**Report Document**

**Data extraction Part 1 and Part 2:**

1. config.R

* It has parameters to take values for Username and password for the Freddie Mac data
* Or it takes session cookie value after a manual login.
* The session cookie should be captured after accepting the Terms and Conditions, using the browser's development tools.

1. MidTerm\_CaseStudy\_Data.R

* It refers to config.R file and based on input either a session cookie or username , password values, extracts(download) data from

<https://freddiemac.embs.com/FLoan/Data/download.php> (both sample (part1) and historical(part2) data)

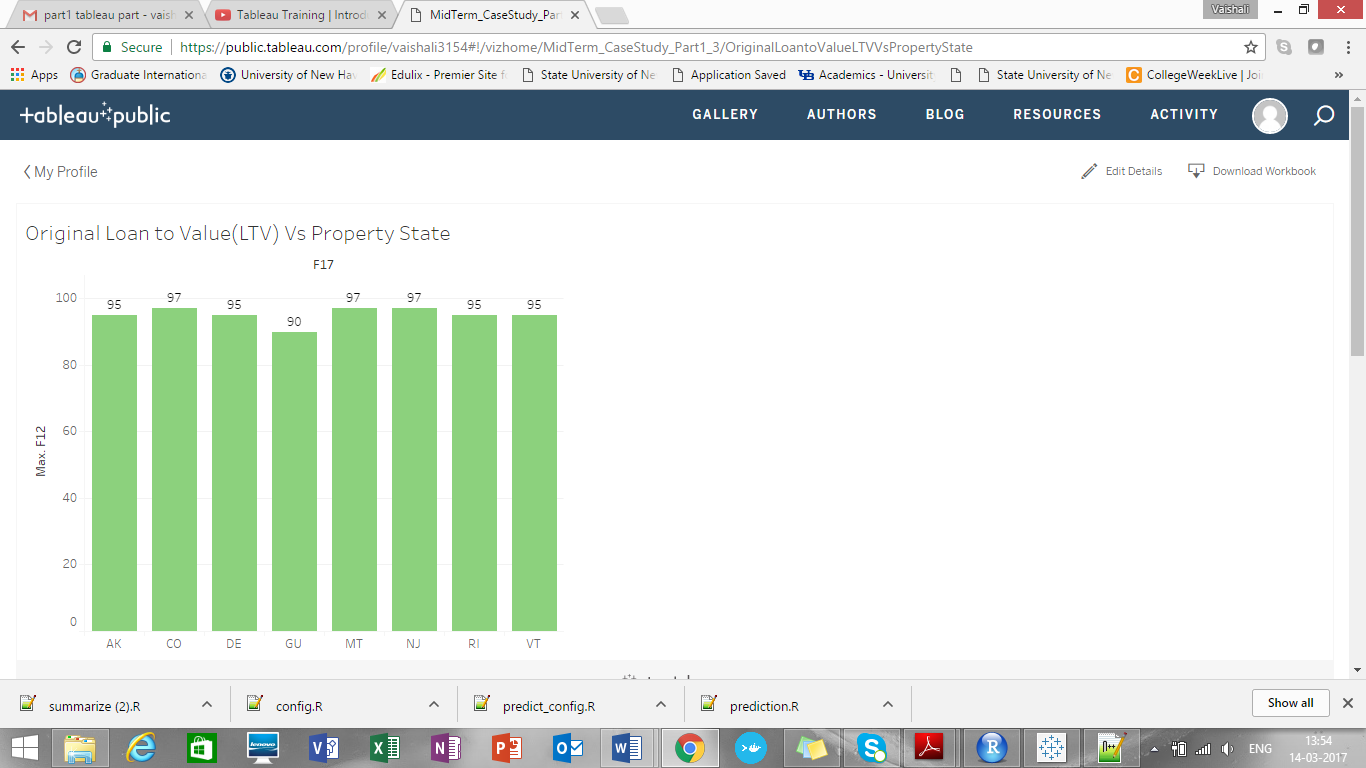
* Unzips data

**Part1: Exploratory analysis**

1. Tableau dashboards using 2005 origination data:

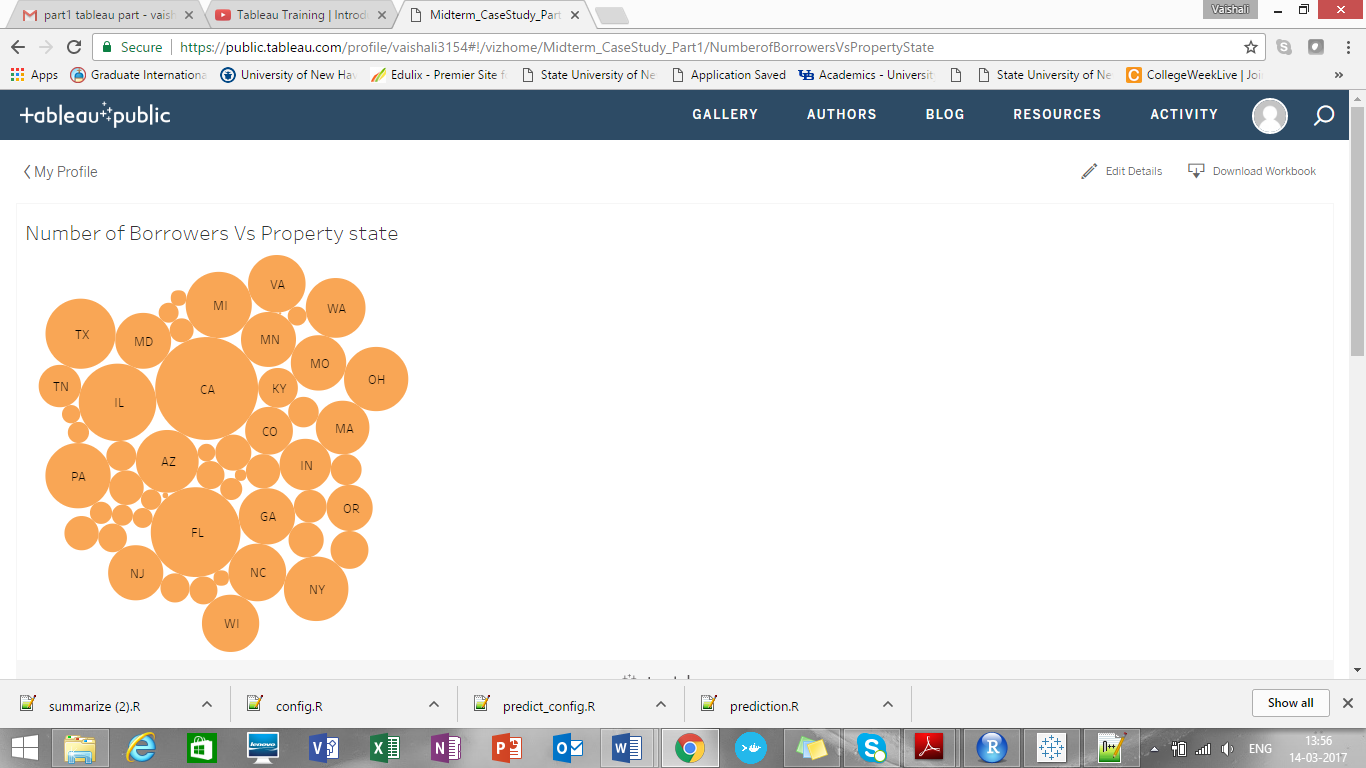
* <https://public.tableau.com/profile/vaishali3154#!/>

**Analysis 1:** Original Loan to Value(LTV) Vs Property State



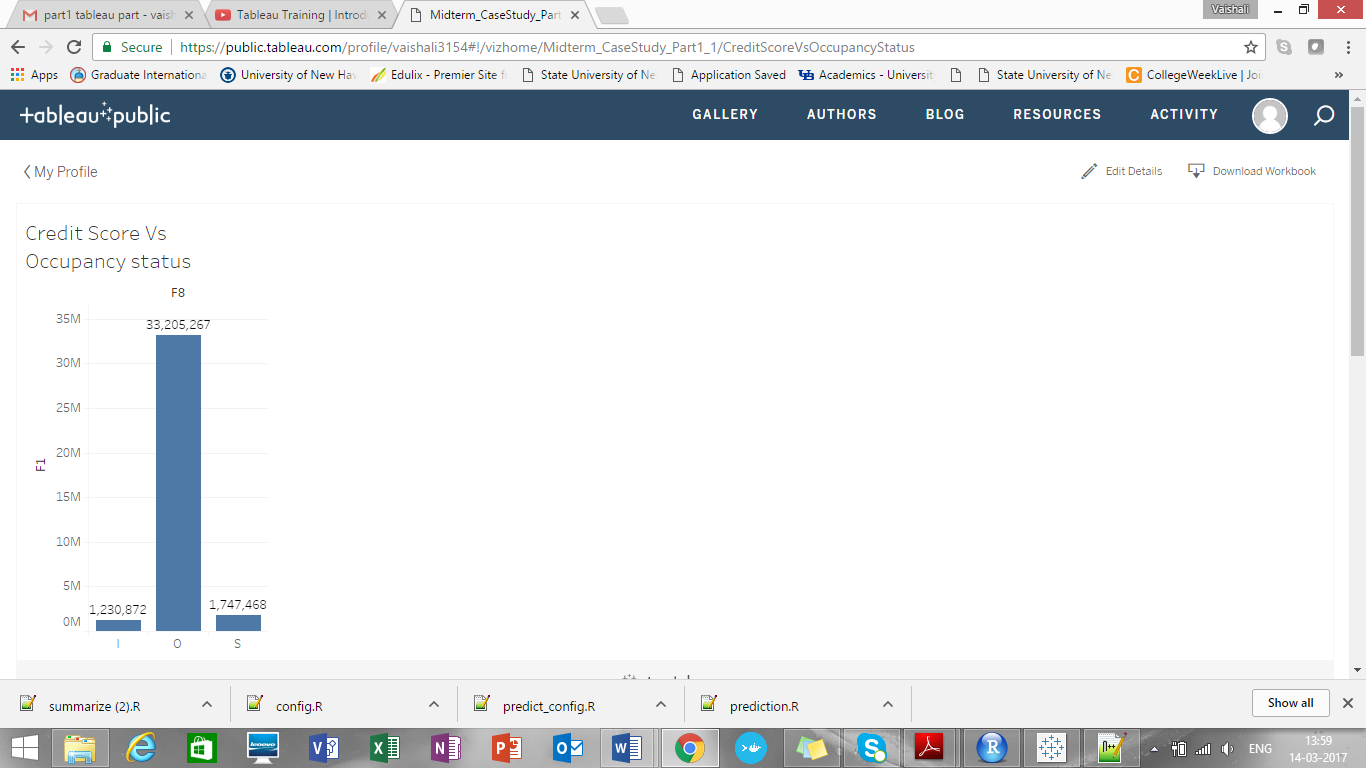
It shows states which has LTV ratio greater than or equal to 90%

**Analysis 2:** Number of Borrowers Vs Property state

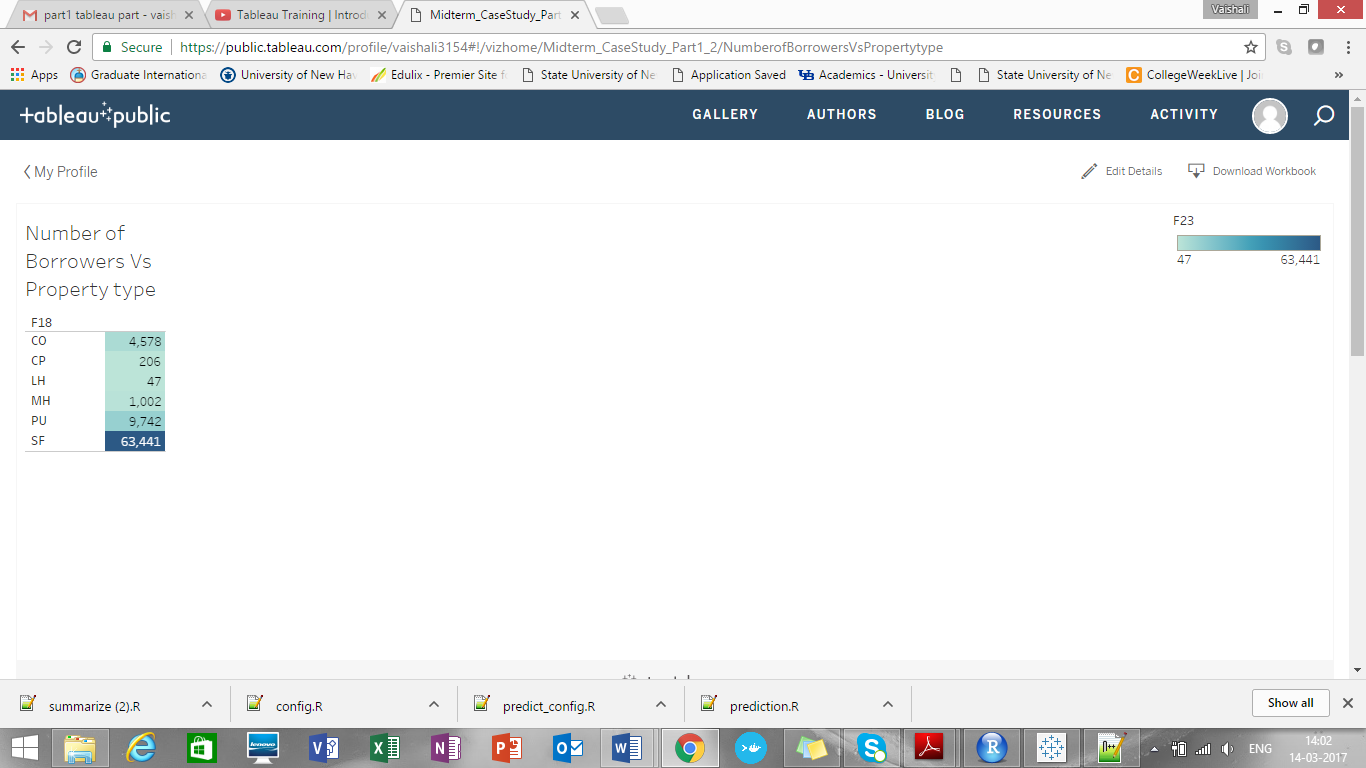


It shows that CA has maximum number of borrowers amongst all property states.

**Analysis 3:** Credit Score Vs Occupancy status

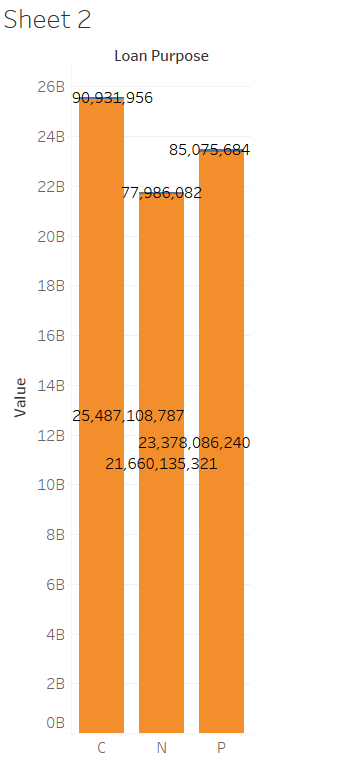


Owners have maximum credit score than investment property or second home

**Analysis 4:** Number of Borrowers Vs Property type 

It shows that 1-4 Fee Simple property type contains maximum borrowers

**Analysis 5:** From this, we can infer that maximum people replace their existing mortgage with a new home loan



**Part2: Prediction**

1. predict\_config.R

* It contains training\_quarter variable, which contains value empty or some value like Q22005

1. prediction.R

* In this file, it refers to predict\_config.R file to find value for training quarter, if it is not present it takes default value as Q12005.
* The test data will be considered as quarter next to selected training quarter
* Then it unzips data and refers origination file for prediction purpose

It has 26 columns, out of which we have removed below columns for training\_df data frame and prediction purpose.

|  |  |
| --- | --- |
| **Name** | **Referred as** |
| PRODUCT TYPE | prod\_type |
| METROPOLITAN STATISTICAL AREA (MSA) OR METROPOLITAN DIVISION | cd\_msa |
| LOAN SEQUENCE NUMBER | id\_loan |
| Super Conforming Flag | flag\_sc |
| MATURITY DATE | dt\_matr |
| POSTAL CODE | zipcode |
| PROPERTY TYPE | prop\_type |
| ORIGINAL LOAN-TO-VALUE (LTV) | ltv |
| SELLER NAME | seller\_name |
| SERVICER NAME | servicer\_name |

* Test\_df contains all 26 columns

**Reasons for removing above variables:**

* Super Conforming Flag has no values, so dropped
* PRODUCT TYPE has only 1 value: FRM, so dropped
* LOAN SEQUENCE NUMBER is a synthetic value, so dropped
* METROPOLITAN STATISTICAL AREA (MSA) OR METROPOLITAN DIVISION is a factor with a large number of values that causes problems (memory, run time) for creating the regression model.  The model also contains zipcode and state, and they should be correlated since they are all to do with geographic location.  We leave state (st) and remove cd\_msa and POSTAL CODE, since state has the fewest factor levels
* MATURITY DATE is highly correlated  with origi\_loan\_term, and we leave orig\_loan\_term in the model:

> cor(numeric\_training\_df, method='pearson')  
  
                   orig\_loan\_term   
dt\_matr            0.99986971

* PROPERTY TYPE shows as insignificant when we re-run the model after removing these variables:

> training\_df <- subset(training\_df, select=-c(prod\_type, cd\_msa, id\_loan, flag\_sc, dt\_matr))

> model\_all <- lm(int\_rt ~ ., training\_df)

> summary(model\_all)

Call:

lm(formula = int\_rt ~ ., data = training\_df)

Residuals:

    Min      1Q  Median      3Q     Max

-2.8399 -0.1625 -0.0053  0.1487  4.1393

Coefficients: (44 not defined because of singularities)

                             Estimate Std. Error  t value Pr(>|t|)

...

ltv                                2.271e-04  1.092e-04    2.079 0.037606 \*

...

prop\_typeCO                       -1.933e-02  1.117e-01   -0.173 0.862587  
prop\_typeCP                       -5.488e-02  1.121e-01   -0.490 0.624383  
prop\_typeLH                        6.243e-02  1.129e-01    0.553 0.580324  
prop\_typeMH                        2.532e-01  1.117e-01    2.266 0.023477 \*  
prop\_typePU                       -2.246e-02  1.117e-01   -0.201 0.840620  
prop\_typeSF                       -1.203e-02  1.117e-01   -0.108 0.914172

* ORIGINAL LOAN-TO-VALUE (LTV) also shows as insignificant (and cltv, which is correlated, remains in the model)
* dropped SELLER NAME and SERVICER NAME, had values in the test data set that weren’t present in the training data set, and the predict () function couldn’t handle that:

> predictions = predict.lm(model\_all, test\_df)

Error in model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels) :

factor seller\_name has new levels BRANCHBANKING&TRUSTC

* **Regression method:**
* After doing the predictive analysis, it will predict the interest rate for next quarter and save it into .csv file e.g. historical\_data1\_predicted\_Q22005.csv

|  |  |  |
| --- | --- | --- |
| id\_loan | int\_rt | pred\_int\_rt |
| F105Q2000001 | 5.75 | 5.7699 |
| F105Q2000002 | 5.875 | 5.688596 |
| F105Q2000003 | 5.5 | 5.753888 |
| F105Q2000004 | 5.875 | 5.588366 |
| F105Q2000005 | 6 | 6.095431 |
| F105Q2000006 | 6 | 5.962271 |
| F105Q2000007 | 5.875 | 5.948064 |
| F105Q2000008 | 6 | 5.942049 |
| F105Q2000009 | 5.75 | 5.745632 |
| F105Q2000010 | 5.75 | 5.987929 |
| F105Q2000011 | 5.75 | 5.561317 |
| F105Q2000012 | 5.75 | 5.757765 |
| F105Q2000013 | 5.625 | 5.677143 |
| F105Q2000014 | 6.125 | 5.996066 |
|  |  |  |

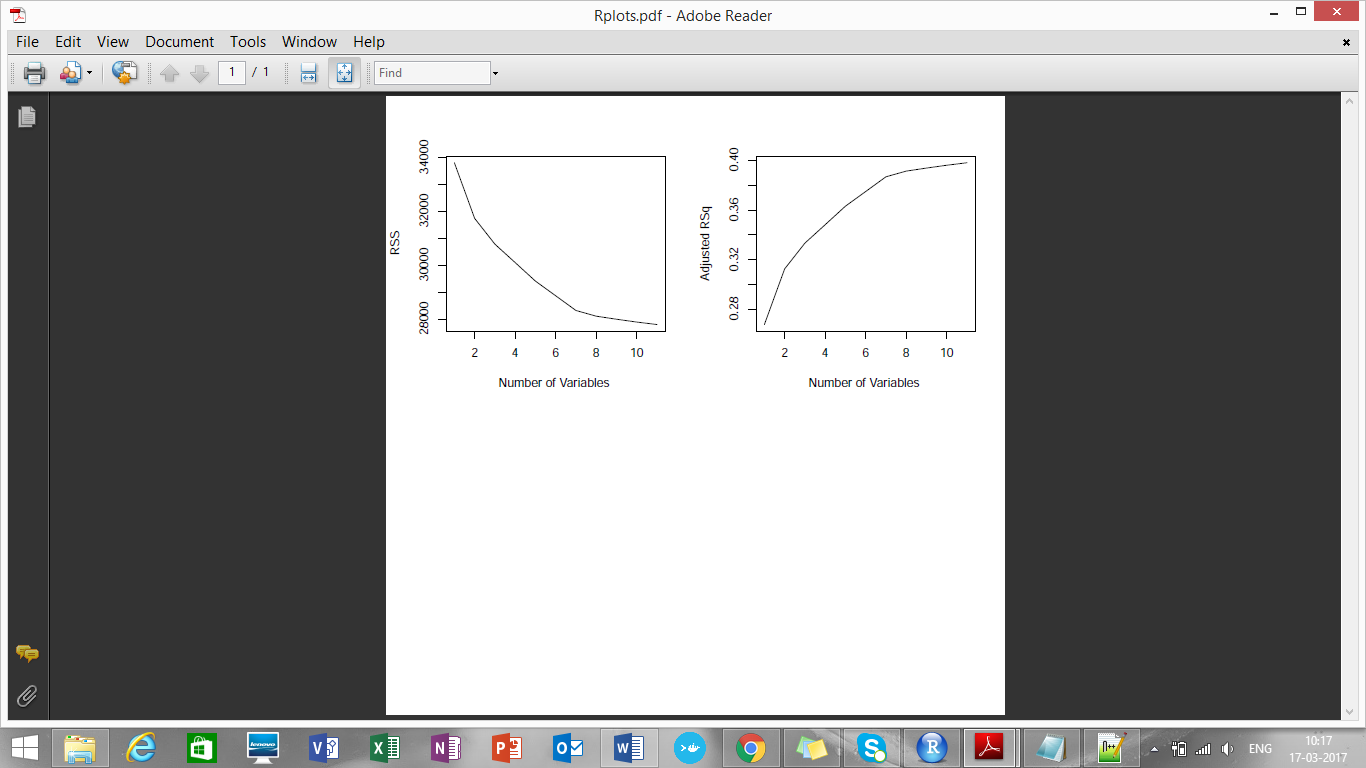
* It shows little variation between actual interest rate and predicted interest rate
* The adjusted R-squared value for this method is 0.4177, meaning that the independent variables explain around 42% of the interest rate variation.
* The predictive output generates the MAE, RMS, MAPE

ME RMSE MAE MPE MAPE

Test set 0.1081813 0.3148204 0.2406594 1.638906 4.10804

**Exhaustive Search:**

After exhaustive search, it draws Rplot like this:



**Significant variables by exhaustive search:**

orig\_loan\_term

orig\_upb

fico

occpy\_stsO

occpy\_stsS

mi\_pct

channelT

loan\_purposeP

stGA

dt\_first\_pi

cltv

[1] 33805.92 31739.40 30781.56 30097.55 29409.27 28864.46 28318.16 28106.52 27993.57 27888.20

[11] 27796.10

[1] 0.2678799 0.3126317 0.3333733 0.3481848 0.3630889 0.3748859 0.3867152 0.3912970 0.3937412

[10] 0.3960215 0.3980144

[1] 0.2678821 0.3126357 0.3333792 0.3481924 0.3630982 0.3748970 0.3867278 0.3913113 0.3937572

[10] 0.3960392 0.3980338

The adjusted R-squared value for this method is 0.3980, meaning that the independent variables explain around 40% of the interest rate variation.

**Forward selection:**

**Significant variables from forward selection:**

orig\_loan\_term

orig\_upb

fico

occpy\_stsO

occpy\_stsS

mi\_pct

channel

loan\_purposeP

stGA

dt\_first\_pi

cltv

[1] 33805.92 31739.40 30781.56 30097.55 29409.27 28864.46 28318.16 28106.52 27993.57 27888.20

[11] 27796.10

[1] 0.2678799 0.3126317 0.3333733 0.3481848 0.3630889 0.3748859 0.3867152 0.3912970 0.3937412

[10] 0.3960215 0.3980144

***Both exhaustive and forward selection has same significant variables***

**Backward Selection:**

**Significant variable from backward selection:**

orig\_loan\_term

orig\_upb

fico

occpy\_stsO

occpy\_stsS

mi\_pct

channel

loan\_purposeP

stGA

dt\_first\_pi

cltv

[1] 33805.92 31739.40 30781.56 30097.55 29409.27 28864.46 28318.16 28106.52 27993.57 27888.20

[11] 27796.10

[1] 0.2678799 0.3126317 0.3333733 0.3481848 0.3630889 0.3748859 0.3867152 0.3912970 0.3937412

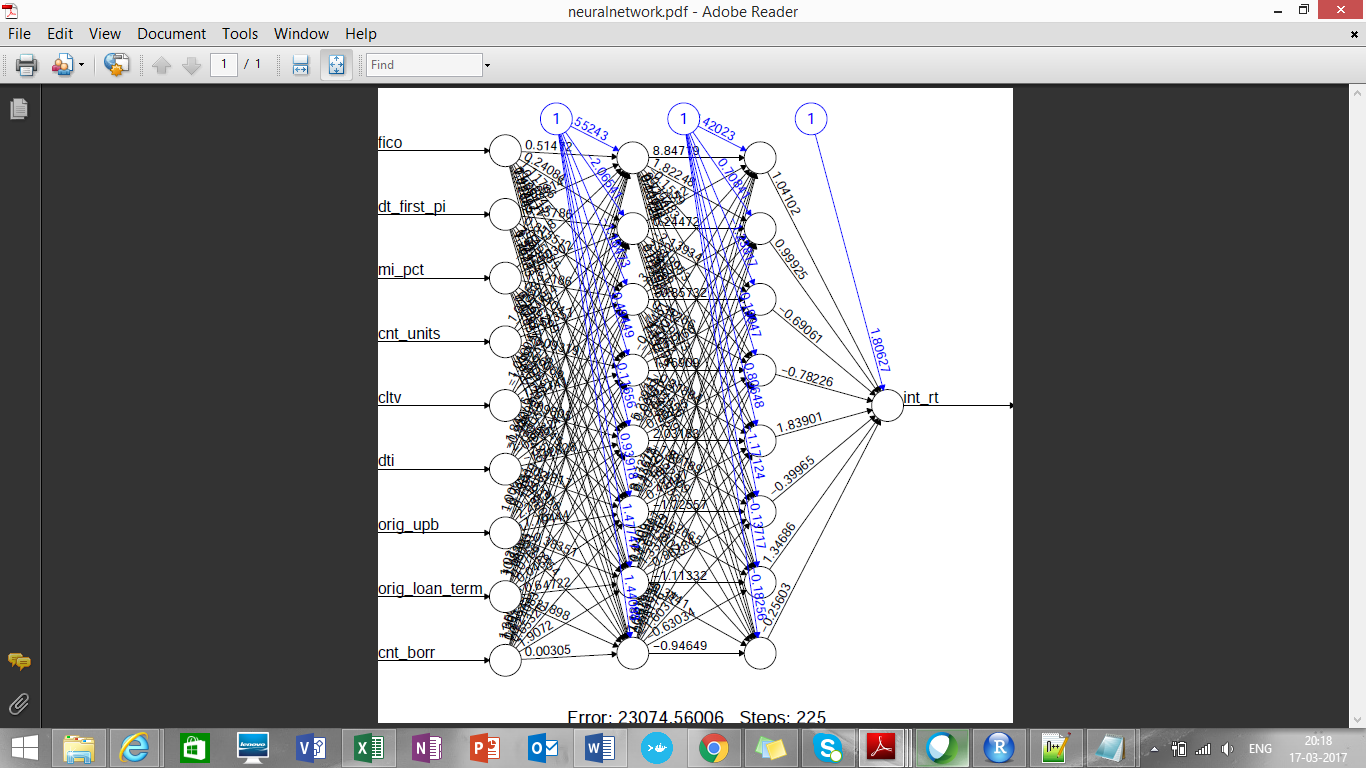
[10] 0.3960215 0.3980144

**Algorithms:**

* **Neural Networks:**

From Q12005 training data predicts interest rate for Q22005. So here is the actual IR and predicted IR(Refer: historical\_data1\_nn\_predicted\_Q22005.csv)

|  |  |  |
| --- | --- | --- |
| id\_loan | int\_rt | nn\_pred\_int\_rt |
| F105Q2000001 | 5.75 | 5.657186 |
| F105Q2000002 | 5.875 | 5.657186 |
| F105Q2000003 | 5.5 | 5.657186 |
| F105Q2000004 | 5.875 | 5.657186 |
| F105Q2000005 | 6 | 5.657186 |
| F105Q2000006 | 6 | 5.657186 |
| F105Q2000007 | 5.875 | 5.657186 |
| F105Q2000008 | 6 | 5.657186 |
| F105Q2000009 | 5.75 | 5.687506 |
| F105Q2000010 | 5.75 | 5.657186 |
| F105Q2000011 | 5.75 | 5.657186 |
| F105Q2000012 | 5.75 | 5.687506 |
| F105Q2000013 | 5.625 | 5.657186 |
| F105Q2000014 | 6.125 | 5.657186 |
| F105Q2000015 | 5.625 | 5.657186 |
| F105Q2000016 | 6.25 | 5.657186 |

****

* **Random Forest:**

From Q12005 training data predicts interest rate for Q22005. So here is the actual IR and predicted IR (Refer: historical\_data1\_rf\_predicted\_Q22005.csv)

|  |  |  |
| --- | --- | --- |
| id\_loan | int\_rt | rf\_pred\_int\_rt |
| F105Q2000001 | 5.75 | 5.688085 |
| F105Q2000002 | 5.875 | 5.880617 |
| F105Q2000003 | 5.5 | 5.643221 |
| F105Q2000004 | 5.875 | 5.835239 |
| F105Q2000005 | 6 | 5.898522 |
| F105Q2000006 | 6 | 5.749752 |
| F105Q2000007 | 5.875 | 5.886094 |
| F105Q2000008 | 6 | 5.783099 |
| F105Q2000009 | 5.75 | 5.896165 |
| F105Q2000010 | 5.75 | 5.779096 |
| F105Q2000011 | 5.75 | 5.756696 |
| F105Q2000012 | 5.75 | 5.863063 |
| F105Q2000013 | 5.625 | 5.781675 |
| F105Q2000014 | 6.125 | 5.829047 |
| F105Q2000015 | 5.625 | 5.669621 |
|  |  |  |

* **KNN Algorithm:**

Implemented but getting some random results

**Conclusion:**

**Regression method gives more appropriate results for predicting interest rates, then random forest and then neural network**

**Part2: Classification**

1. classification\_config.R

* It contains training\_quarter variable, which contains value empty or some value like Q22005

1. classification.R

* In this file, it refers to classification\_config.R file to find value for training quarter, if it is not present it takes default value as Q12005.
* The test data will be considered as quarter next to selected training quarter

1. Then it unzips data and refers performance file for prediction purpose
2. Adds a new variable “Deliquent” for col 4 is > 0, and creates new column for it