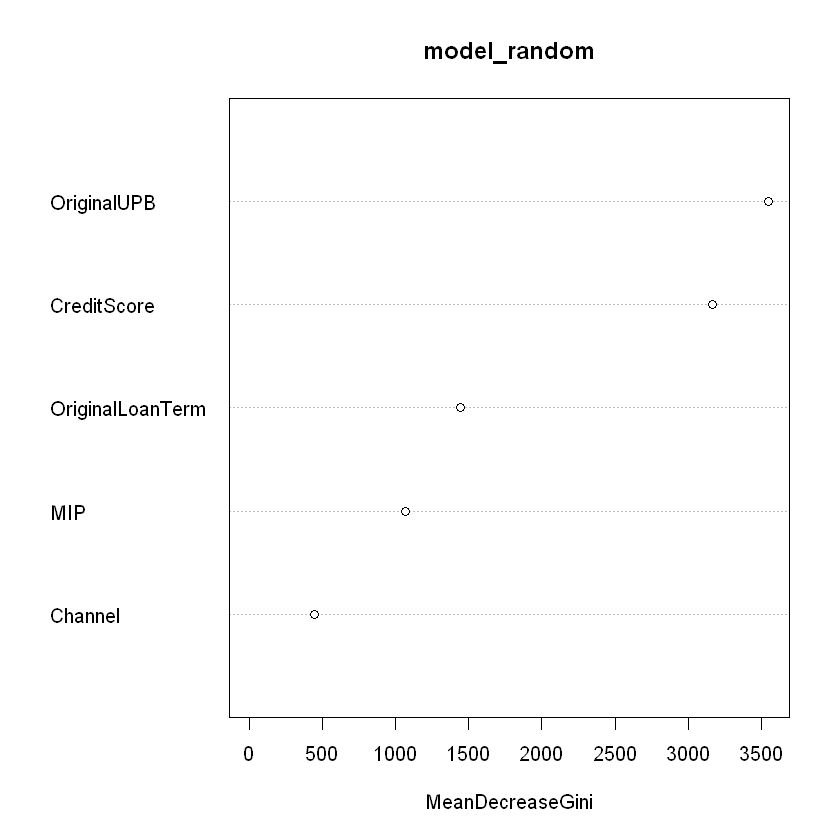
* Our model came out to be:



* Importance matrix is:



* Model plot for Random Forest:



* Accuracy is: sum(diag(t))/sum(t) -> 0.71103

### Summary:

* Regression Model gave us the best results with 5 variables
* Second best came out to Random Forest
* ANN gave us third best accuracy
* KNN gave us the least accuracy

### Financial Crisis

* We ran our best algorithm for the rolling quarters: Q12007, Q22007, Q32007, Q42007, Q12008
* We trained our algorithm for the quarters Q12007, Q22007, Q32007, Q42007
* Validated our model for the quarters Q22007, Q32007, Q42007
* We got an RMSE value of 0.447
* MSE value of ~0.226



* We also cross validated our result with our second best model: Random Forest
* We received an accuracy of ~65%

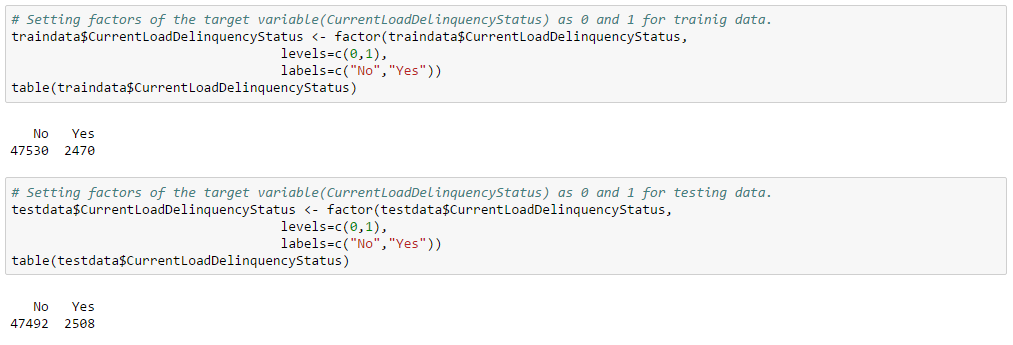
## Classification

### Data Cleaning and Preprocessing:

1. Imported data set from Q12005 as training data and Q22005 as test data programmatically.
2. **Preprocessed and cleaned** **data**: Converted all data entry in our target variable i.e. Current Delinquent Status > 0 to 1 and ==0 to 0.
3. Removed all the null rows in Current Delinquent Status column.
4. Replaced blank spaces in Repurchase flag, Zero Balance Code and Modification Flag with arbitrary values as they are not null. They are not applicable.
5. Dropped the unnecessary columns to speed up data retrieval.

### Logistic Regression Model:

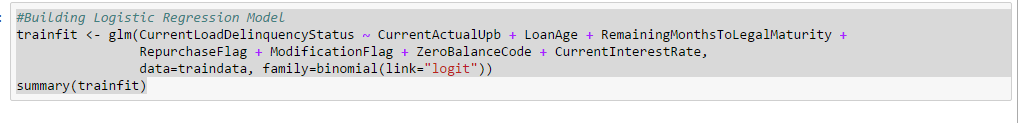
1. Converting the target variable to factors as Yes and No for the prediction.



1. After trying forward, backward and exhaustive selection, following variables were selected as the predictor variables:

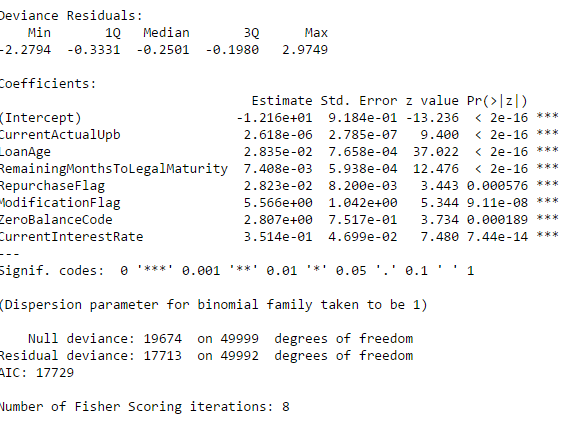
* CurrentActualUpb
* LoanAge
* RemainingMonthsToLegalMaturity
* RepurchaseFlag
* ModificationFlag
* ZeroBalanceCode
* CurrentInterestRate

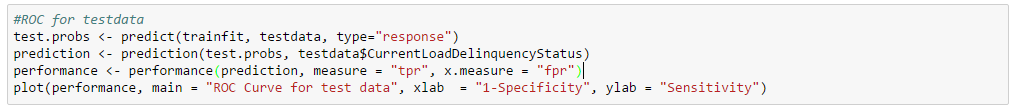
1. Build the logistic regression model using these X variables on the training data.

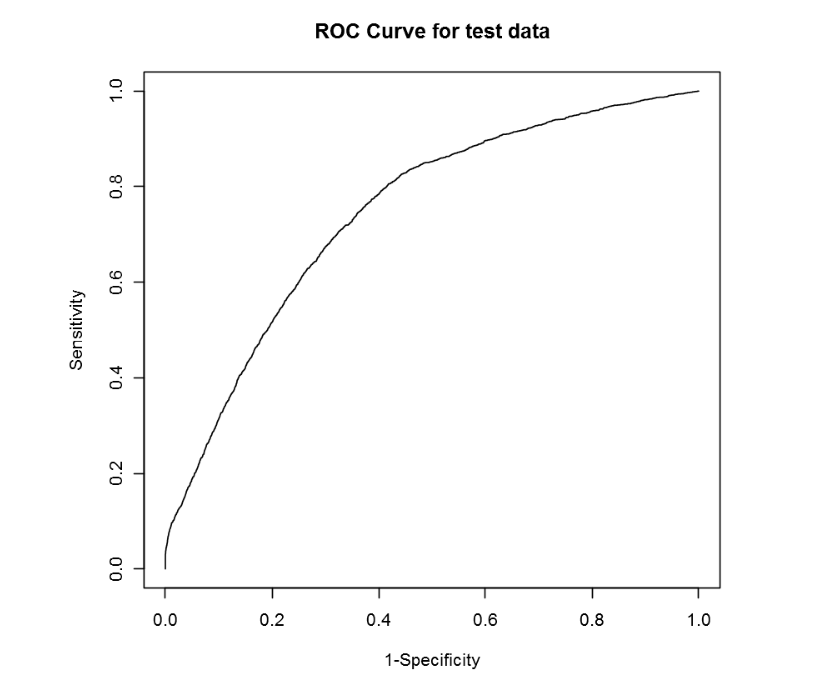


The output will be showing the variables with the coefficients value and other data.

Now we will create a ROC curve to check our prediction and its accuracy using the trained model i=on both test and train data.

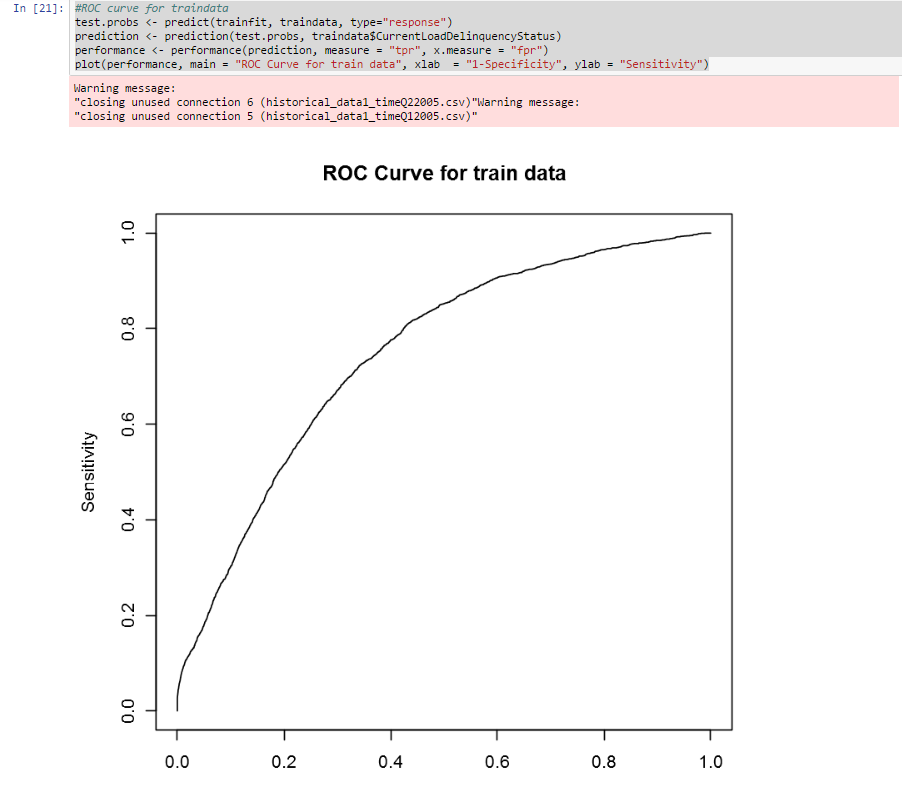


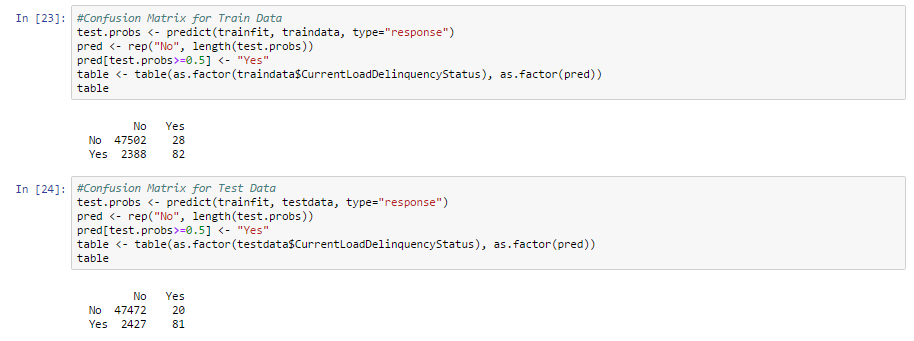




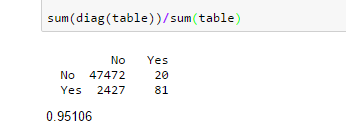
**ROC Curve for train data:**

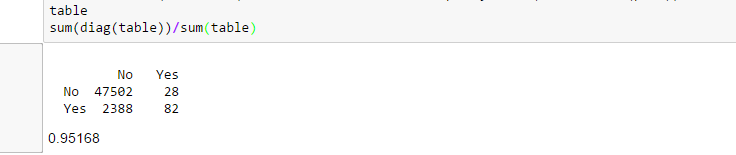
**Now**



* **Creating confusion matrix :**
* 

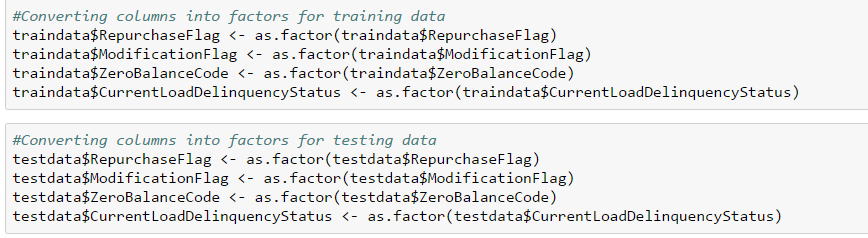
As shown above, we have predicted correct N0 47472 and correct YES 81 times for test data and correct N0 47502 and correct YES 82 times for test data. If we calculate our accuracy by dividing the total no of diagonal elements by total records it comes out to be 95 percent accurate.



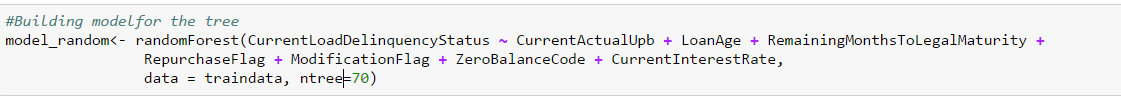


### Random Forest Model:

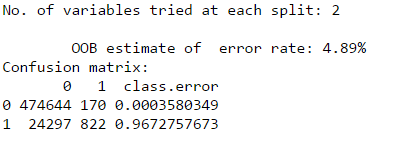
1. Import the training and test data set and save as a data frame.
2. **We will convert all our categorical variables to factors that will ease in tree based classification including the target variable.**



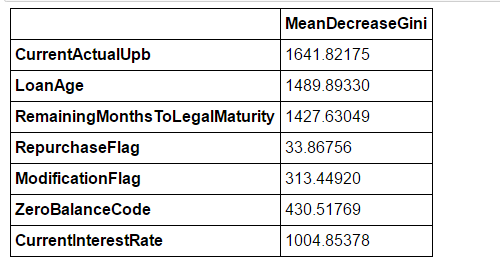
1. Build the model on the trained data using the previously selected variables in the logistic model.



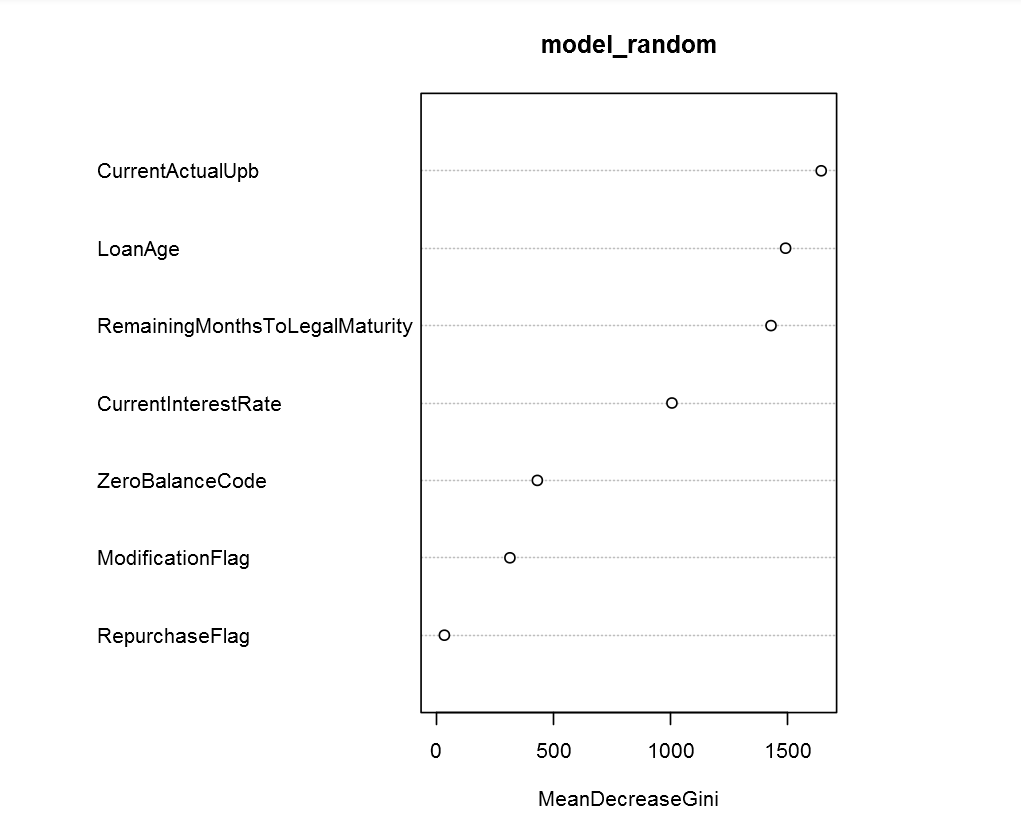
1. Check the output of the model build.



1. mtry variable is automatically assigned as square root of number of varaibles i.e. 2 in our case.
2. Now, we will check the important variables and plot a graph based on that.



1. This shows the important variables effecting our target variable. Will build a plot to understand more:



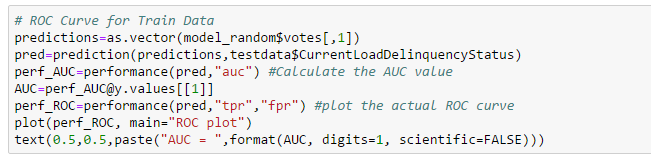
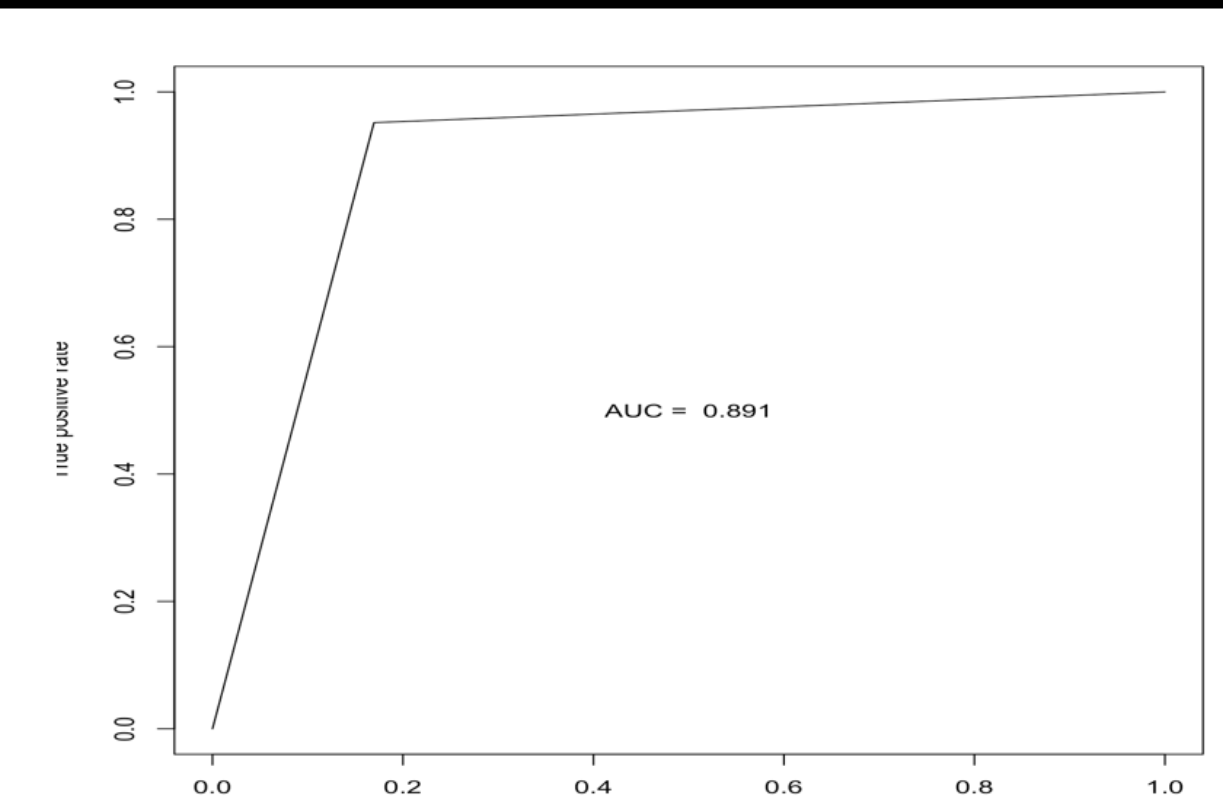
This shows how Current Actual UPB is most important followed by Loan Age and others. Repurchase flag is least important.

1. **Classification Matrix:**



**It shows that our model is 95.0 percent accurate.**

**ROC CURVE:**



* 1. Contribution

# References

1. https://www.creditsesame.com/blog/credit/credit-score-range-for-experian-transunion-equifax/. (n.d.).