Advances in Data sciences

Midterm Project

**Under the guidance of Professor Srikanth Krishnamurthy**

**Team 6:** Vaidehi Deshpande,

Puneet Kumar,

Yamini Sehrawat

**Table of Contents**

[Abstract 3](#_Toc477555952)

[Part 1: Data wrangling 5](#_Toc477555953)

[Data Download and pre-processing 5](#_Toc477555954)

[User account set up: 5](#_Toc477555955)

[Exception Handling: 6](#_Toc477555956)

[Cleaning data: 6](#_Toc477555957)

[Summarization: 12](#_Toc477555958)

[Flow Chart: 15](#_Toc477555959)

[Problem 1 15](#_Toc477555960)

[Tableau Charts Entire Dataset: 16](#_Toc477555961)

[Delinquency Trends by State and Year 16](#_Toc477555962)

[UPB by State and Year: 17](#_Toc477555963)

[Analysis for one year 18](#_Toc477555964)

[Origination File 2003 18](#_Toc477555965)

[Percentage Contribution of different Property Types 18](#_Toc477555966)

[Top 20 states by Total Count and respective Average Credit Scores: 18](#_Toc477555967)

[Quarterly Analysis: 19](#_Toc477555968)

[Performance File 19](#_Toc477555969)

[Actual Loss and DAI per quarter: 19](#_Toc477555970)

[Delinquency Trends as per quarter: 21](#_Toc477555971)

[Python Graphs: 21](#_Toc477555972)

[Delinquency Trends by Year 22](#_Toc477555973)

[Part II: Building and evaluating models 23](#_Toc477555974)

[Prediction 23](#_Toc477555975)

[Dataset: 24](#_Toc477555976)

[Classification: 28](#_Toc477555977)

[Script 1 Approach: 28](#_Toc477555978)

[Preprocess Data: 29](#_Toc477555979)

[Handling Missing Data: 30](#_Toc477555980)

[Handle Categorical Variables: 30](#_Toc477555981)

[Run Classification algorithms on 13 Features: 31](#_Toc477555982)

[Recursive Feature Elimination: 33](#_Toc477555983)

[Selecting Best Algorithm: 35](#_Toc477555984)

[Script 2 (To Compute Matrix for all the Quarters with best model generated out of Script 1): 36](#_Toc477555985)

[Docker Image: 36](#_Toc477555986)

**US Housing and Urban Development**

# Abstract

The United States Department of Housing and Urban Development is a Cabinet department in the Executive branch of the United States federal government. As a Data Scientist, we must understand the US housing market. We are given Single-Family Loan Data to analyze and asked to present our results. Then, using the datasets, build predictive analytics models.

Link for Single-Family Loan Dataset: <http://www.freddiemac.com/news/finance/sf_loanlevel_dataset.html>

# Part 1: Data wrangling

## Data Download and pre-processing

The first step here is to programmatically download the data from: <https://freddiemac.embs.com/FLoan/Data/download.php>

### User account set up:

This link will redirect automatically to <https://freddiemac.embs.com/FLoan/secure/login.php> where the user should enter his credentials for Freddiemac website to download the files. If the user account does not exist already, user will get an error message asking to sign up using his/her email ID. The password will be mailed to this email ID. The password cannot be changed later. Once user account is setup the required (Sample\_<year>.zip) files can be pre-processed and downloaded.

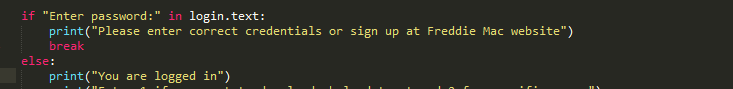
We have used the requests module from Python. By creating a session we are passing the login credentials of the user and saving the cookies and allowing the user to login to the page.

Below is the code snippet of the same:

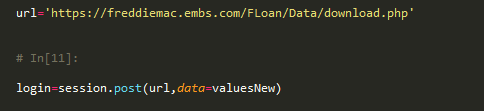


### Exception Handling:

In case the user account is not already created on Freddie Mac website an error message will be displayed and program will not run ahead. Following condition checks whether the credentials are valid:



Here login variable contains the source code of open web page:



### Cleaning data:

We calculated the percentage of NULLS in all the columns in the data frame using following code:

mis\_val\_percent = 100 \* df.isnull().sum()/len(df)

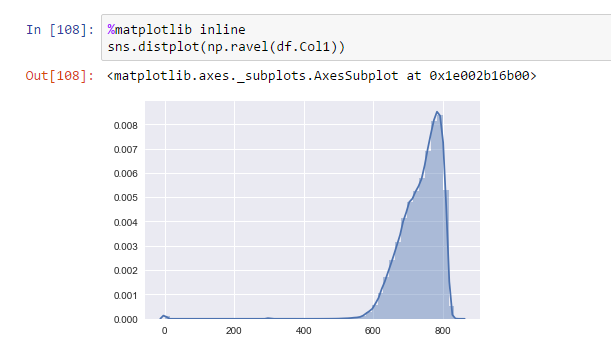
Following rules have been applied for handling missing data and cleaning unwanted data:

#### Origination File:

* Credit Score: Outlier detection was done by plotting normal distribution graph.

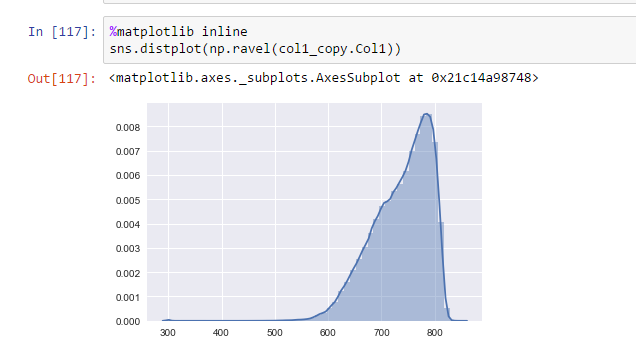
1. By replacing NULL values by 0:

It was observed that data was skewed and we can see a small jump at 0 which is incorrect:



1. Graph after replacing NULL values by mean value:

After the NULL values were replaced by “mean” value, the graph spans a larger area and we can see most of the values lie between 550 and 810:



1. Replacing values below 301 by mean:



* First Payment Date:

We did not find any missing values or NULL values for this column

* First Time Home Buyer Flag:

As mentioned in the User Guide, an empty space or a NULL value means “Not Applicable” or “Unknown”. Hence the NaN is replaced with string: ‘Not Applicable’

* Maturity Date:

We did not find any blank spaces or NULL values for this column

* MSA Code:

As per User Guide there are following meanings of an empty space:

* Indicates that the area in which property is located is neither MSA nor Metropolitan Division
* Unknown

Here since we do not know where exactly the MSA is unknown we have filled the empty spaces with “mode” value from this column since this is a categorical variable.

* MI percentage:

As per user guide empty space means MI percentage is unknown. This being a continuous variable we have filled empty spaces with “mean” of the values in this column.

* Number of units:

Though this is a numerical variable, it defines the categories in which the property types can be divided based on number of units. Since, the empty space here means value is unknown, we have entered the “mode” value for this column

* Occupancy Status:

This column contains string values which define the type of property depending on Occupancy Status. We have entered “mode” of the values in this column to replace the blank spaces.

* Original Combined LTV:

Space in this case means the CLTV value is < 0 or > 200 and LTV is < 80 or > 200 or unknown.

Since we cannot predict what exactly a blank space for given entry means, we have replaced all NULL values with “mode” of the values from the column

* DTI ratio:

Here blank space means the value of DTI ratio is greater than 65%, for these rows we have entered value of 70% and the rows where value was NULL/NaN, it is being replaced by mean value of the column

* Original UPB:

We did not find any blank spaces or NULL values for this column

* Original Loan-To-Value (LTV):

Space here means Ratios below 6% or greater than 105%. These spaces are replaced using “mode” value.

* Original Interest Rate:

We did not find any blank spaces or NULL values for this column

* Channel:

Here blank space means that the Channel through loan has been issued. If it empty, we have considered the mode value from the column

* Prepayment Penalty Mortgage (PPM):

This is a categorical column and hence we have replaced the empty spaces with mode value from the column

* Product Type:

We did not find any empty spaces or NULL values for this column

* Property State:

We did not find any empty spaces or NULL values for this column

* Property Type

We did not find any empty spaces or NULL values for this column

* Postal Code

The empty spaces here have been replaced with a Median value for the column

* Loan Sequence Number

We did not find any empty spaces or NULL values for this column

* Loan Purpose:

We did not find any empty spaces or NULL values for this column

* Original Loan Term:

We did not find any empty spaces or NULL values for this column

* Number of Borrowers:

The empty spaces or NULL values have been replaced by “mode” value

* Seller Name:

We did not find any empty spaces or NULL values for this column

* Service Name:

We did not find any empty spaces or NULL values for this column

* Super Conforming Flag:

As per definition in user guide, empty space here means, Not Super Conforming, hence the blank spaces have been replaced by ‘N’ in this column.

#### Performance File

##### Actual Loss and Delinquent Accrued Interest Calculation:



##### Missing data handling rules applied to all columns:

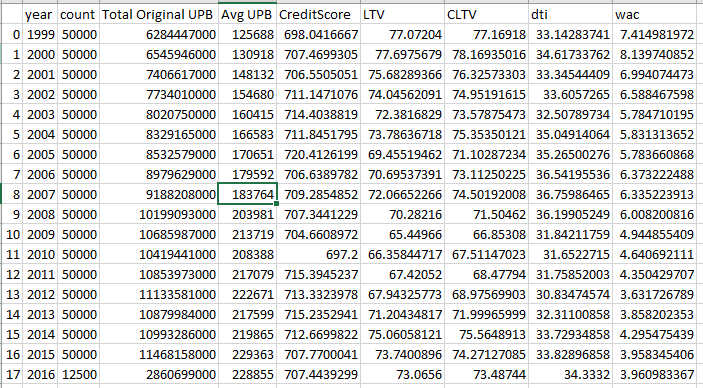
|  |  |
| --- | --- |
| Columns and Misisng data % | Rules applied |
| Col1 0.000000 | No missing data |
| Col2 0.000000 | No missing data |
| Col3 0.000000 | No missing data |
| Col4 0.000000 | No missing data |
| Col5 0.000000 | No missing data |
| Col6 0.000000 | No missing data |
| Col7 97.875986 | NA |
| Col8 99.984409 | N |
| Col9 97.876159 | 10 |
| Col10 97.876159 | 190012 |
| Col11 0.000000 | No missing data |
| Col12 0.000000 | No missing data |
| Col13 99.886357 | 190012 |
| Col14 99.969813 | 0 |
| Col15 99.969813 | 0 |
| Col16 99.969813 | 0 |
| Col17 99.969813 | Col17 = df.Col18 + df.Col19 +df.Col20 +df.Col21 |
| Col18 99.969813 | 0 |
| Col19 99.969813 | 0 |
| Col20 99.969813 | 0 |
| Col21 99.969813 | 0 |
| Col22 99.969813 | Calculate |
| Col23 | Calculate |
| year 0.000000 | No missing data |

### Summarization:

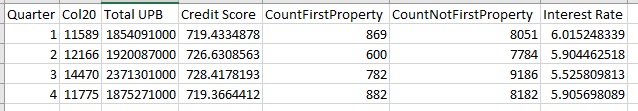
Following Summary statistics have been generated for Origination and Performance files:

1. **Origination File:**

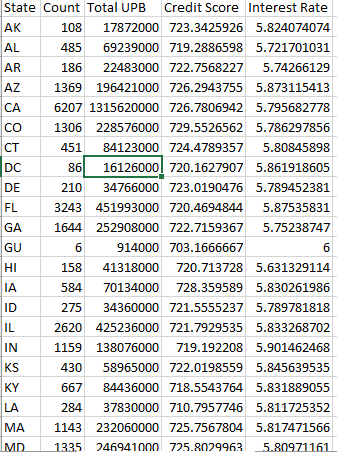
* For all years:



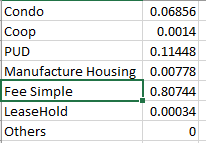
* For 2003 year:



* Analysis as per location:

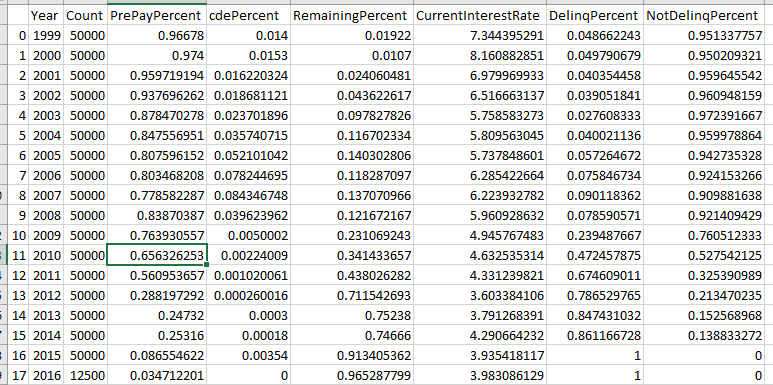


* Percentage contribution by Property Types:

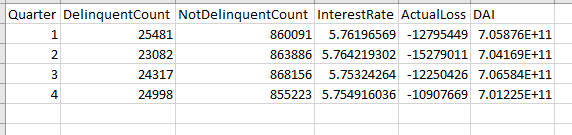


1. **Performance File:**

* For all years:



* For 2003 year:



# Flow Chart:

## Problem 1

Start

Enter username and password

Check if user account exists

**N**

**Y**

Direct to download URL

Display error message and ask user to enter credentials again

Fetch the files named as “Sample\_<year>.zip”

Parse the web page to access subfiles in sample file zip folder

Preprocess and clean the data inside Origination and Performance files

Summarize the critical information in the files that needs to be presented to management

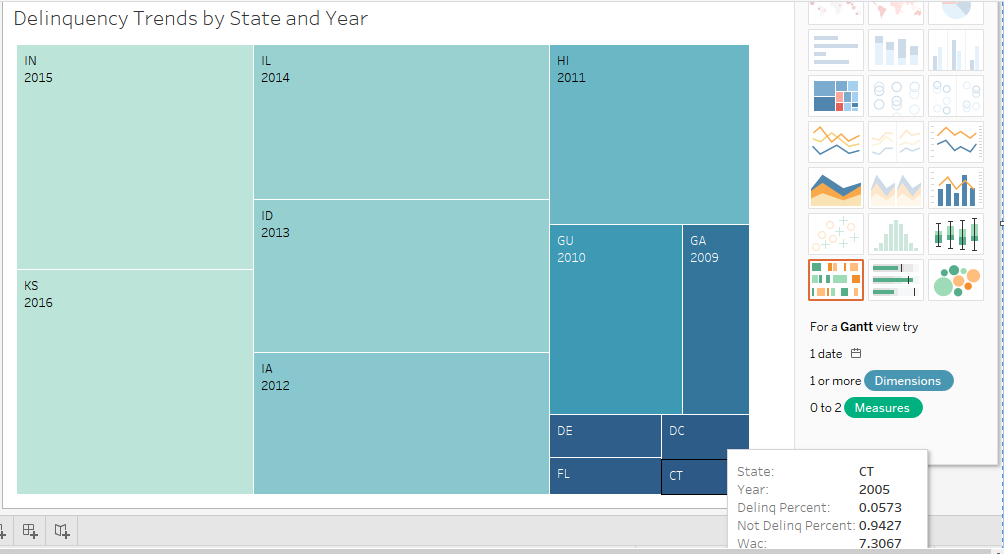
Plot the relevant graphs in Python and Tableau

Stop

## Tableau Charts Entire Dataset:

### Delinquency Trends by State and Year

Years 2005-2016

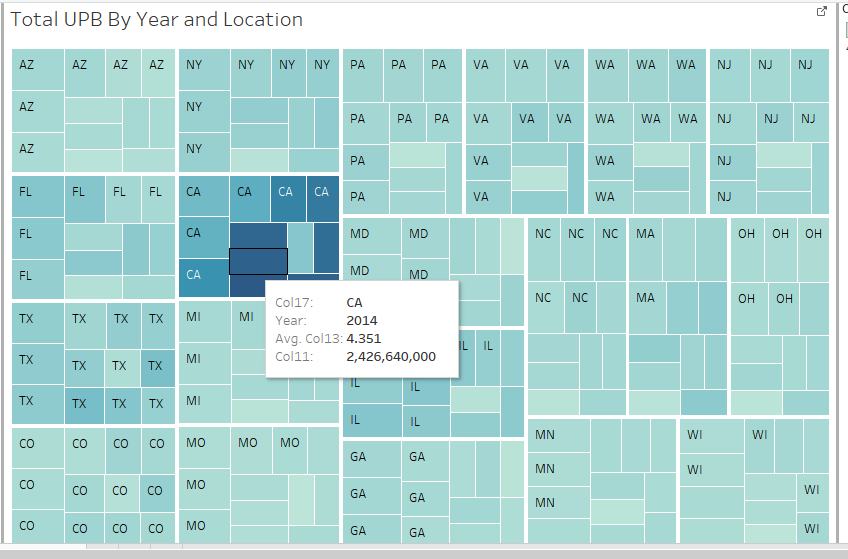


##### Tableau public link:

<https://public.tableau.com/views/DelinquencyByState/Sheet1?:embed=y&:display_count=yes>

### UPB by State and Year:

Top 20 states 2005 onwards:



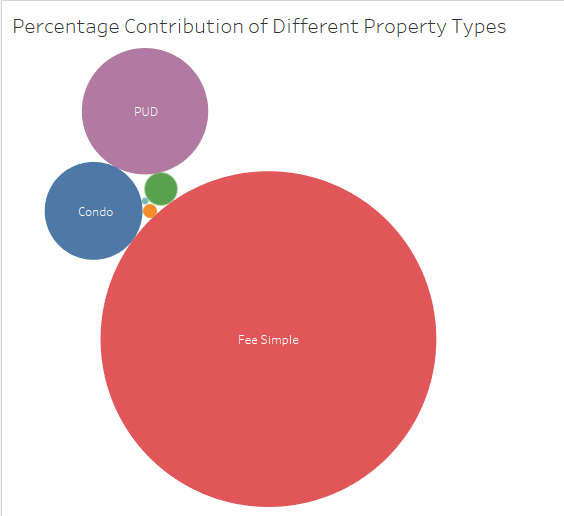
##### Tableau Public link:

<https://public.tableau.com/views/UPBbyYearandstate/Sheet1?:embed=y&:display_count=yes>

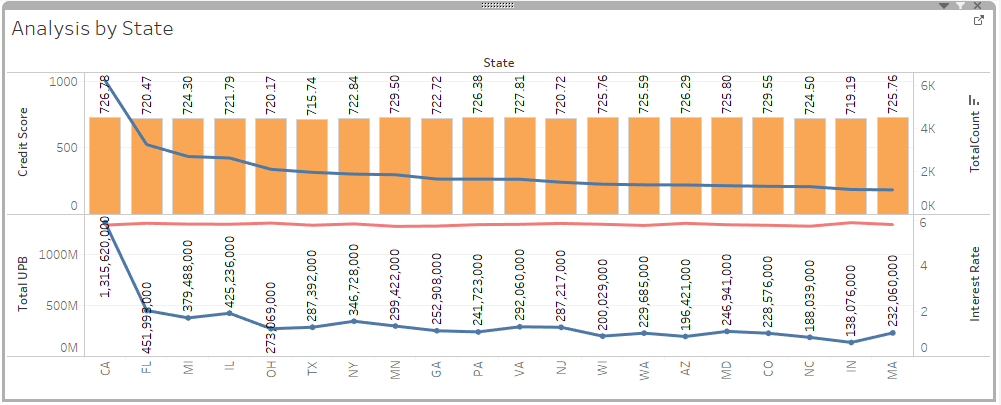
# Analysis for one year

## Origination File 2003

### Percentage Contribution of different Property Types



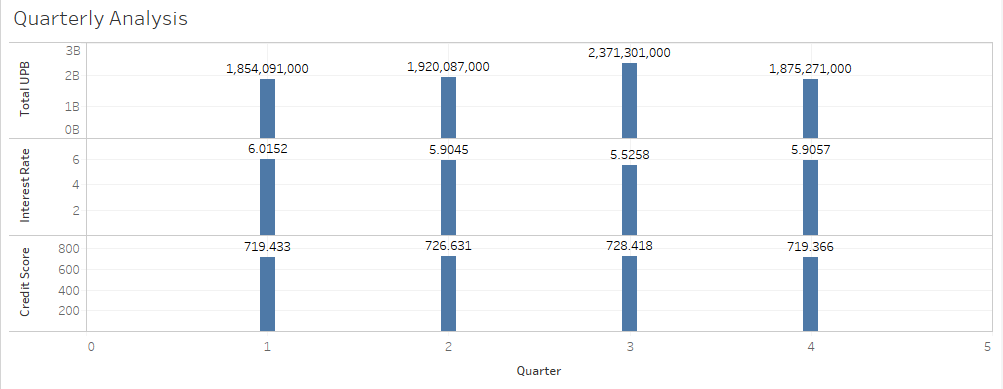
### Top 20 states by Total Count and respective Average Credit Scores:



##### Tableau Public link:

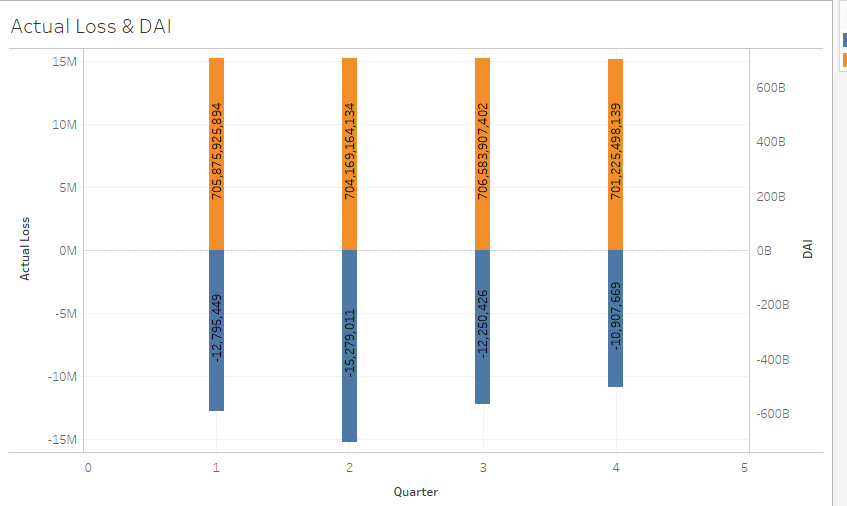
<https://public.tableau.com/views/ByState_Quarter/Sheet3?:embed=y&:display_count=yes>

### Quarterly Analysis:



## Performance File

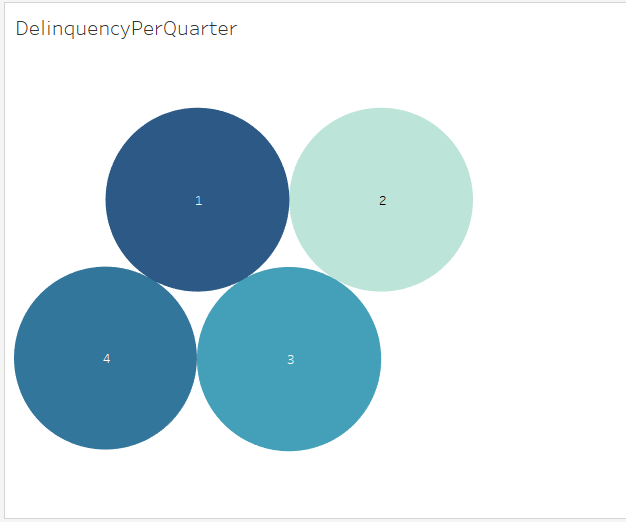
### Actual Loss and DAI per quarter:



##### Tableau Public link:

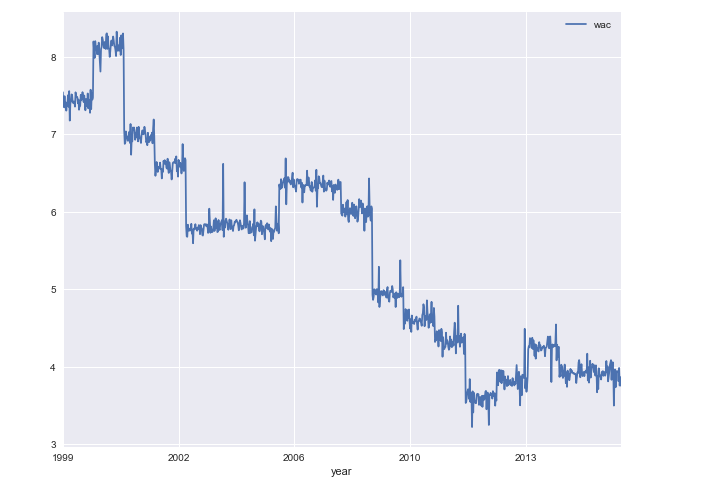
<https://public.tableau.com/views/PerformanceDelinquencyTrendsFor2003/Sheet2?:embed=y&:display_count=yes>

### Delinquency Trends as per quarter:

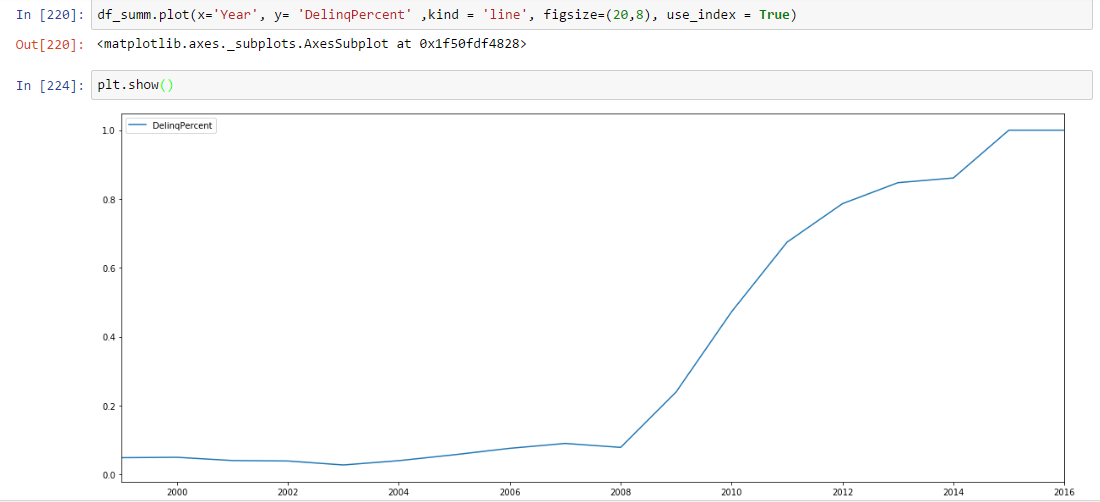


## Python Graphs:

Weighted Average of Interest Rate per year:



### Delinquency Trends by Year



# Part II: Building and evaluating models

## Prediction

It is given that work is now shared with economists and other experts who are intrigued by various parameters.

They are interested in predicting interest rates in the next quarter based on information from the origination data from the prior quarter. In addition, they are interested in predicting whether a record is delinquent or not based on performance data. Fortunately, we have quarterly datasets readily available on the same data download page.

We are to write a prediction script in a Jupyter notebook that given input (For example Q12005),

* Programmatically downloads Q12005 and Q22005 origination data and pre-processes it.
* Builds a Regression model for the interest rate using Q12005 data as training data (col 13)
* Does variable selection to choose the best Regression model using Forward, Backward, Stepwise and Exhaustive search methods.
* Validates against the Q22005 datasets
* Computes MAE, RMS, MAPE for training and testing dataset
* Repeat this using Random Forest, Neural Network models and KNN algorithms.
* Choose the best model amongst the 4 types of algorithms.

We are asked to do what-if analysis had the algorithm used in various scenarios:

Financial crisis (https://www.stlouisfed.org/financial-crisis/full-timeline)

---Run the algorithm for 4 rolling quarters and report your findings and discuss it in your report. (i.e Use Q12007, Q22007, Q32007, Q42007 for training and predict for Q22007, Q32007, Q42007, Q12008)

---Run your algorithm 2 years later (i.e, 2009 for all 4 quarters)

o Economic boom (1999, 2013) (https://www.thebalance.com/stock-market-returns-by-year2388543) Discuss your design and results in a report. Would you recommend using this model for the next quarter? Justify

### Dataset:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **File Name Format** | **Contents** | **File Type** | **Delimiter** |
| Full | historical\_data1\_QnYYYY.zip | historical\_data1\_QnYYYY.txt, | Origination Data, | Pipe(“|”) |
| historical\_data1\_time\_QnYYYY.txt | Monthly Performance Data |

**For Prediction we will be working on Origination Data File: historical\_data1\_QnYYYY.txt**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Position** | **Column Name** | **Valid Values** | **Type** | **Length** |
| 1 | Credit Score | * 301 - 850 * Space (3) = Unknown, if Credit Score is < 301 or > 850. | Numeric | 3 |
| 2. | FIRST PAYMENT DATE | YYYYMM | Date | 6 |
| 3. | FIRST TIME HOMEBUYER FLAG | * Y = Yes * N = No * Space (1) = Unknown or Not Applicable | Alpha | 1 |
| 4. | MATURITY DATE | YYYYMM | Date | 6 |
| 5. | METROPOLITAN STATISTICAL AREA (MSA) OR METROPOLITAN DIVISION | * Metropolitan Division or MSA Code. * Space (5) = Indicates that the area in which the mortgaged property is located is a) neither an MSA nor a Metropolitan Division, or b) unknown. | Numeric | 5 |
| 6. | MORTGAGE INSURANCE PERCENTAGE (MI %) | * 1% - 55% * 000 = No MI * Space(3) = Unknown | Numeric | 3 |
| 7. | NUMBER OF UNITS | * 1 = one-unit * 2 = two-unit * 3 = three-unit * 4 = four-unit * Space(1) = Unknown | Numeric | 1 |
| 8. | OCCUPANCY STATUS | * O = Owner Occupied * I = Investment Property * S = Second Home * Space (1) = Unknown | Alpha | 1 |
| 9. | ORIGINAL COMBINED LOAN-TO-VALUE (CLTV) | * 0% - 200% * Null = Unknown | Full Dataset: Numeric Literal Decimal; Sample Dataset: Numeric | Full Dataset: 7; Sample Dataset: 3 |
| 10. | ORIGINAL DEBT-TO-INCOME (DTI) RATIO | * 0%<DTI<=65% * Space(3) >65% * Null = Unknown | Numeric | 3 |
| 11. | ORIGINAL UPB | * Amount will be rounded to the nearest $1,000 | Numeric | 12 |
| 12. | ORIGINAL LOAN-TO-VALUE (LTV) | * 6% - 105% * Space(3) = Unknown | Numeric | 3 |
| 13. | ORIGINAL INTEREST RATE |  | Numeric Literal decimal | 6 |
| 14. | CHANNEL | * R = Retail * B = Broker * C = Correspondent * T = TPO Not Specified * Space = Unknown | Alpha | 1 |
| 15. | PREPAYMENT PENALTY MORTGAGE (PPM) FLAG | * Y = PPM * N = Not PPM * Space (1) = Unknown | Alpha | 1 |
| 16. | PRODUCT TYPE | * FRM – Fixed Rate Mortgage | Alpha | 5 |
| 17. | PROPERTY STATE | * AL, TX, VA, etc | Alpha | 2 |
| 18. | PROPERTY TYPE | * CO = Condo * LH = Leasehold * PU = PUD * MH = Manufactured Housing * SF = 1-4 Fee Simple * CP = Co-op * Space (2) = Unknown | Alpha | 2 |
| 19. | POSTAL CODE | * ###00, where “###” represents the first three digits of the 5digit postal code * Space(5)= Unknown | Numeric | 5 |
| 20 | LOAN SEQUENCE NUMBER | F1YYQnXXXXXX   * F1 = product (Fixed Rate Mortgage); * YYQn = origination year and quarter; and, * XXXXXX = randomly assigned digits | Alpha numeric | 12 |
| 21 | LOAN PURPOSE | * P = Purchase * C = Cash-out Refinance * N = No Cash-out Refinance * Space = Unknown | Alpha | 1 |
| 22. | ORIGINAL LOAN TERM | * Calculation: (Loan Maturity Date (MM/YY) – Loan First Payment Date (MM/YY) + 1) | Numeric | 3 |
| 23. | NUMBER OF BORROWERS | * 01 = 1 borrower * 02 = > 1 borrowers * Space (2) = Unknown | Numeric | 2 |
| 24. | SELLER NAME | Name of the seller, or “Other Sellers | Alpha-Numeric | 20 |
| 25. | SERVICER NAME | Name of the servicer, or “Other Servicers” | Alpha-Numeric | 20 |
| 26. | Super Conforming Flag | * Y = Yes * Space (1) = Not Super Conforming | Alpha | 1 |

**Data Downloading**

Data is downloaded programmatically for prior and next quarter.

For ex: If user enters Quarter: Q1 and year:2005

Then program will download **Q12005,** and immediate next quarter **Q22005**.

The data is downloaded as CSV Files: **priorQuarter.csv and nextQuarter.csv**

User will input the quarter and year for which he wants to download the data.

We are using Q12005 as Training and Q22005 as testing dataset.

## Linear Regression

**Q12005 data as training data on Interest\_Rate variable**

We checked the percentage of missing values in the dataset:

Credit\_Score 0.000000

First\_Payment\_Date 0.000000

First\_Time\_Homebuyer\_Flag 12.125184

Maturity\_Date 0.000000

MSA 15.578881

MI 0.000000

Number\_Of\_Units 0.001137

Occupancy\_Status 0.000000

CLTV 0.005402

DTI\_Ratio 0.129073

Original\_UPB 0.000000

LTV 0.005402

Interest\_Rate 0.000000

Channel 0.000000

PPM 0.094672

Product\_Type 0.000000

Property\_State 0.000000

Property\_Type 0.000000

Postal\_Code 0.000569

Loan\_Sequence\_Nmber 0.000000

Loan\_Purpose 0.000000

Original\_Loan\_Term 0.000000

Number\_Of\_Borrowers 0.031557

Seller\_Name 0.000000

Service\_Name 0.000000

Building a model with Q12005 cleansed dataset. A linear regression model is implemented with all the attributes. The measures of predictive accuracy as:

lm.coef\_, lm.intercept\_, R-Squared

[ -8.28770393e-04 3.55994452e-03 1.72242789e-02 -2.15325504e-03

5.37975918e-07 4.48834070e-03 4.04610256e-02 1.39821748e-01

9.87169836e-04 -4.72687540e-04 -7.57521843e-07 1.95486614e-04

-7.78179076e-02 2.99245568e-01 -1.94289029e-16 -1.22111098e-03

1.78978075e-02 -1.04109371e-07 2.45139822e-07 -3.91336447e-02

2.07563314e-02 -2.57882810e-02]

-276.608833654

R-Square 0.393438145245

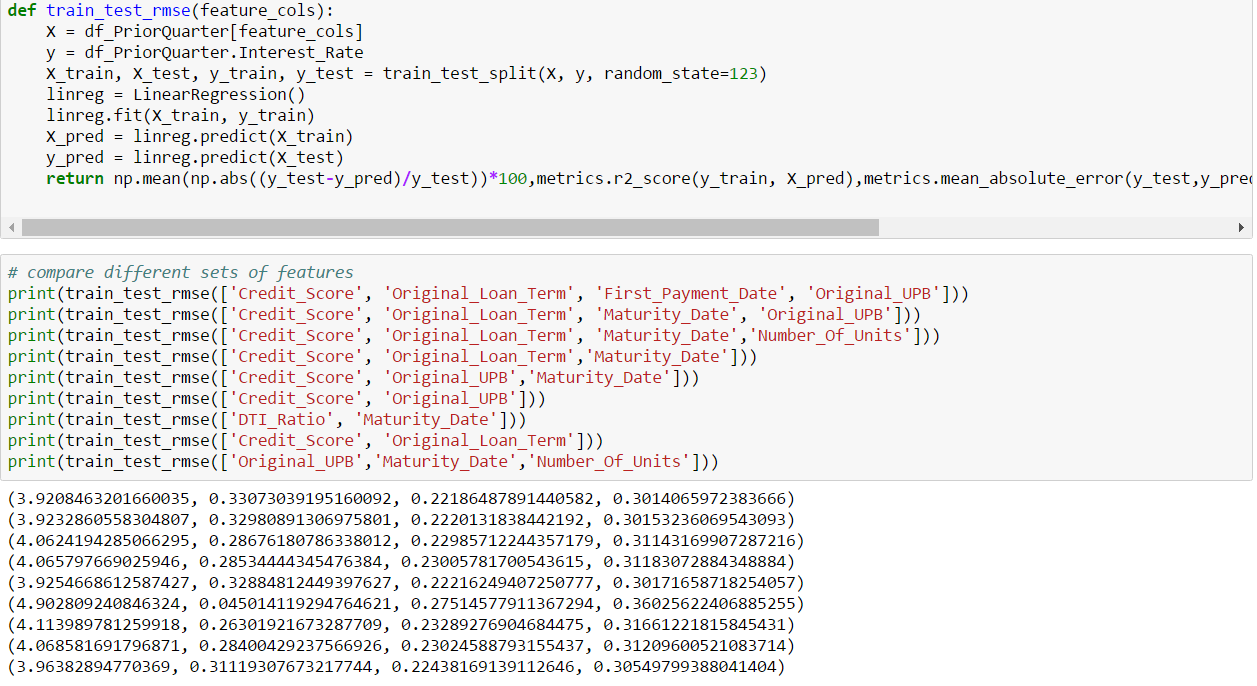
Model Prediction and evaluation metrics:

(array([ 5.74995348, 5.62244285, 5.7265196 , ..., 6.00641581,

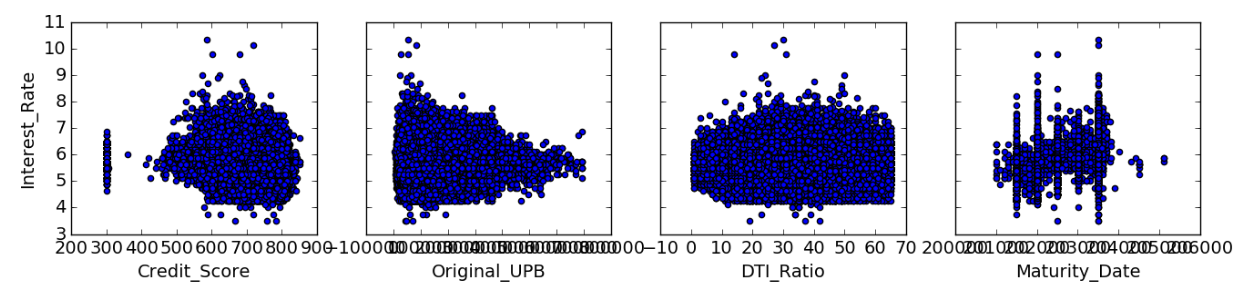
6.07177263, 6.06140455]),

|  |  |  |  |
| --- | --- | --- | --- |
|  | RMSE | MAE | MAPE |
| TEST SET | 0.32003543771201615 | 0.24600933331878819 | 4.206309025374067 |

Performed **train-test-split** to compare different sets of features in the training dataset.



Visualizing the data





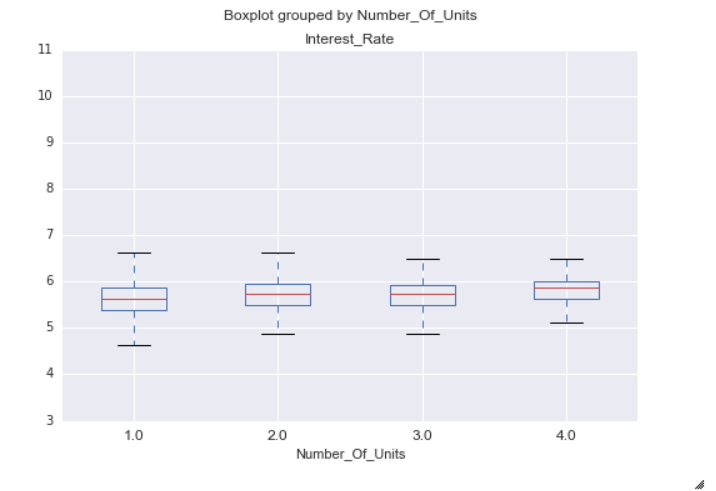


Figure 1

## Feature Selection

We have used Recursive feature selection in Python to obtain variables based on their ranking:

And the result is:

RFE(estimator=LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=1, normalize=False),

n\_features\_to\_select=3, step=1, verbose=0)

We got following variables with their ranks:

[(1, 'First\_Time\_Homebuyer\_Flag'),(1, 'Number\_Of\_Units'), (2, 'MI'), (3, 'Channel'), (4, 'First\_Payment\_Date'), (5, 'PPM'), (6, 'Property\_Type'), (7, 'Loan\_Purpose'), (8, 'Original\_Loan\_Term'), (9, 'Service\_Name'), (10, 'Maturity\_Date'), (11, 'Seller\_Name'), (12, 'CLTV'), (13, 'Property\_State'), (14, 'Number\_Of\_Borrowers'), (15, 'Product\_Type'), (16, 'MSA'), (17, 'Loan\_Sequence\_Nmber'), (18, 'Credit\_Score'), (19, 'LTV'), (20, 'Occupancy\_Status'), (21, 'DTI\_Ratio'), (22, 'Postal\_Code'), (23, 'Original\_UPB')]

## Ordinary Least Squares Assumptions

OLS measures the accuracy of a linear regression model.

OLS is built on assumptions which, if held, indicate the model may be the correct lens through which to interpret our data. If the assumptions don't hold, our model's conclusions lose their validity.

We build our model on testing various variables and following features gives suitable results:

Interest\_Rate~ Credit\_Score+Maturity\_Date+MI+Original\_Loan\_Term+Occupancy\_Status+Original\_UPB.

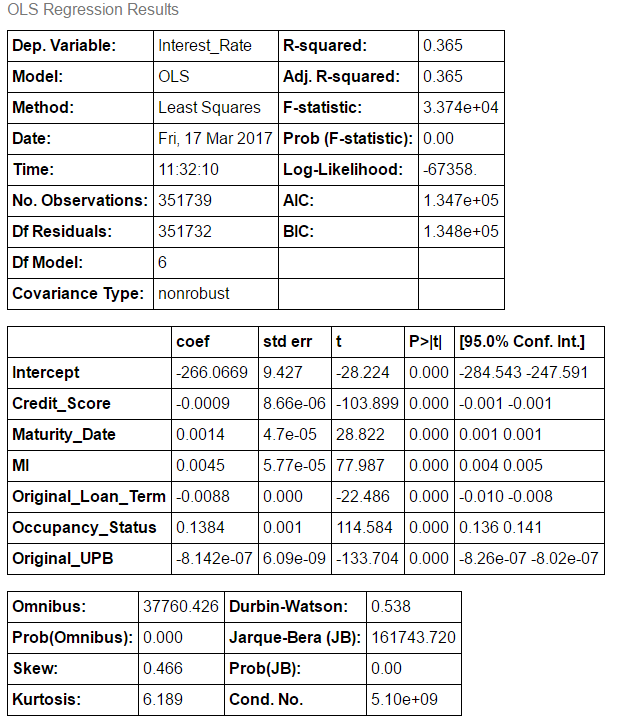
ols\_model = ols("Interest\_Rate ~Credit\_Score+Maturity\_Date+MI+Original\_Loan\_Term+Occupancy\_Status+Original\_UPB", data=df\_PriorQuarter).fit()

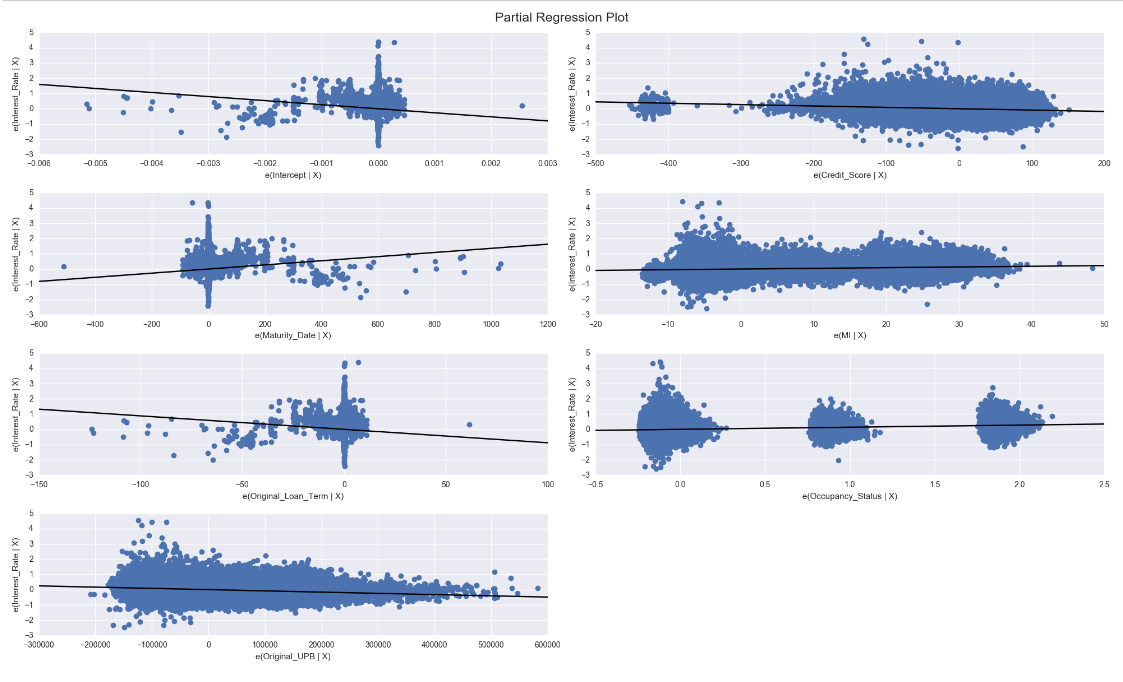
From the results below we can see **Adj. R-squared** is 36%.

The **regression coefficient (coef)** represents the change in the dependent variable resulting from a one unit change in the predictor variable, all other variables being held constant.

The **standard error** measures the accuracy of the variable’s coefficient by estimating the variation of the coefficient if the same test were run on a different sample of our population. Our standard error is low in case of variables and therefore appears accurate.

The **confidence interval** is a range within which our coefficient is likely to fall. The coefficients we can see will be within our confidence interval, [-284.543 -247.591]

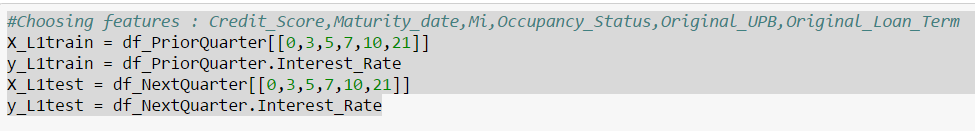


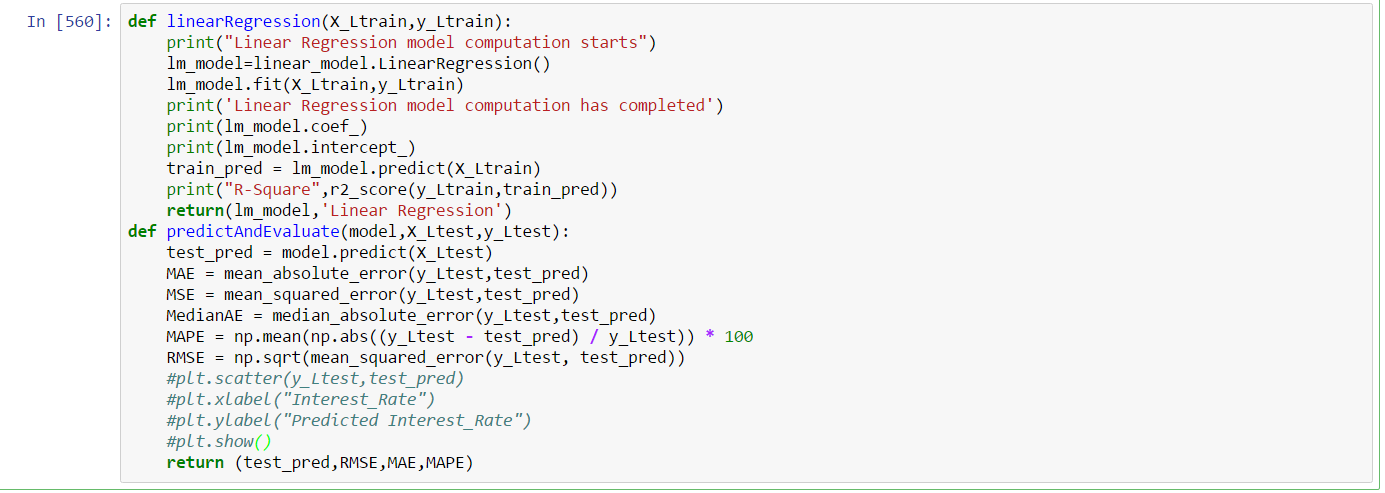


**Now, based on RFE and ordinary least squares assumptions,**

**Build Regression Model with Variables:**

**Credit\_Score, First\_Time\_Homebuyer\_Flag, Maturity\_Date, MI, Number\_Of\_Units, Original\_UPB**





The regression model is first trained on a cleaned dataset, then run the model on complete dataset. On fitting the regression model, testing against the complete test dataset measures of predictive accuracy are:

|  |  |  |  |
| --- | --- | --- | --- |
|  | RMSE | MAE | MAPE |
| TEST SET | 0.329327 | 0.251134 | 4.281914 |

## Random Forest

First, build the model for all the variables and obtain a list of important features:

feature importance

14 Product\_Type 0.000000

13 PPM 0.001382

6 Number\_Of\_Units 0.002619

21 Number\_Of\_Borrowers 0.008146

2 First\_Time\_Homebuyer\_Flag 0.010309

16 Property\_Type 0.013789

19 Loan\_Purpose 0.015851

1 First\_Payment\_Date 0.018275

5 MI 0.019655

12 Channel 0.020137

23 Service\_Name 0.028803

7 Occupancy\_Status 0.033177

22 Seller\_Name 0.036715

15 Property\_State 0.036883

4 MSA 0.038807

8 CLTV 0.039839

11 LTV 0.043921

9 DTI\_Ratio 0.044808

17 Postal\_Code 0.052212

0 Credit\_Score 0.065206

10 Original\_UPB 0.076680

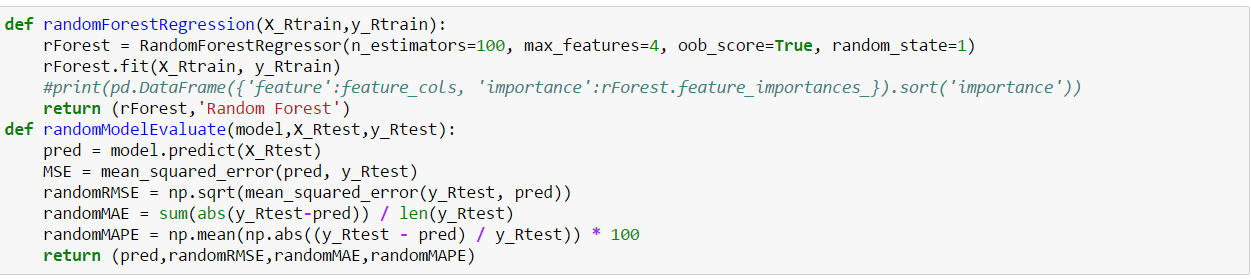
20 Original\_Loan\_Term 0.105457

3 Maturity\_Date 0.110886

18 Loan\_Sequence\_Nmber 0.176444

Them we build the model on the basis of their importance:

So, we selected, Credit\_Score+ Original\_UPB +Original\_Loan\_Term+Maturity\_Date



The predictive accuracy measures are:

'Random Forest',

0.33478593530307088,

0.25654889279095255,

4.455899688112888,

RandomForestRegressor(bootstrap=True, criterion='mse', max\_depth=None,

max\_features=4, max\_leaf\_nodes=None, min\_impurity\_split=1e-07,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=100, n\_jobs=1,

oob\_score=True, random\_state=1, verbose=0, warm\_start=False)

## KNN Regression

We are using k-nearest neighbors implementation in scikit-learn. We will be using the regressor.

Sklearn performs the normalization and distance finding automatically,

The variables we have selected after doing feature selection is:

'Credit\_Score','Original\_Loan\_Term','Maturity\_Date','Original\_UPB','MI','Occupancy\_Status'



The predictive accuracy measures are:

'Knn Regression',

0.36305893676002154,

0.28146139017554067,

4.8163083099684156,

KNeighborsRegressor(algorithm='auto', leaf\_size=30, metric='minkowski',

metric\_params=None, n\_jobs=1, n\_neighbors=3, p=2,

weights='uniform')

RMSE : 0.36305893676002154

MAE: 28146139017554067

MAPE: 4.8163083099684156

## Neural Networks

We have used h2o library to perform neural networks regression.

The model is trained on complete dataset.

#Defining columns for X and Y

y = "Interest\_Rate"

x = data.names

x.remove(y)

m = h2o.estimators.deeplearning.H2ODeepLearningEstimator()

m.train(x, y, trainingData)

The model predictive accuracy are :

p = m.predict(testH2OData)

m.model\_performance(testH2OData)

\*\* Reported on test data. \*\*

MSE: 0.08526408709350661

RMSE: 0.2920001491326787

MAE: 0.22438189817548285

RMSLE: 0.0429421339230831

Mean Residual Deviance: 0.08526408709350661

Model Details

=============

H2ODeepLearningEstimator : Deep Learning

Model Key: DeepLearning\_model\_python\_1489710420009\_1

ModelMetricsRegression: deeplearning

\*\* Reported on train data. \*\*

MSE: 0.05877940213965719

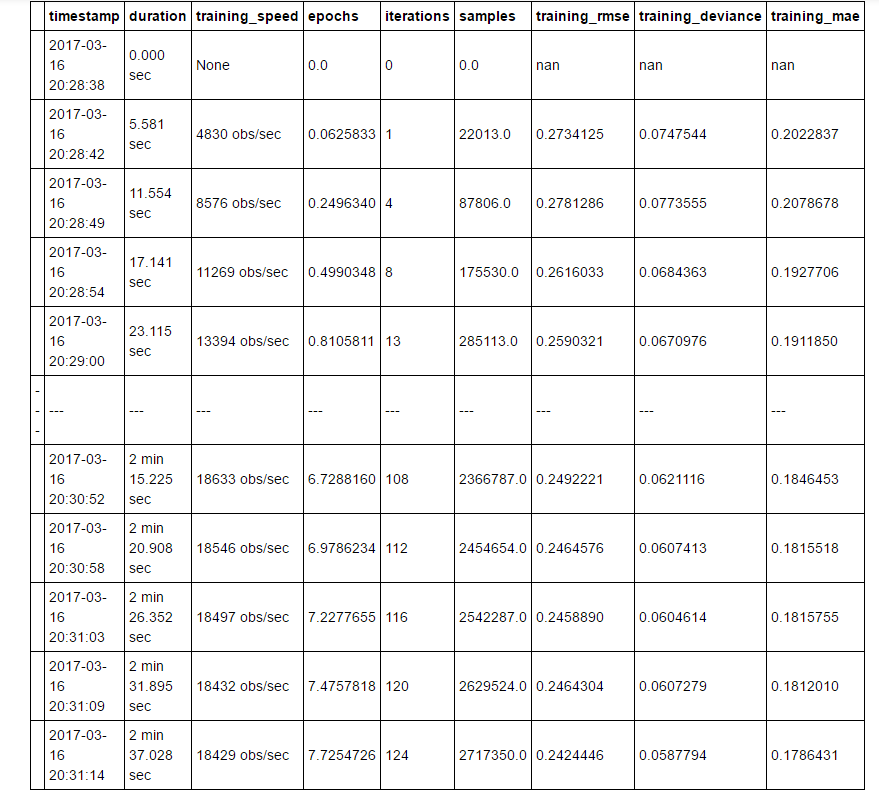
RMSE: 0.24244463726726806

MAE: 0.17864308111486663

RMSLE: 0.03615252680630111

Mean Residual Deviance: 0.05877940213965719

Scoring History:



## Best Model

Comparing the MAPE from the all the regression model we generate, its concluded the multiple linear regression gives good results:

We have ranked the model in column Model\_Rank, as we can see Linear Regression is ranked as 1, as with low MAPE.

The metrics below are saved in file : **modelMetrics.csv** which also includes the ranking.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Algorithm** | **RMSE** | **MAE** | **MAPE** | **Model** | **Model\_Rank** |
| 0 | Linear Regression | 0.324312 | 0.249246 | 4.255456 | LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=1, normalize=False) | 1 |
| 1 | Random Forest | 0.334786 | 0.256549 | 4.4559 | RandomForestRegressor(bootstrap=True, criterion='mse', max\_depth=None,   max\_features=4, max\_leaf\_nodes=None, min\_impurity\_split=1e-07,   min\_samples\_leaf=1, min\_samples\_split=2,   min\_weight\_fraction\_leaf=0.0, n\_estimators=100, n\_jobs=1,   oob\_score=True, random\_state=1, verbose=0, warm\_start=False) | 2 |
| 2 | Knn Regression | 0.363059 | 0.281461 | 4.816308 | KNeighborsRegressor(algorithm='auto', leaf\_size=30, metric='minkowski',   metric\_params=None, n\_jobs=1, n\_neighbors=3, p=2,   weights='uniform') | 3 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

Neural Network is done using h2o library, we give some different observations too.

We have considered its RMSE, MSE, and MAE. The metrics are saved in separate CSV file:

**neuralNetworkMetrics.csv**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | MSE | RMSE | MAE | Model |
| Neural Network | 0.283171 | 0. 0.080186 | 0.213005 | H2omodel |

We will choose Mulltple **Linear Regression** as the best model, Based on the performance, flexibility, good metrics and selection of various features.

## Financial Crisis

We will be doing what-if analysis of our algorithm used in various situations.

In Financial crisis 2007, we will be running our algorithm for 4 rolling quarters.

Training dataset: Q12007, Q22007, Q32007, Q42007

Testing dataset: Q22007, Q32007, Q42007, Q12008

We have used our multiple linear algorithm with best selected features:

Credit\_Score+Maturity\_Date+MI+ Occupany\_status +Original\_UPB +Original\_Loan\_Term

The predictive accuracy results are:

|  |  |  |  |
| --- | --- | --- | --- |
| Quarter\_Period | RMSE | MAE | MAPE |
| Q12007-Q22007 | 0.366396 | 0.278060 | 4.355950 |
| Q22007-Q32007 | 0.489688 | 0.404815 | 5.934174 |
| Q32007-Q42007 | 0.401141 | 0.313130 | 4.990402 |
| Q42007-Q12008 | 0.488185 | 0.402792 | 7.106593 |

We can see the MAPE for Q12007-Q22007 as 4.35% which is low, for next quarter there is 1% increase

For Q32007-Q42007, the value is 4.99%.

For Q42007-Q12008, there is an increasing in the percentage,

If we look at the model coefficient and intercept for Q42007-Q12008:

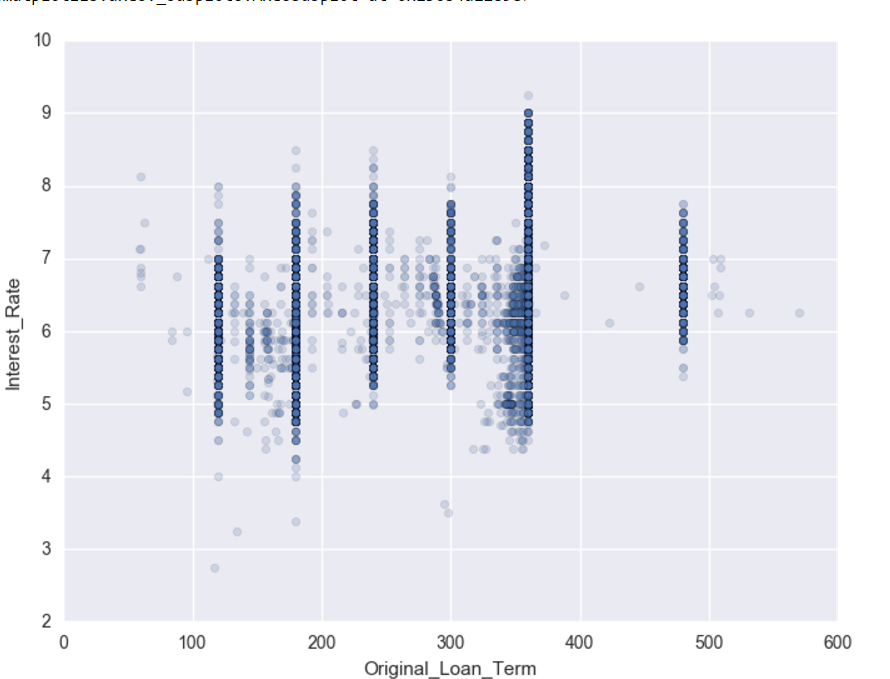
[ -1.55392258e-03 -2.32118353e-03 6.89047407e-03 1.99123471e-01

-6.19591361e-07 2.11923856e-02]

472.849990046

R-Square 0.326695653816

Test results show that the model can estimate the samples, and help control the samples. The low percentage error of discriminate analysis of the model shows that the model has a good reputation for having the early-warning effect.



In above graph, just analyzing how Interest\_Rate varies with Original\_Loan\_Term in 2007.

## Year -2009

We run the algorithm for 2009 quarters, and below are the metrics results:

|  |  |  |  |
| --- | --- | --- | --- |
| **Quarter-Period** | **RMSE** | **MAE** | **MAPE** |
| Q12009-Q22009 | 1.011569 | 0.965222 | 20.343296 |
| Q22009-Q32009 | 0.783248 | 0.712936 | 14.424321 |
| Q32009-Q42009 | 0.852148 | 0.807421 | 16.794102 |

There is increase in MAPE, if we compare to our previous year quarter predictions. The market is assumed to be highly unstable.

We analyse the data set and saw some variations.

Summary metrics of Interest\_Rate for 2005 and 2009

Q12009

count 587170.000000

mean 4.939940

std 0.367988

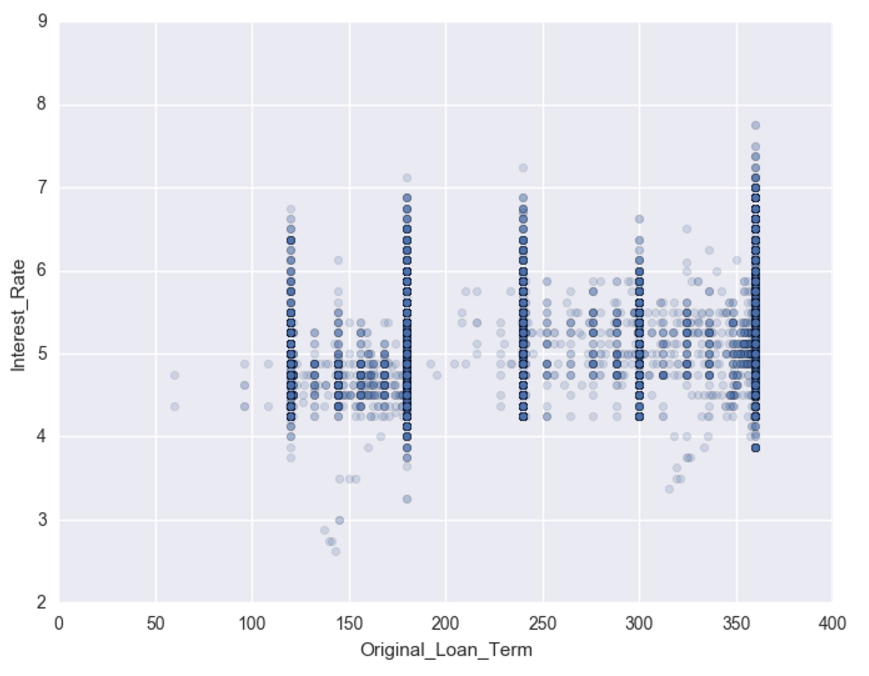
min 2.750000

25% 4.750000

50% 4.875000

75% 5.125000

max 7.875000



## Economic Boom (1999,2013)

Year 1999 and 2013 were the year of economic booms, we will try our algorithm on both these years and see how good our model is to make the predictions.

**Year 1999**

Run the algorithm for Year 1999, the predicted results are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Quarter-Period** | **RMSE** | **MAE** | **MAPE** |
| Q11999-Q21999 | 0.766827 | 0.670401 | 8.867243 |
| Q21999-Q31999 | 0.557627 | 0.465656 | 6.004151 |
| Q31999-Q41999 | 1.304490 | 1.255114 | 17.816717 |

**Year 2013**

|  |  |  |  |
| --- | --- | --- | --- |
| **Quarter-Period** | **RMSE** | **MAE** | **MAPE** |
| Q11999-Q21999 | 0.287583 | 0.215391 | 6.062558 |
| Q21999-Q31999 | 0.698267 | 0.603381 | 13.886196 |
| Q31999-Q41999 | 0.336275 | 0.267301 | 6.251445 |

Looking at the predictions, RMSE, MAE and MAPE we can see model is working fine when tested against the these datasets. MAPE is low. For Q21999-Q31999 , there is increase, but still the the error is not that high,

# GitHub Links:

Github Repo: https://github.com/Yamini-S/ADS

# Docker hub links:

We have registered our docker image on docker hub and the links are:

**For Part1 and Part2:Prediction**

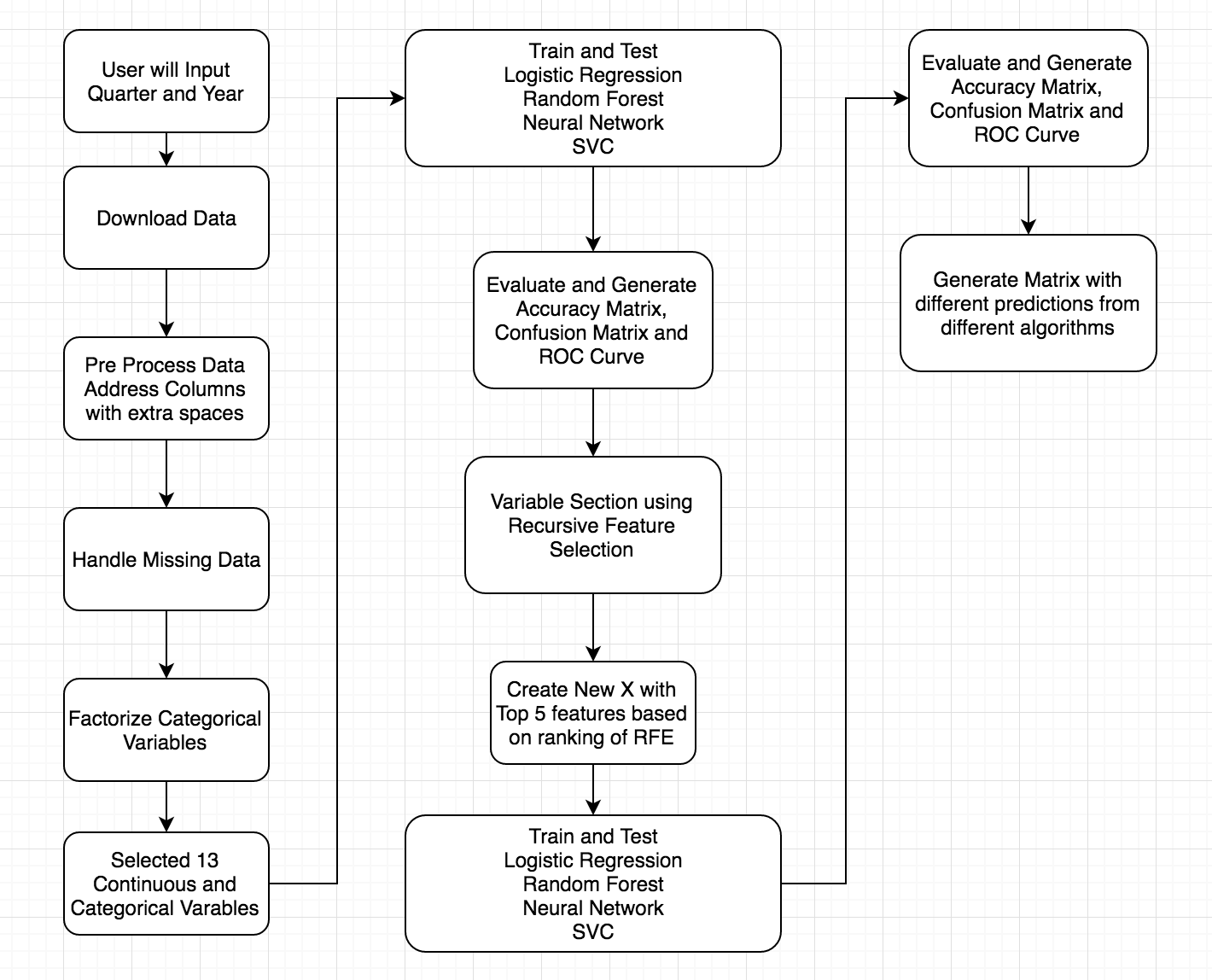
**Repo: yaminis/midterm**

**Link:** <https://hub.docker.com/r/yaminis/midterm/>

This repo contains two tags: summarization and prediction

# Classification:

## Script 1 Approach:



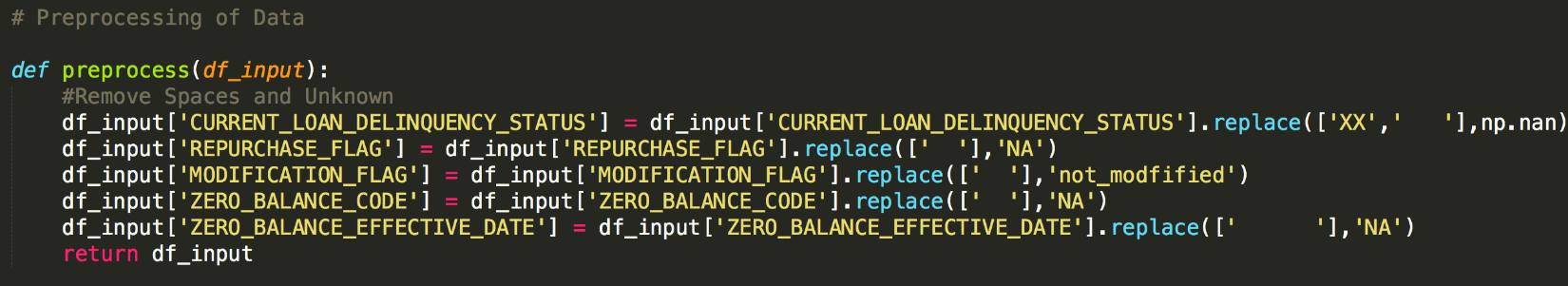
Note: Python will throw few warning, please ignore them.

Steps Involved:

1. Preprocess Data
2. Handle Missing Data
3. Factorize Categorical Variables
4. Run Classification algorithms on all Features
5. Recursive Feature Selection
6. Run Classification algorithms on Top 5 ranked Features

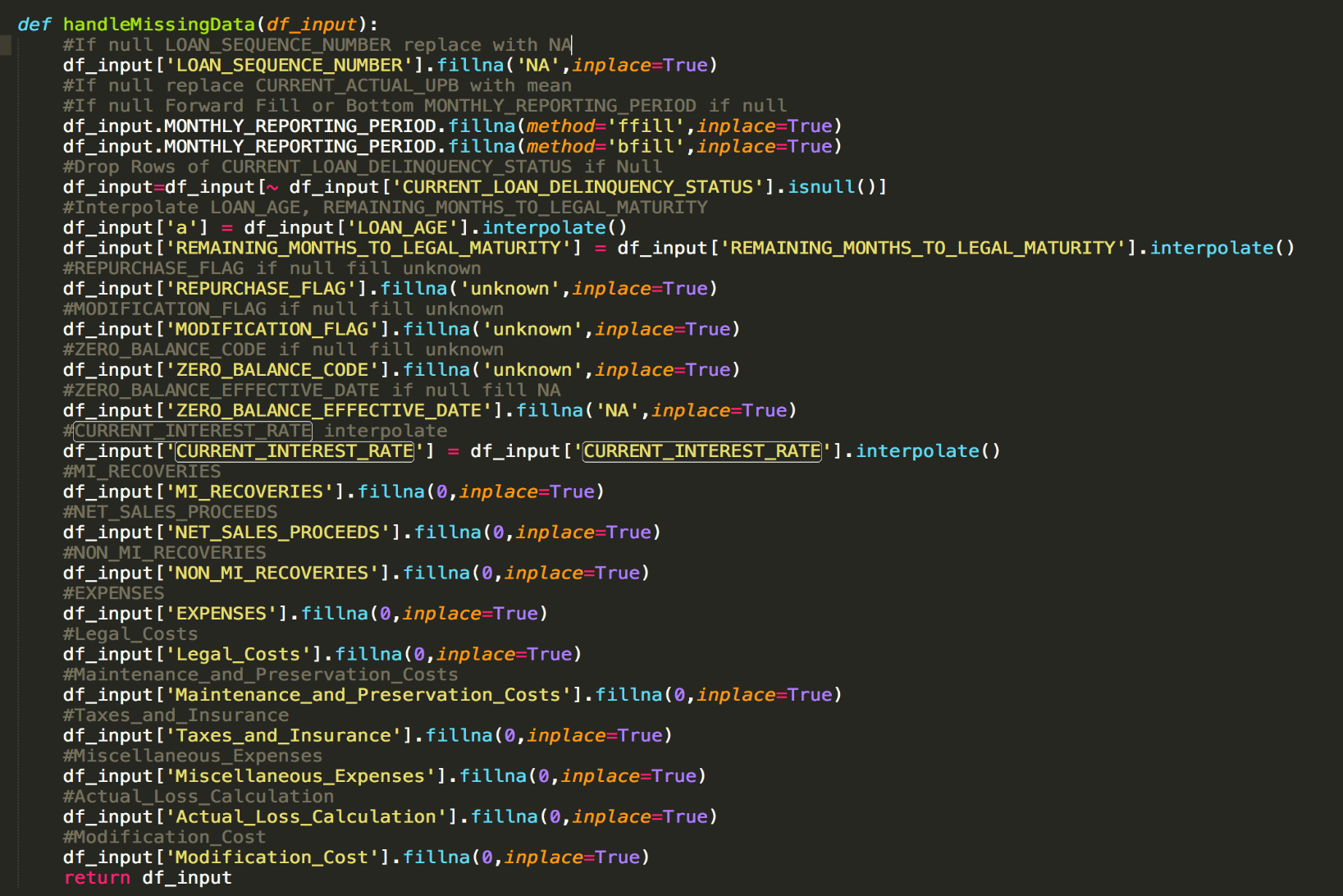
## Preprocess Data:

As part of processing, many columns in data set have empty spaces as per the Documentation. If there are any empty spaces these needs to be addressed. Below code snippet shows that columns that were preprocessed:



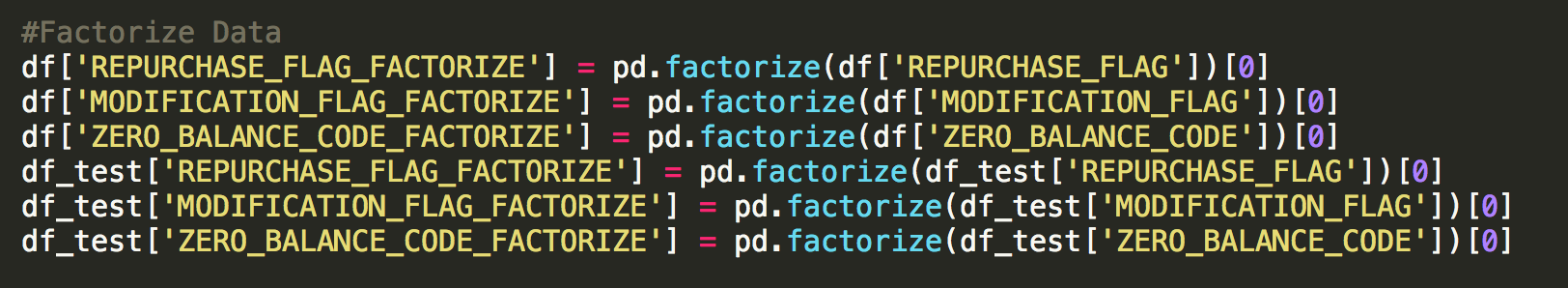
## Handling Missing Data:

* Interpolating Interest Rate, Loan Age, Remaining Months to Legal Maturity.
* For categorical, all the null values are filled with either NA or Unknown
* For time series data, We have used bottom fill and top fill.



## Handle Categorical Variables:

All the Categorical variables are factorized to integers and are being used as features in training and testing the model:



## Run Classification algorithms on 13 Features:

#### Logistic Regression Output: (Q1 2006 for Training and Q2 2006 for Testing)

#### 

#### Random Forest Output:

#### 

#### Neural Net Output:

#### 

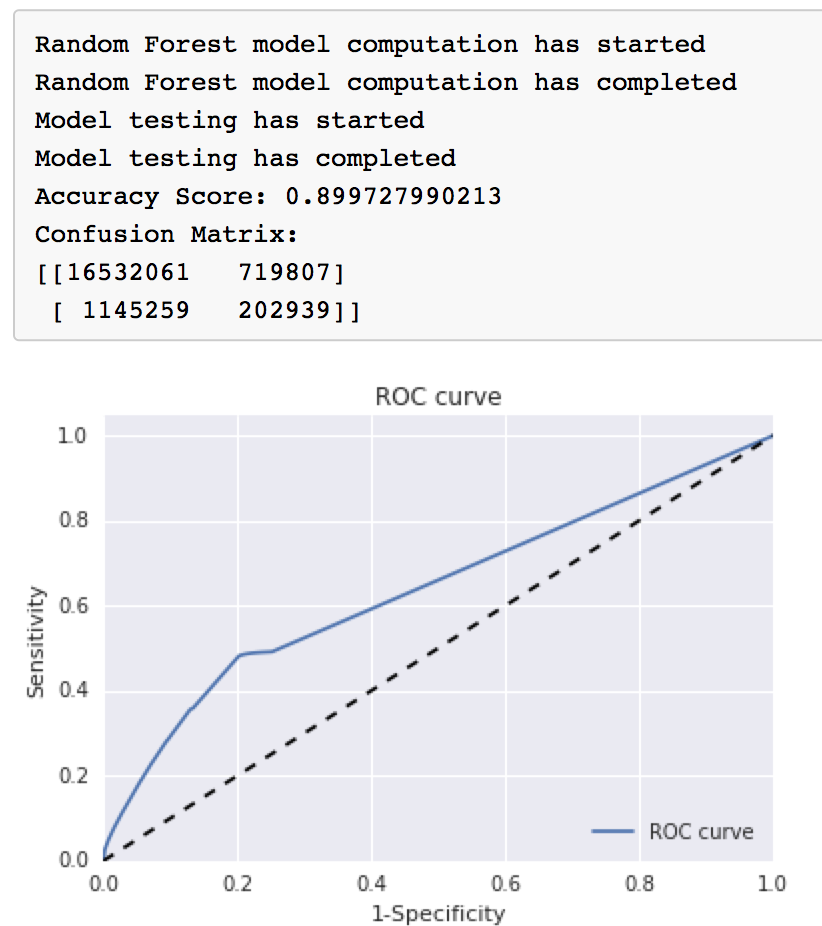
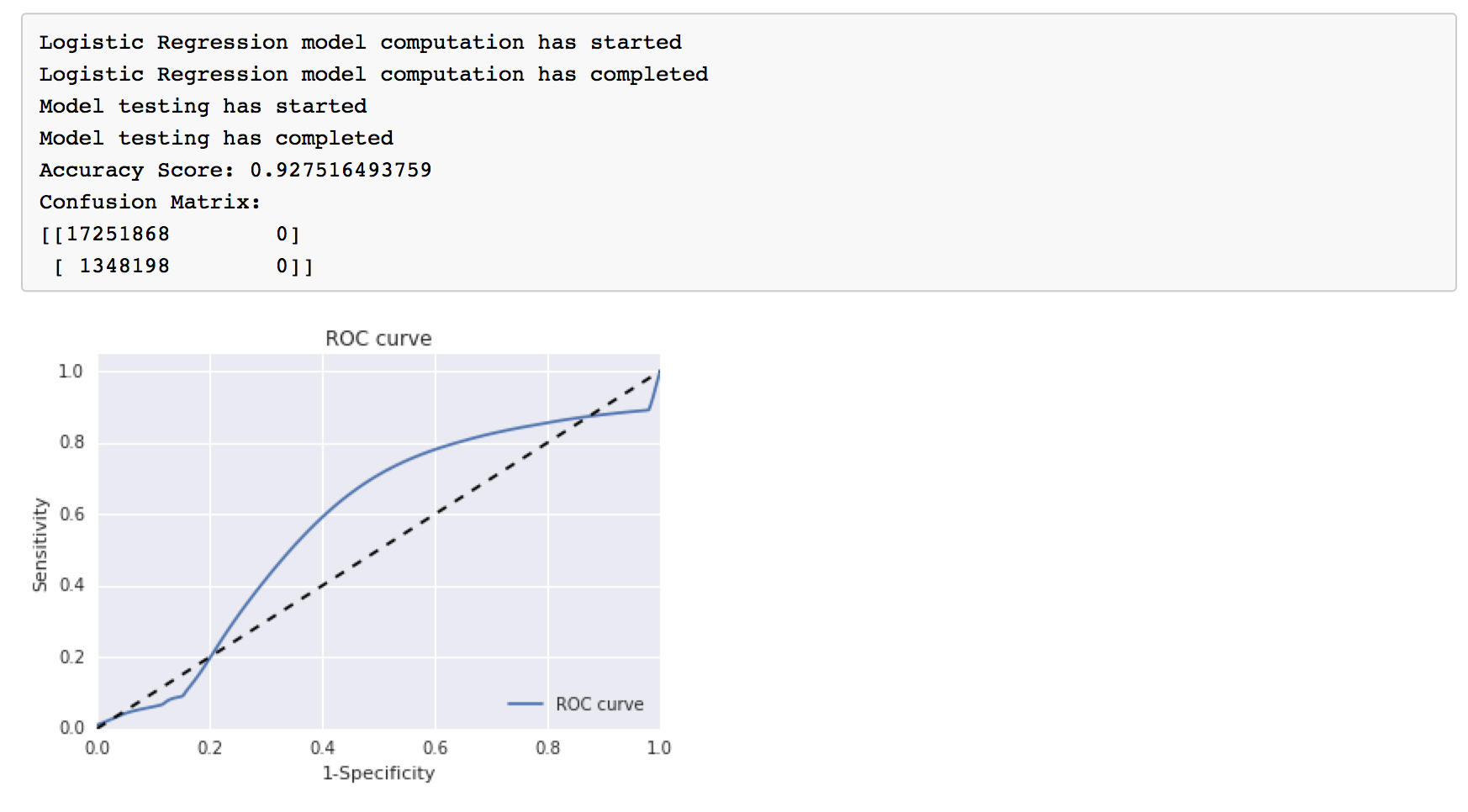
### Recursive Feature Elimination:

We have used Recursive Feature Elimination and tried to pull different number of features and tried to compute algorithms on those features. Finally we concluded with 5 Features and Feature elimination step size of 2.

4 Features selected Rankings: (Q1 2006)

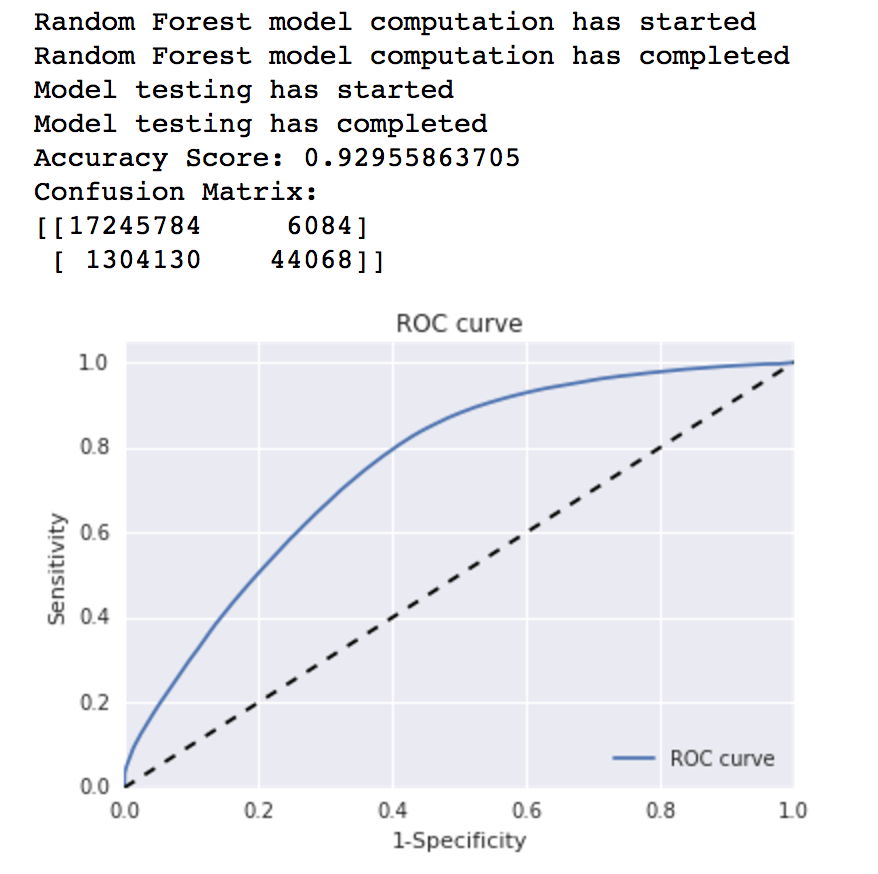
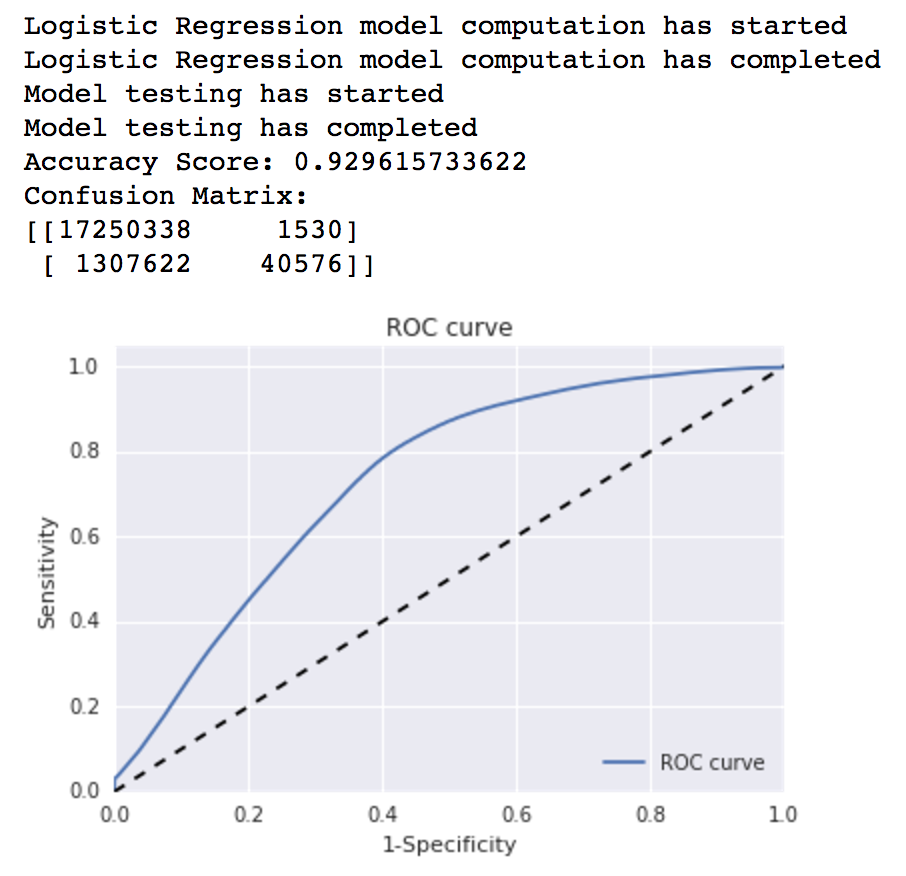
[(1, 'CURRENT\_ACTUAL\_UPB'), (1, 'CURRENT\_INTEREST\_RATE'), (1, 'LOAN\_AGE'), (1, 'REMAINING\_MONTHS\_TO\_LEGAL\_MATURITY'), (2, 'EXPENSES'), (3, 'CURRENT\_DEFERRED\_UPB'), (3, 'MODIFICATION\_FLAG\_FACTORIZE'), (4, 'Actual\_Loss\_Calculation'), (4, 'ZERO\_BALANCE\_CODE\_FACTORIZE'), (5, 'Modification\_Cost'), (5, 'REPURCHASE\_FLAG\_FACTORIZE'), (6, 'MI\_RECOVERIES'), (6, 'NON\_MI\_RECOVERIES')]

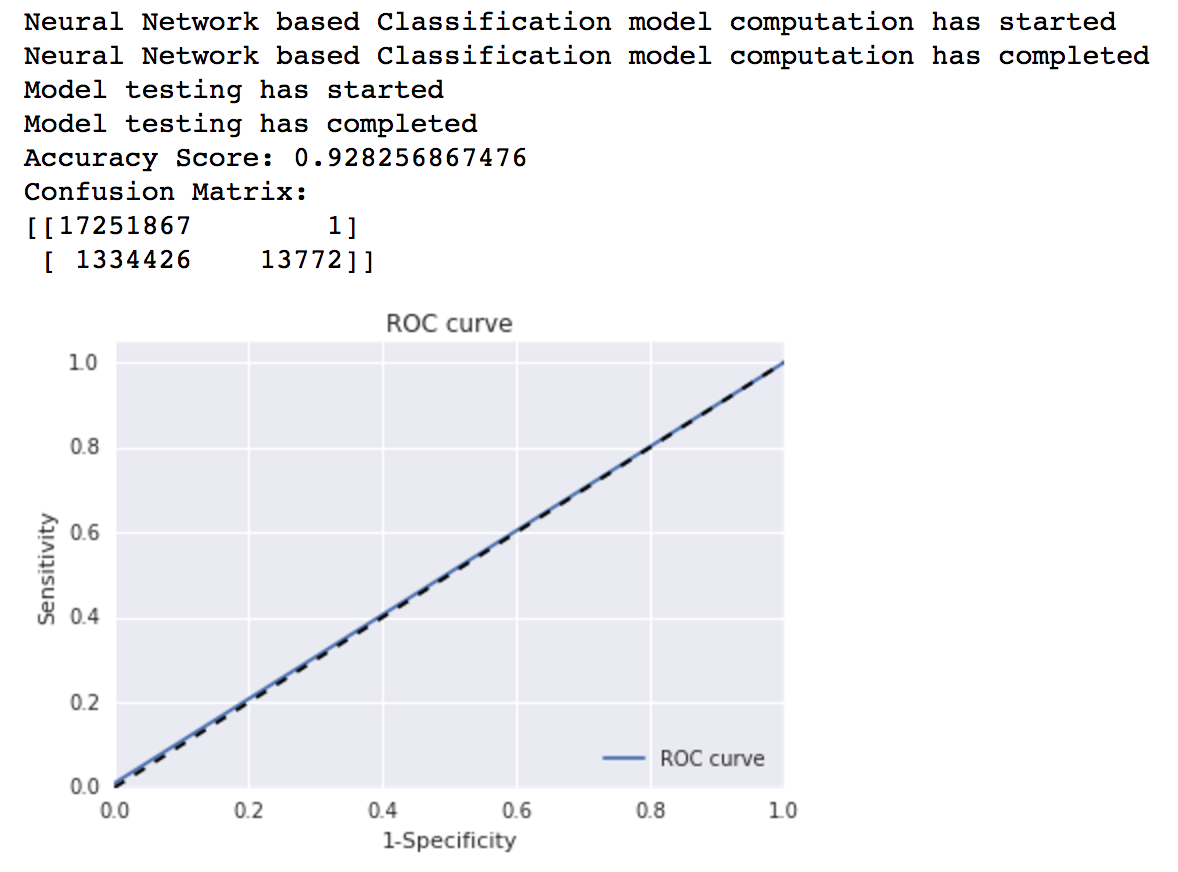
Empty DataFrame



5 Features Selected Ranking and Outcomes: (Q1 2006)

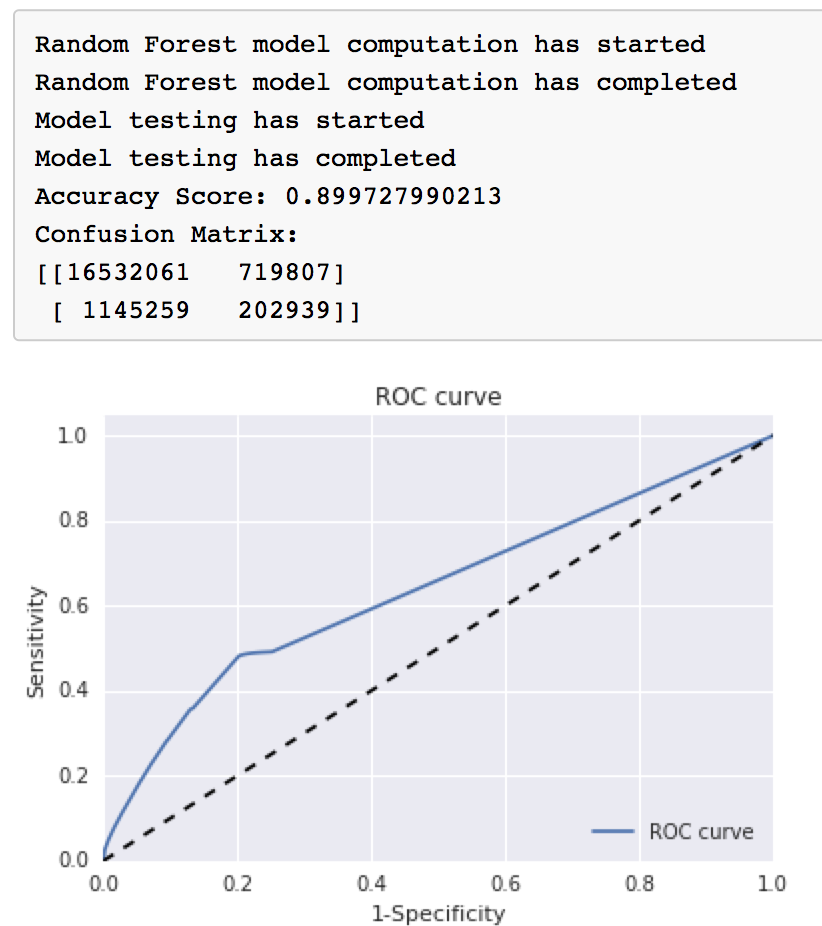
[(1, 'CURRENT\_INTEREST\_RATE'), (1, 'LOAN\_AGE'), (1, 'MODIFICATION\_FLAG\_FACTORIZE'), (1, 'REPURCHASE\_FLAG\_FACTORIZE'), (1, 'ZERO\_BALANCE\_CODE\_FACTORIZE'), (2, 'EXPENSES'), (2, 'REMAINING\_MONTHS\_TO\_LEGAL\_MATURITY'), (3, 'MI\_RECOVERIES'), (3, 'NON\_MI\_RECOVERIES'), (4, 'Actual\_Loss\_Calculation'), (4, 'CURRENT\_DEFERRED\_UPB'), (5, 'CURRENT\_ACTUAL\_UPB'), (5, 'Modification\_Cost')]





3 Features Selected: (Q1 2006)

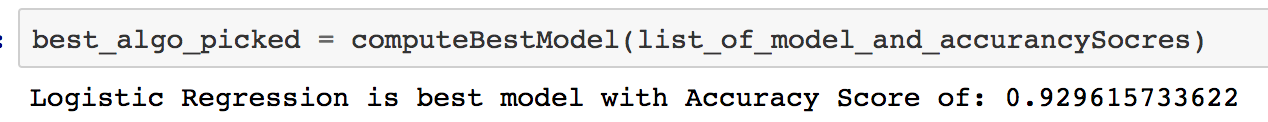
[(1, 'MODIFICATION\_FLAG\_FACTORIZE'), (1, 'REPURCHASE\_FLAG\_FACTORIZE'), (1, 'ZERO\_BALANCE\_CODE\_FACTORIZE'), (2, 'CURRENT\_INTEREST\_RATE'), (2, 'LOAN\_AGE'), (3, 'EXPENSES'), (3, 'REMAINING\_MONTHS\_TO\_LEGAL\_MATURITY'), (4, 'MI\_RECOVERIES'), (4, 'NON\_MI\_RECOVERIES'), (5, 'Actual\_Loss\_Calculation'), (5, 'CURRENT\_DEFERRED\_UPB'), (6, 'CURRENT\_ACTUAL\_UPB'), (6, 'Modification\_Cost')]



5 Features with 2006 Q1 and 2005 Q1 performed well so we went ahead with these 5 features;

### Selecting Best Algorithm:

We are select best algorithm based on Accuracy Score of Data. Below is the code snippet for deciding the best algorithm and this will return the mode which could be used in Script 2 for testing all quarters data.



## Script 2 (To Compute Matrix for all the Quarters with best model generated out of Script 1):

Matrix has been computed form Q2 1999 to Q1 2016, the output Jupyter notebook is attached with this documentation.

## Docker Image:

**CAUTION:** To run this image, you need a high memory machine. I have tested in machine with 16 GB RAM and it still hangs. I have used IBM Data Science Experience to run both the scripts in Jupyter Notebook.

Two Images are computed for classification, one for Script 1 which asks for Year and Month when ran.

* docker pull melavoyrajap/midterm\_assignment:classification
* docker run –it melavoyrajap/midterm\_assignment:classification

Docker Image for Script 2:

This doesn’t take any input, it’s default input is Q1 1999 and it generates matrix as requested in assignment. By testing against rest of Q21999 – Q12016 data.

* docker pull melavoyrajap/midterm\_assignment:classification\_part2
* docker run –it melavoyrajap/midterm\_assignment:classification\_part2