INST737: Introduction to Data Science

Project Milestone-2: Report

Instructor: Dr. Vanessa Frias-Martinez

Team5 : Sadaf Nasir Davre, Tanya Gupta, Ushasri Bhogaraju

Term : Fall '23

Milestone1 Summary : We started with a financial dataset of Fannie Mae with over 1M rows and 108 variables and a layout file of 108 observations and 10 variables, and performed Data transformation, Data cleaning and Software engineering steps. We identified our focus variables and performed statistical analysis on them using 'R'. We identified columns with Null values and removed the rows with them. We corrected data types and visualized and examined outliers.

We reproduce our research questions here for quick reference:

1. Can we accurately predict the 'Interest Rate' applied to a mortgage loan by a lending institution, using criterion such as the Quantum of loan, Loan-To-Value ratio, Debt-To-Income ratio, Loan Purpose, Number of Borrowers and Credit Score of the Borrower/s, using the Fannie Mae Acquisition and Performance 2022Q4 dataset?
2. Do all the borrower criteria such as the Quantum of loan, Loan-To-Value ratio, Debt-To-Income ratio, Loan Purpose, Number of Borrowers and Credit Score of the Borrower/s have equal influence over 'Interest Rate'? If not, which criteria have more effect on Interest Rate?
3. Do all lending institutions (presumed to be the 'Seller' in the dataset), evaluate the criterion variables similarly and fix the interest rate? (This question was revised because Milestone-1 analysis revealed that most values fall under the 'unknown' category named as 'other'. This aspect may or may not not provide for accuracy in training the models in Milestone-2.)

   Revised : *'Can we accurately predict the class of Occupancy Status("I", or"P" or 'S") in the Loan dataset, based on the IVs Original interest rate, bondRate, fedFundsRate, Borrower Credit Score and Original UPB (from the revised dataset)?*

Incorporating feedback from Instructors and from Peer reviews from Milestone 1:

1. Exploring new variables that may have more influence on our outcome variable, Original interest rate of the Loan : We added new variables after ascertaining the lack of predictive power of existing variables after attempting linear regression. Therefore, the entire process of data cleaning, removing null and NA values, removing outliers and performing linear regression was done twice, once with an existing set of IVs and again after adding new IVs.(Since the dataset has changed in dimensions)

2. Detailed explanation of variables :

S. No	Variable Name	Data type	Description
1	Original.Interest.Rate	numeric	The original interest rate on a mortgage loan as identified in the original mortgage note. It is a numeric continuous variable. This was the outcome variable we used to train models for linear regression, and Random Forests
2	Original.UPB	numeric	The dollar amount of the loan as stated on the note at the time the loan was originated.
3	Original.Loan.to.Value.Ratio..LTV.	numeric	The ratio, expressed as a percentage, obtained by dividing the amount of the loan at origination by the value of the Property
4	Original.Combined.Loan.to.Value.Ratio..CLTV.	numeric	The ratio, expressed as a percentage, obtained by dividing the amount of all known outstanding loans at origination by the value of the property.
5	Number.of.Borrowers	numeric	The number of individuals obligated to repay the mortgage loan.
6	Debt.To.Income..DTI.	numeric	The ratio obtained by dividing the total monthly debt expense by the total monthly income of the borrower at the time the loan was originated.

S. No	Variable Name	Data type	Description
7	Borrower.Credit.Score.at.Origination	numeric	A numerical value used by the financial services industry to evaluate the quality of borrower's credit. Credit scores are typically based on a proprietary statistical model that is developed for use by credit data repositories. These credit repositories apply the model to borrower credit information to arrive at a credit score. When this term is used by Fannie Mae, it is referring to the "Classic" FICO score developed by Fair Isaac Corporation.
8	Number.of.Units	numeric	The number of units comprising the related mortgaged property (one, two, three, or four).
9	Property.State	chr	Property Unpaid Balance. This represents the quantum of loan sanctioned to the Borrower
10	Seller.Name	chr	The name of the entity that delivered the mortgage loan to Fannie Mae.
11	Loan.Purpose	chr	An indicator that denotes whether the mortgage loan is either a refinance mortgage or a purchase money mortgage. Purpose may be the purchase of a new property or refinance of an existing lien (with cash out or with no cash out).
12	Occupancy.Status	chr	A categorical variable with 3 classes namely "P", "S" and "I". The classification describes the property occupancy status at the time the loan was originated.This variable describes whether the property being purchased with the loan, is the Primary residence of the borrower, Secondary Home or for investment. This is the variable we used as outcome variable for training Logistic Regression, Naïve Bayes and Decision Tree models.

S. No	Variable Name	Data type	Description
13	Property.Type	chr	An indicator that denotes whether the property type secured by the mortgage loan is a condominium, co-operative, planned urban development (PUD), manufactured home, or single-family home.
14	bondRate	num	10- year Treasury Bond Yield Rate
15	fedFundsRate	numeric	The 'Fed funds rate' is the interest rate at which lending institutions lend or borrow funds from other institutions overnight, on an uncollateralized basis
16	benchMarkMortgageRate	numeric	Benchmark average rate of 30 year fixed rate mortgages in the United States

***Milestone 2 Report** :*

Our dataset names and contents:

1)New_Variables_FM_AP_R_2022Q4.csv, (Dataset after infusion of new variables)

2) FinalVariables_FM_AP_R_2022Q4.csv (Old dataset with focus variables)

3)Final_FM_AP_R_2022Q4.csv files. (Parent dataset on which we used vlookup to infuse new variables and arrived at New_Variables_FM_AP_R_2022Q4.csv)

Notes on our Code and output:

1. As recorded in ReadMe, we have to have the above-mentioned files in the home directory before starting to run our code.
2. To create random training and test datasets for 1-d, we did not use setseed() function. This could result in minor changes in the results.

Learning from Milestone-2: We wish to summarize our learning and approach in a few lines here, to put our analysis in Milestone-2 into context.

1. We started with the focus variables dataset from Milestone-1 and after performing multivariate analysis understood that none of our variables had adequate predictive power and therefore, we infused new variables and performed the entire exercise with these variables.

2. In most parts of Milestone-2, we used the entire observations in the dataset and therefore, our graphs are busy. We realized that we could have worked with random samples of our dataset after we had to use them for Random Forests.
3. Every step of the analysis did not yield expected results in the first run. After completing the entire analysis and finding answers to the research questions, out of curiosity, we did a Run-2 for Logistic, Naive Bayes, Decision Trees, Bagging and Random Forest Models. Here, we infused additional variables from the existing dataset to train the models on. We got better results. In these discussions all the statistics are not explained as it was done for Run-1. Only the results are discussed. To preserve the integrity of the research, we preserved Run-1 results and analysis as it is and added Run-2 results and discussion separately.

Phase -1 cleaning and removing outliers (before performing linear regression and deciding to infuse new variables)

Cleaning our Dataset by removing outliers from the numerical values.

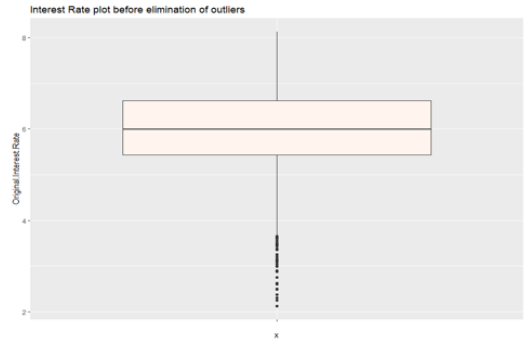
Our "FinalVariables_FM_AP_R_2022Q4.csv" dataset compiled using R in Milestone-1, had 8 Numerical, 3 Categorical and 2 Nominal variables as described above. These are all variables that describe various parameters of the loans from the Fannie Mae dataset. Our primary research question was to understand how these variables were helping in predicting the 'Original Interest Rate', also called the 'Note Rate' at the time of sanction of the loan. The presence of Boundary Outliers in these variables was detected using Histograms and summary statistics in Milestone-1 analysis. In this phase, we used BoxPlots to visualize and the IQR method to remove Outliers.

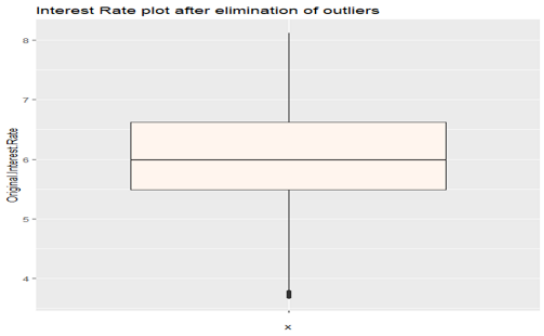
The IQR method involves determining the spread of the middle 50% of the data. The 1st Quantile is the 25th percentile of the data and represents the lower boundary of the middle 50% of the data. The 3rd Quantile is the 75th percentile of the data and represents the upper bound of 50% of the data. The InterQuartile Range is the difference between these two Quantiles. The lower bound is given by the formula Q1 - 1.5*IQR and the upper bound by Q3 + 1.5*IQR. Any datapoint that falls below the lower bound and above the upper bound can be deemed an outlier. We applied this principle to all our Numerical variables and removed the outliers by subsetting the middle 50% of the data.

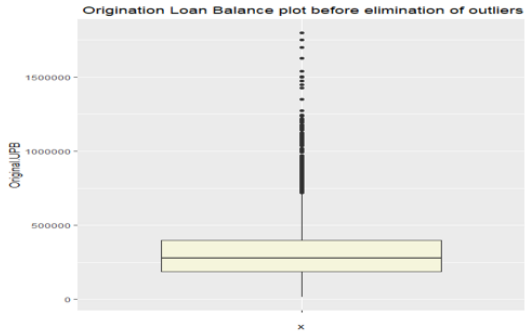
*Original Interest Rate Variable: T*he lower bound detected using the IQR method is 3.655 and the upper bound is 8.407. The interest rate values below and above these bounds may be data entry errors or special cases of loans which are not expected to be useful for prediction of the normal case of the dependent variable. We retained rows that fall between the lower and upper bounds and the dimensions of the dataset after this step are: 270,153 observations of 13 variables.

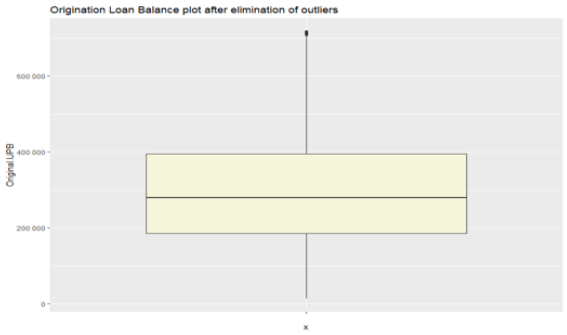
The datasets generated after eliminating outliers in one variable were used as input dataset for the next variable. Thus outliers were eliminated from all the variables and a final dataset named "Cleaned_FM2022Q4.csv" with dimensions 248,356 observations of 13 variables was saved. This dataset is being split into Test and Training datasets for use with our Prediction Model. .

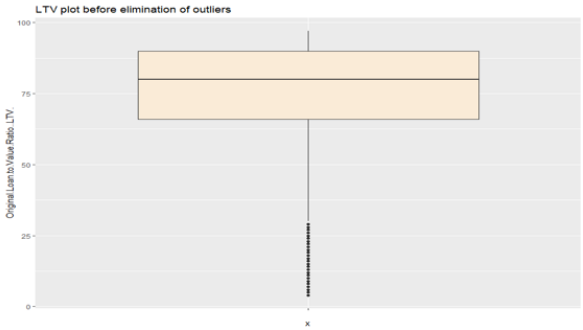
The Visualizations using BoxPlots are :

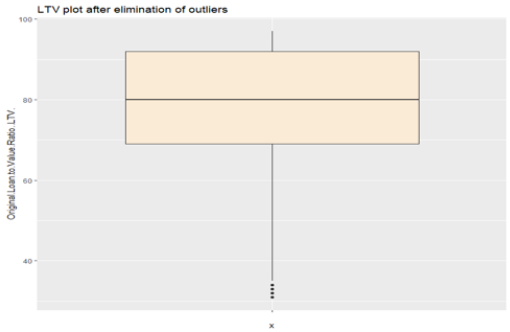


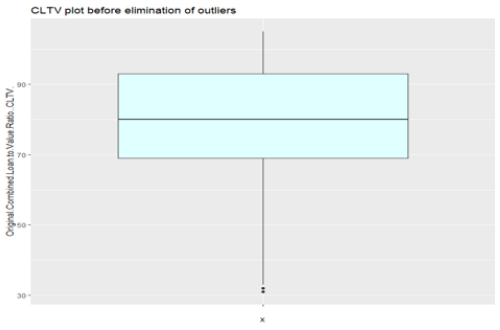


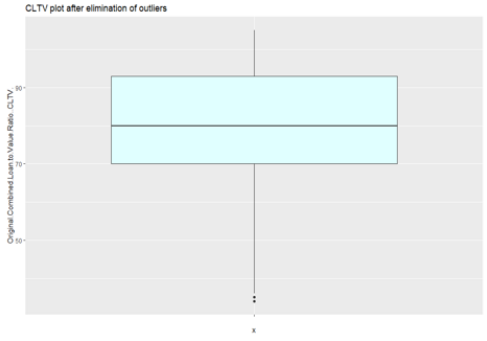




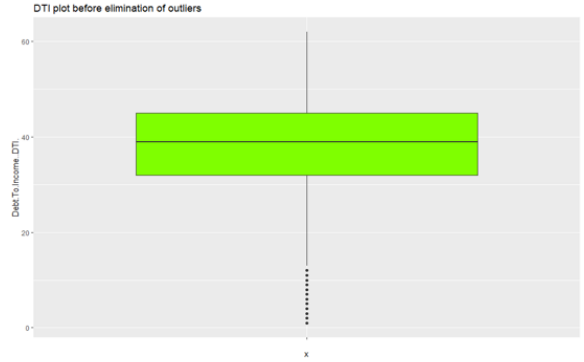




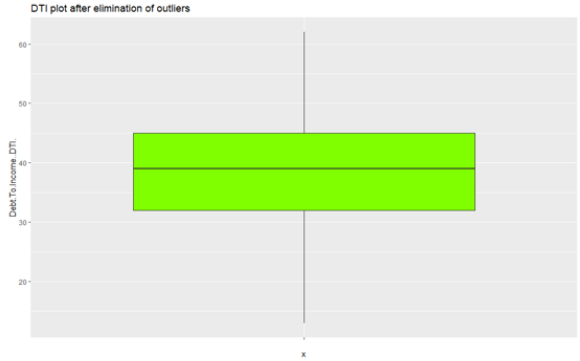




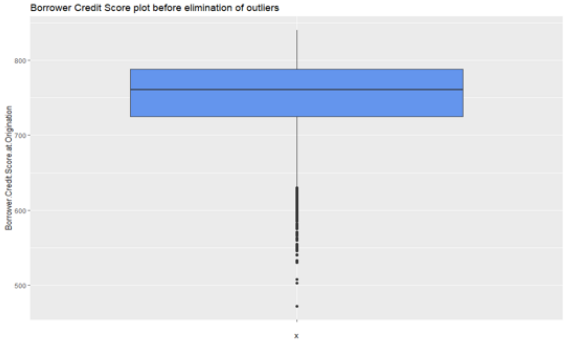
DTI plot before elimination of outliers



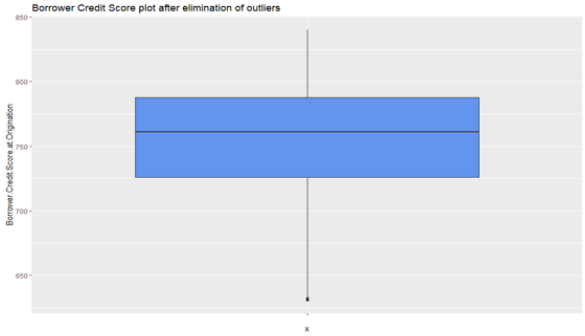
DTI plot after elimination of outliers



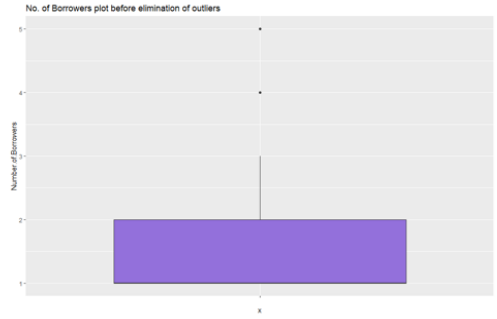
Borrower Credit Score plot before elimination of outliers



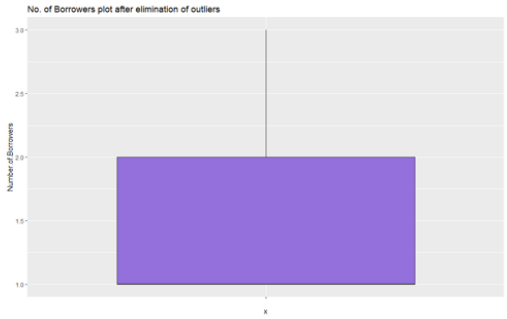
Borrower Credit Score plot after elimination of outliers



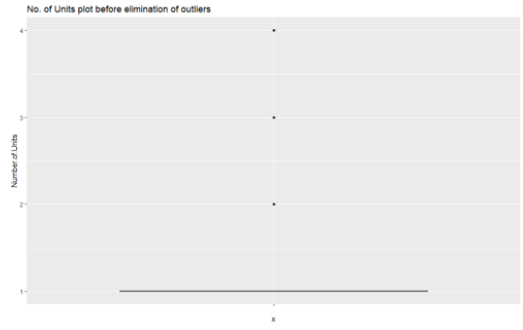
No. of Borrowers plot before elimination of outliers



No. of Borrowers plot after elimination of outliers



No. of Units plot before elimination of outliers



No. of Units plot after elimination of outliers

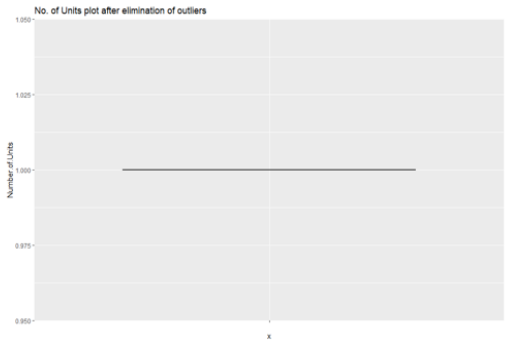


Table listing the number of outliers in each of the numeric variables and the corresponding Data Loss:

Variable Name	No of Obs with Outliers	No of Obs without Outliers	Data Loss (No. of Rows) %
Original.Interest.Rate	270993	270153	840 (0.3%)
Original.UPB	270153	267070	3083(1.1%)
Original.Loan.to.Value.Ratio..LTV.	267070	258261	8809(3.2%)
Original.Combined.Loan.to.Value.Ratio..CLTV.	258261	255948	2313(0.9%)
Debt.To.Income..DTI.	255948	254262	1686(0.7%)
Borrower.Credit.Score.at.Origination	254262	252632	1630(0.6%)
Number.of.Borrowers	252632	252145	487(0.2%)
Number.of.Units	252145	248356	3789(1.5%)

Question 1. Linear Regressions.

Splitting Data into Training and Test Datasets:

After elimination of outliers, we have our Cleaned dataset with the dimensions of 248,356 observations of 13 variables. We chose to split this dataset into Test and Training datasets with 25% and 75% of observations in the dataset respectively. To ensure random selection of rows, so that no bias enters either the training or the test datasets, we used R code to divide the row numbers in our cleaned dataset by 4 and if the remainder is zero, that row gets added to the test set. We thus made our "Training_FM2022Q4.csv" dataset with 186,267 observations of 13 variables and "Test_FM2022Q4.csv" dataset with 62,089 observations of 13 variables.

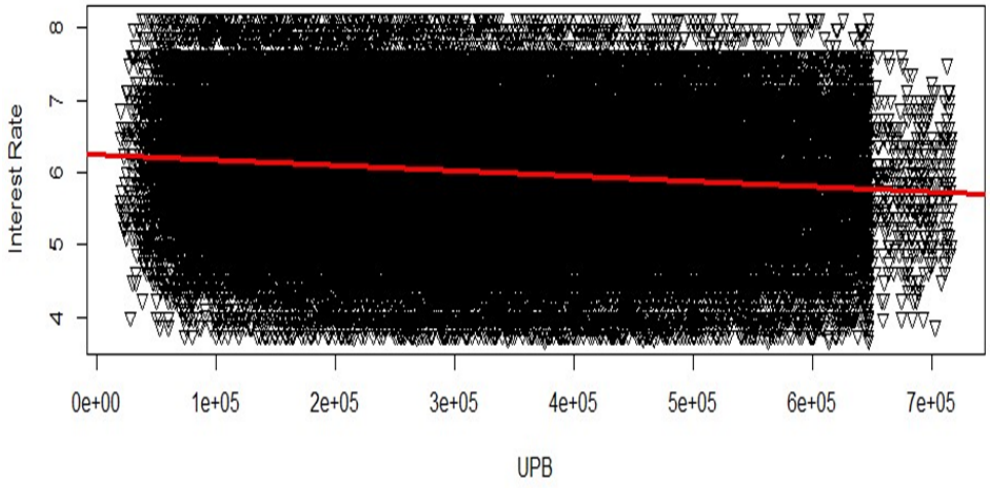
a. Linear Regression with each IV and DV:

We computed the linear models with each IV and the DV. We compiled a tabular representation of the main summary statistics of each of the models. We also visualized the Linear regression line using plots. At the end, we provide a summary table with the statistics of the models. The interpretation of intercept value is not done separately for each model. Intercept represents the value of DV when the value of all IVs in the linear model is equal to zero. It is the baseline value of the DV given IV = 0

1.Original.UPB and Original.Interest Rate

```
> data <- read.csv("Training_FM2022Q4.csv")
> lmUPB <- lm(Original.Interest.Rate ~ Original.UPB, data = data)
> summary(lmUPB)

Call:
lm(formula = Original.Interest.Rate ~ Original.UPB, data = data)

Residuals:
     Min       1Q   Median
-2.44386 -0.59876 -0.03537
      3Q      Max
 0.62890  2.34395

Coefficients:
              Estimate
(Intercept)   6.247e+00
Original.UPB -7.204e-07
             Std. Error
(Intercept)   4.516e-03
Original.UPB  1.337e-08
            t value Pr(>|t|)
(Intercept)   1383.2   <2e-16
Original.UPB   -53.9   <2e-16

(Intercept)  ***
Original.UPB ***
---
Signif. codes:
  0 '***' 0.001 '**' 0.01
  '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8268 on 186265 degrees of freedom
Multiple R-squared:  0.01536,   Adjusted R-squared:  0.01535
F-statistic:  2905 on 1 and 186265 DF,  p-value: < 2.2e-16
```

Plot of the Linear Regression line



Summary : The negative coefficient and the highly significant t-value suggest that there is a statistically significant negative relationship between the Original.UPB predictor variable and the Original.Interest.Rate outcome variable. However, the magnitude of the coefficient is extremely small, indicating that while there is a statistically significant relationship, the practical impact on one-unit change in Original.UPB on Original.Interest.Rate is minimal.

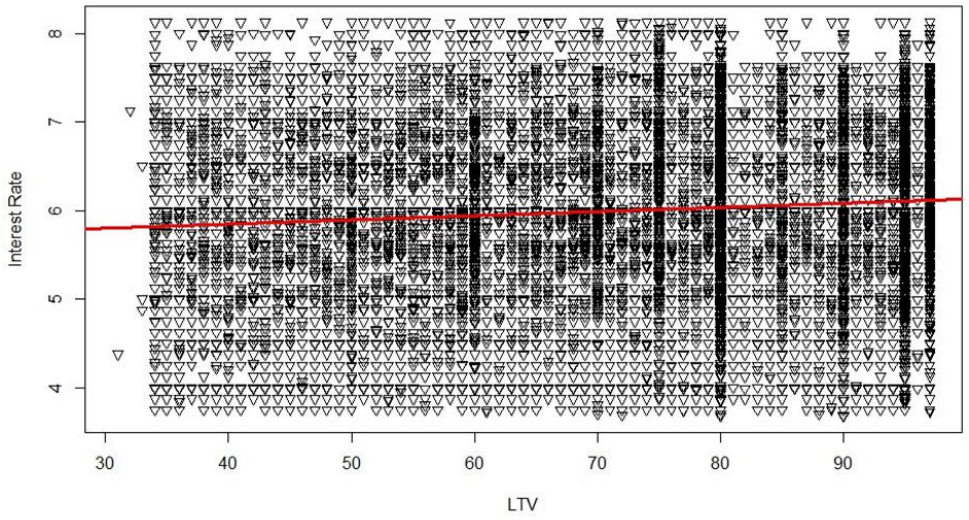
Interpretation:

Independent Variable Name	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
Original.UPB The model's Null Hypothesis: The correlation between the independent variable Original UPB and dependent variable Original Interest Rate is '0' (the value of the predictor variable 'Original UPB' has no effect on the value of the outcome variable)	Intercept value estimated by the model is 6.247e+00 The Standard error associated with this estimate is 4.516e-03 which is quite low. The t value for the intercept is 1383.2 which is large and the associated p-value is <2e-16, which is statistically highly significant	The magnitude of the coefficient estimated by the linear regression model is -7.204e-07. 1. The very small value 7.204e-07 indicates that one-unit change in the independent variable results in a very small change in the value of the dependent variable. 2. The relationship between the variables is inverse. One unit increase in UPB results in a very small decrease in interest rate. The standard error associated with this estimate is 1.337e-08 which is very low. The t-statistic is -53.09 which is large and the associated p-value <2e-16 indicates that this value is statistically significant.	Because the large value of F-statistic of the model, (2905 on 1 with 186265 degrees of freedom) and the associated small p-value < 2.2e-16, indicate that the explained variation of the model is significantly larger than what would be expected by chance. This suggests that the model is statistically significant in explaining the variation in the dependent variable and the Null can therefore be rejected. But the R squared value is 0.01536 which is low and the goodness of fit of the model between these two variables is small. We can say that UPB is therefore a predictive feature but with a low predictive power. The plot confirms the same.

2. Linear Model for LTV IV and DV Original Interest Rate

```
> lmLTV <- lm(Original.Interest.Rate ~ Original.Loan.to.Value.Ratio..LTV.)
> summary(lmLTV)

Call:
lm(formula = Original.Interest.Rate ~ Original.Loan.to.Value.Ratio..LTV.)

Residuals:
     Min       1Q   Median       3Q      Max
-2.40844 -0.60545 -0.02322  0.61646  2.30373

Coefficients:
                                  Estimate Std. Error
(Intercept)                      5.6620914  0.0095491
Original.Loan.to.Value.Ratio..LTV. 0.0046817  0.0001201
                                  t value Pr(>|t|)
(Intercept)                        592.95   <2e-16 ***
Original.Loan.to.Value.Ratio..LTV.   38.98   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8298 on 186265 degrees of freedom
Multiple R-squared:  0.008093,  Adjusted R-squared:  0.008088
F-statistic:  1520 on 1 and 186265 DF,  p-value: < 2.2e-16
```

Plot of the regression line for LTV variable



Interpretation:

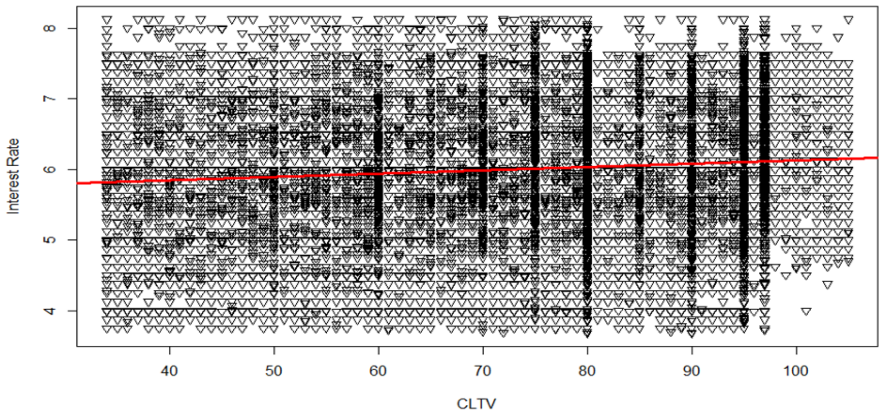
Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
LTV	5.662.0914	0.0046817 and associated p val is <2e-16 which is statistically significant.	Yes. The F-statistic value of 1520 on1 with 186265 DF and p-value <2.2e-16 indicates that the variability in the model cannot be explained as due to chance. Since the size of the coefficient of the independent variable is small, the variability is minimal. The R squared value is 0.008093 which is low and the goodness of fit of the model between these two variables is small. Hence we can say that this variable is a predictive feature but with a low predictive power. The plot confirms the same.

3.Linear Model for CLTV and Interest Rate:

3.

```
> lmCLTV <- lm(Original.Interest.Rate ~ Original.Combined.Loan.to.Value.Ratio..CLTV.)
> summary(lmCLTV)

Call:
lm(formula = Original.Interest.Rate ~ Original.Combined.Loan.to.Value.Ratio..CLTV.)

Residuals:
     Min       1Q   Median       3Q      Max
-2.40721 -0.60562 -0.02197  0.61303  2.30502

Coefficients:
                                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                            5.6607652  0.0095036  595.65   <2e-16 ***
Original.Combined.Loan.to.Value.Ratio..CLTV. 0.0046827  0.0001191   39.32   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8298 on 186265 degrees of freedom
Multiple R-squared:  0.008232,  Adjusted R-squared:  0.008227
F-statistic:  1546 on 1 and 186265 DF,  p-value: < 2.2e-16
```

Plot of the regression line



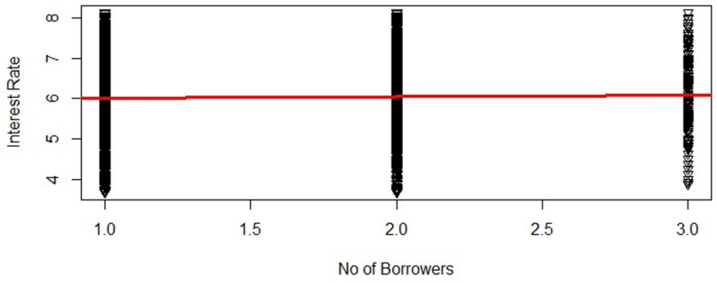
Interpretation

Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
CLTV	5.6607652	0.0046827 and associated p val is <2e-16 which is statistically significant.	Yes. The F-statistic value of 1546 on1 with 186265 DF and p-value <2.2e-16 indicates that the variability in the model cannot be explained as due to chance. Since the size of the coefficient of the independent variable is small, the variability is minimal. But the R squared value again is 0.008232 which is low and the goodness of fit of the model between these two variables is small. Hence we can say that this variable is a predictive feature but with a low predictive power. The plot confirms the same.

4.Linear Model for Borrowers variable

```
> lmBorrowers <- lm(Original.Interest.Rate ~ Number.of.Borrowers)
> summary(lmBorrowers)

Call:
lm(formula = Original.Interest.Rate ~ Number.of.Borrowers)

Residuals:
     Min       1Q   Median       3Q      Max
-2.36911 -0.55411 -0.02194  0.61306  2.11306

Coefficients:
                    Estimate Std. Error  t value Pr(>|t|)
(Intercept)         5.979772   0.005694 1050.139   <2e-16 ***
Number.of.Borrowers 0.032171   0.003670    8.765   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.833 on 186265 degrees of freedom
Multiple R-squared:  0.0004123, Adjusted R-squared:  0.0004069
F-statistic: 76.82 on 1 and 186265 DF,  p-value: < 2.2e-16
```

Plot of the regression Line



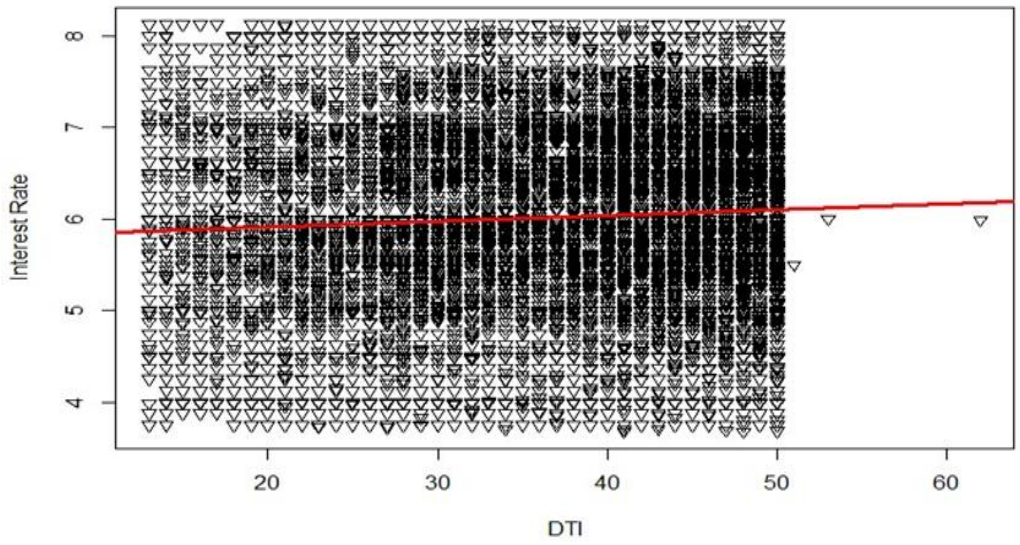
Interpretation

Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
No of Borrowers	5.979772	0.032171 and associated p val is <2e-16 which is statistically significant.	No The R squared value again is 0.0004123 which is very small and therefore goodness of fit of the model for the data between these two variables is minimal. We can say that this variable is not a predictive feature. The Plot also confirms the same.

5.Linear Model for DTI and OIR

```
> abline(lmBorrowers, col = "red", lwd =3)
> lmDTI <- lm(Original.Interest.Rate ~ Debt.To.Income..DTI.)
> summary(lmDTI)

Call:
lm(formula = Original.Interest.Rate ~ Debt.To.Income..DTI.)

Residuals:
     Min       1Q   Median       3Q      Max
-2.42035 -0.58602 -0.03447  0.61630  2.25806

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          5.7831743  0.0084962  680.67   <2e-16 ***
Debt.To.Income..DTI. 0.0064435  0.0002189   29.43   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8313 on 186265 degrees of freedom
Multiple R-squared:  0.004629,  Adjusted R-squared:  0.004624
F-statistic: 866.3 on 1 and 186265 DF,  p-value: < 2.2e-16
```

Plotting the regression line



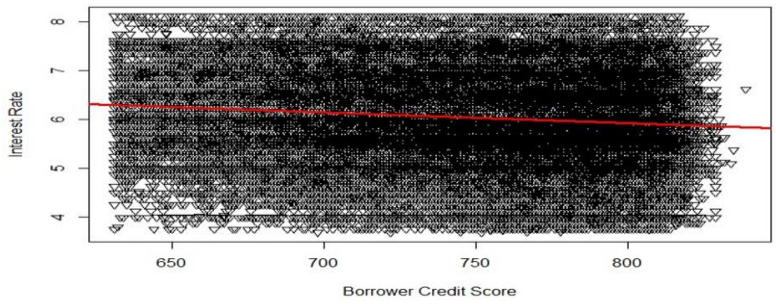
Interpretation:

Independent Variable	*What is the intercept?*	*What is the coefficient? (Slope)*	*Is it a Predictive Feature*
DTI	*5.7831743*	*0.0064435* and associated p val is <2e-16 which is statistically significant.	*Yes. The F-statistic value of 866.3 on1 with 186265 DF and p-value <2.2e-16 indicates that the variability in the model cannot be explained as due to chance. But since the size of the coefficient of the independent variable is small, the variability is minimal.* *But the R squared value again is 0.004629 which is low and the goodness of fit of the model for the data between these two variables is small. Hence, we can say that this variable is a predictive feature but with a low predictive power. The Plot confirms the same.*

6. Linear Model of Borrower Credit Score

```
> lmCS <- lm(Original.Interest.Rate ~ Borrower.Credit.Score.at.Origination)
> summary(lmCS)

Call:
lm(formula = Original.Interest.Rate ~ Borrower.Credit.Score.at.Origination)

Residuals:
     Min       1Q   Median       3Q      Max
-2.54853 -0.58972 -0.03241  0.63154  2.23765

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                      7.693e+00  3.474e-02  221.44   <2e-16 ***
Borrower.Credit.Score.at.Origination -2.211e-03  4.601e-05  -48.05   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8281 on 186265 degrees of freedom
Multiple R-squared:  0.01224,   Adjusted R-squared:  0.01224
F-statistic:  2309 on 1 and 186265 DF,  p-value: < 2.2e-16
```

Plotting the regression line



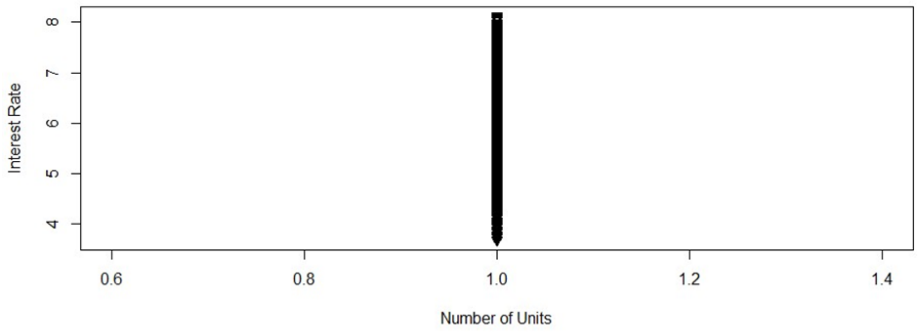
Interpretation:

Independent Variable	*What is the intercept?*	*What is the coefficient? (Slope)*	*Is it a Predictive Feature*
Borrower Credit Score	*7.693e+00*	*-2.211e-03* and associated p val is <2e-16 which is statistically significant.	*Yes. The F-statistic value of 2309 on1 with 186265 DF and p-value <2.2e-16 indicates that the variability in the model cannot be explained as due to chance. But since the size of the coefficient of the independent variable is small, the variability is minimal. Also, the variables are inversely related, and the R squared value is 0.01224 which is low and therefore the goodness of fit of the model for the data between these two variables is small. Hence, we can say that this variable is a predictive feature but with a low predictive power. The Plot confirms the same.*

7. Linear Model for No. of Units variable and OIR

```
> lmUnits <- lm(Original.Interest.Rate ~ Number.of.Units)
> summary(lmUnits)

Call:
lm(formula = Original.Interest.Rate ~ Number.of.Units)

Residuals:
     Min       1Q   Median       3Q      Max
-2.35173 -0.53673 -0.03673  0.59827  2.09827

Coefficients: (1 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     6.026726   0.001931    3122   <2e-16 ***
Number.of.Units       NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8332 on 186266 degrees of freedom

> plot(Number.of.Units, Original.Interest.Rate,
+      xlab= "Number of Units",
+      ylab = "Interest Rate", pch =6)
> abline(lmUnits, col = "red", lwd =3)
Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...) :
  'a' and 'b' must be finite
>
```

Plotting the variable also shows that the linear regression line cannot be visualized because of singularities in values of the IV (there is only one value for this IV after removing outliers)



Interpretation:

Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
No of Units	6.026726	NA. Not determinable due to singularities	No. All the interest rate values are converged at No. of Units =1. This indicates that while removing outliers, our data does not have variability for the number of units. The Plot confirms the same. Hence No of Units is not a predictive feature for our dependent variable.

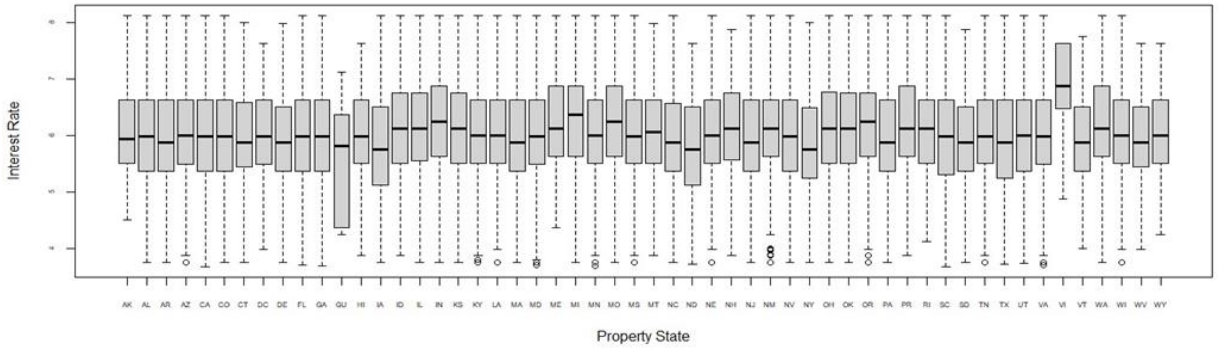
Linear models of Categorical and nominal variables:

To build linear models for our Categorical and nominal variables we used as.factor() function and built our models using R code

8.Linear Model for Property State Variable (Nominal)

```
> par(cex.axis = 0.5)
> summary(lmPS)

Call:
lm(formula = Original.Interest.Rate ~ Property.State, data = trainFactors)

Residuals:
     Min       1Q   Median       3Q      Max
-2.51306 -0.59918 -0.01972  0.62640  2.27145

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        6.043307   0.050372 119.973  < 2e-16 ***
Property.StateAL -0.051623   0.052899  -0.976 0.329124
Property.StateAR -0.059941   0.054772  -1.094 0.273793
Property.StateAZ -0.007340   0.051450  -0.143 0.886555
Property.StateCA -0.059704   0.050875  -1.174 0.240586
Property.StateCO -0.050608   0.051694  -0.979 0.327578
Property.StateCT -0.067535   0.054135  -1.248 0.212211
Property.StateDC -0.046249   0.072205  -0.641 0.521832
Property.StateDE -0.147947   0.058111  -2.546 0.010900 *
Property.StateFL -0.033588   0.050823  -0.661 0.508696
Property.StateGA -0.028099   0.051283  -0.548 0.583752
Property.StateGU -0.418307   0.341642  -1.224 0.220801
Property.StateHI -0.017433   0.070723  -0.247 0.805294
Property.StateIA -0.189754   0.054430  -3.486 0.000490 ***
Property.StateID  0.068899   0.055009   1.253 0.210387
Property.StateIL  0.095538   0.051335   1.861 0.062738 .
Property.StateIN  0.148821   0.051751   2.876 0.004032 **
Property.StateKS  0.080288   0.055083   1.458 0.144960
Property.StateKY  0.034291   0.053908   0.636 0.524711
Property.StateLA  0.005194   0.054467   0.095 0.924021
Property.StateMA -0.069126   0.053175  -1.300 0.193608
Property.StateMD -0.048979   0.052210  -0.938 0.348178
Property.StateME  0.145447   0.060930   2.387 0.016982 *
Property.StateMI  0.219755   0.051469   4.270 1.96e-05 ***
Property.StateMN -0.003111   0.051955  -0.060 0.952259
Property.StateMO  0.194219   0.052100   3.728 0.000193 ***
Property.StateMS  0.002954   0.056791   0.052 0.958522
Property.StateMT  0.006269   0.060382   0.104 0.917314
Property.StateNC -0.113987   0.051228  -2.225 0.026075 *
Property.StateND -0.230117   0.067384  -3.415 0.000638 ***
Property.StateNE -0.002575   0.055281  -0.047 0.962842
Property.StateNH  0.103700   0.058425   1.775 0.075909 .
Property.StateNJ -0.076739   0.051805  -1.481 0.138531
Property.StateNM  0.089069   0.055933   1.592 0.111291
Property.StateNV -0.034709   0.052956  -0.655 0.512190
Property.StateNY -0.207827   0.051624  -4.026 5.68e-05 ***
Property.StateOH  0.085152   0.051392   1.657 0.097541 .
Property.StateOK  0.073823   0.053582   1.378 0.168282
Property.StateOR  0.119209   0.052710   2.262 0.023723 *
Property.StatePA -0.076611   0.051435  -1.489 0.136364
Property.StatePR  0.105859   0.067922   1.559 0.119107
Property.StateRI  0.065660   0.062621   1.049 0.294394
Property.StateSC -0.094736   0.052099  -1.818 0.069008 .
Property.StateSD -0.101897   0.061325  -1.662 0.096597 .
Property.StateTN -0.006560   0.051700  -0.127 0.899031
Property.StateTX -0.112291   0.050706  -2.215 0.026791 *
Property.StateUT -0.014584   0.053078  -0.275 0.783496
Property.StateVA -0.039070   0.051713  -0.756 0.449938
Property.StateVI  0.692393   0.266545   2.598 0.009387 **
Property.StateVT -0.081210   0.069057  -1.176 0.239607
Property.StateWA  0.121158   0.051710   2.343 0.019130 *
Property.StateWI  0.051209   0.052526   0.975 0.329598
Property.StateWV -0.086487   0.060221  -1.436 0.150957
Property.StateWY  0.025254   0.065063   0.388 0.697900
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8277 on 186213 degrees of freedom
Multiple R-squared:  0.01343,   Adjusted R-squared:  0.01315
F-statistic: 47.82 on 53 and 186213 DF,  p-value: < 2.2e-16
```

Plot of Property State against Interest Rates shows slight variations in mean interest rates for few states.



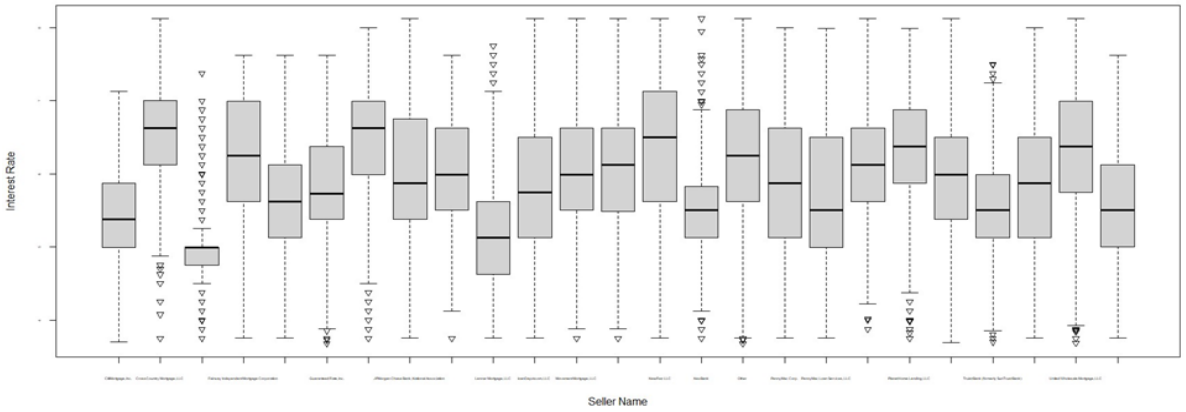
Interpretation:

Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
Property State (Nominal)	6.043307	Coefficient magnitudes vary from 0.006 to 0.21. The statistical significance of these coefficients varies also.Coeffs of 14/55 states have low p-values. For other states, the coefficients have little statistical significance.	Overall, No. The high F-statistic 47.82 and the low associated p-value indicate that the model as a whole is statistically significant in explaining variability in interest rate. However, the low R squared volumes 0.01343 indicate that the predictive power of Property State variable in predicting Original interest rate is not significant. Based on the Plot we can say that some Property State levels such as VI and GU in the variable show variability with interest rate while some others show little variability

9.Linear Model for Seller Name Variable (Nominal)

```
> lmSN <- lm(Original.Interest.Rate ~ Seller.Name, data = trainFactors)
> summary(lmSN)

Call:
lm(formula = Original.Interest.Rate ~ Seller.Name, data = trainFactors)

Residuals:
     Min       1Q   Median       3Q      Max
-2.79773 -0.54652 -0.00125  0.57948  2.62070

Coefficients:
                                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                                          5.39216    0.01699 317.294  < 2e-16 ***
Seller.NameCrossCountry Mortgage, LLC                1.15557    0.02383  48.496  < 2e-16 ***
Seller.NameDHI Mortgage Company, Ltd.               -0.54158    0.02278 -23.774  < 2e-16 ***
Seller.NameFairway Independent Mortgage Corporation  0.84634    0.02075  40.788  < 2e-16 ***
Seller.NameFifth Third Bank, National Association    0.26319    0.02212  11.899  < 2e-16 ***
Seller.NameGuaranteed Rate, Inc.                     0.41544    0.02144  19.373  < 2e-16 ***
Seller.NameGuild Mortgage Company LLC                1.10583    0.02470  44.772  < 2e-16 ***
Seller.NameJPMorgan Chase Bank, National Association 0.56004    0.02265  24.725  < 2e-16 ***
Seller.NameLakeview Loan Servicing, LLC              0.65299    0.02274  28.715  < 2e-16 ***
Seller.NameLennar Mortgage, LLC                     -0.19746    0.02646  -7.461 8.61e-14 ***
Seller.NameloanDepot.com, LLC                        0.39786    0.02145  18.547  < 2e-16 ***
Seller.NameMovement Mortgage, LLC                    0.69828    0.02071  33.725  < 2e-16 ***
Seller.NameNationStar Mortgage, LLC                  0.74056    0.02248  32.937  < 2e-16 ***
Seller.NameNewRez LLC                                0.98265    0.02183  45.011  < 2e-16 ***
Seller.NameNexBank                                   0.11213    0.02391   4.690 2.74e-06 ***
Seller.NameOther                                     0.77836    0.01723  45.177  < 2e-16 ***
Seller.NamePennyMac Corp.                            0.51160    0.02038  25.098  < 2e-16 ***
Seller.NamePennyMac Loan Services, LLC               0.24214    0.02211  10.950  < 2e-16 ***
Seller.NamePHH Mortgage Corporation                  0.71843    0.01975  36.378  < 2e-16 ***
Seller.NamePlanet Home Lending, LLC                  0.95106    0.02116  44.941  < 2e-16 ***
Seller.NameRocket Mortgage, LLC                      0.54113    0.01784  30.328  < 2e-16 ***
Seller.NameTruist Bank (formerly SunTrust Bank)      0.17349    0.02030   8.546  < 2e-16 ***
Seller.NameU.S. Bank N.A.                            0.49123    0.02050  23.960  < 2e-16 ***
Seller.NameUnited Wholesale Mortgage, LLC            0.92635    0.01884  49.179  < 2e-16 ***
Seller.NameWells Fargo Bank, N.A.                    0.23408    0.01889  12.392  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7841 on 186242 degrees of freedom
Multiple R-squared:  0.1144,    Adjusted R-squared:  0.1143
F-statistic:  1003 on 24 and 186242 DF,  p-value: < 2.2e-16
```

Plot:



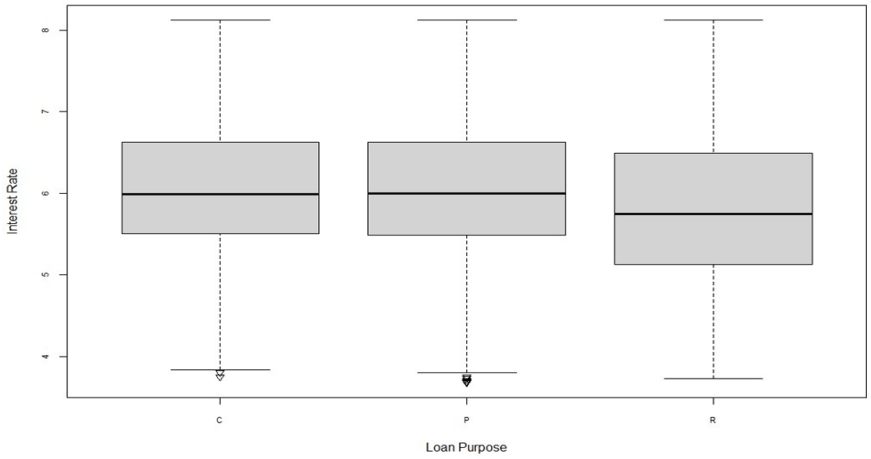
Interpretation:

Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
Seller Name (Nominal)	5.39216	Coefficient values range from 0.23 to 1.15 and all the values are statistically highly significant..	Yes, but Limited. The high F-statistic 1003 and the low associated p-value < 2.2e-16 indicate that the model as a whole is statistically significant in explaining variability in interest rate. The R squared volumes of 0.1144 indicate limited predictive power in explaining the variability in interest rate. Based on the Plot we can say that many Seller Name levels in the variable show variability with interest rate while two levels do not show statistically significant variability

10.Linear model loan purpose

```
> lmLP <- lm(Original.Interest.Rate ~ Loan.Purpose, data = trainFactors)
> summary(lmLP)

Call:
lm(formula = Original.Interest.Rate ~ Loan.Purpose, data = trainFactors)

Residuals:
     Min       1Q   Median       3Q      Max
-2.36524 -0.54124 -0.04024  0.58476  2.35183

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept)    6.022691   0.004857 1239.999  < 2e-16 ***
Loan.PurposeP  0.017550   0.005312    3.304 0.000954 ***
Loan.PurposeR -0.249525   0.010761  -23.188  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8316 on 186264 degrees of freedom
Multiple R-squared:  0.003943,  Adjusted R-squared:  0.003933
F-statistic: 368.7 on 2 and 186264 DF,  p-value: < 2.2e-16
```

Plotting:



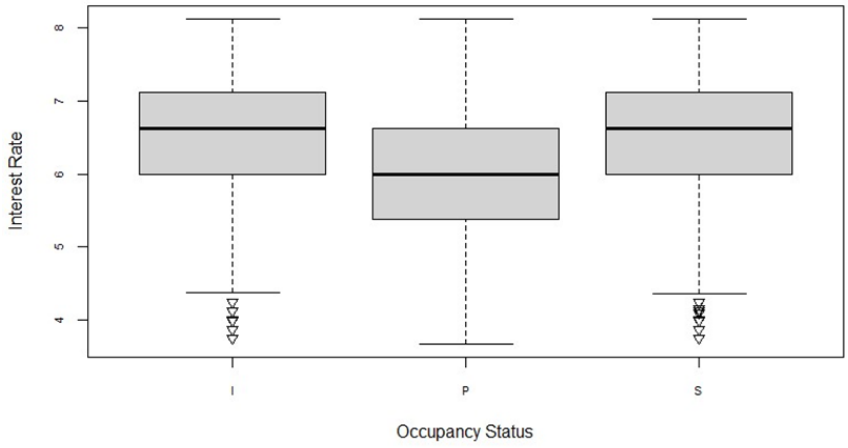
Interpretation:

Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
Loan Purpose (Categorical)	6.022691	For Refinance it is -0.24955 and for Purchase it is 0.0175500. Both have significant p-values	As a whole, No. The high F-statistic 368.7 and the low associated p-value < 2.2e-16 indicate that the model as a whole is statistically significant in explaining variability in interest rate. However, the R squared volumes of 0.003943 indicate insignificant predictive power in explaining the variability in interest rate. Based on the Plot we can say that 'Refinance' level in the Loan Purpose variable shows variability with interest rate while the other two levels do not show any variability

11.Linear model with Occupancy Status as IV and OIR as DV

```
> lmOS <- lm(Original.Interest.Rate ~ Occupancy.Status, data = trainFactors)
> summary(lmOS)

Call:
lm(formula = Original.Interest.Rate ~ Occupancy.Status, data = trainFactors)

Residuals:
     Min       1Q   Median       3Q      Max
-2.83005 -0.59301  0.02199  0.65699  2.15699

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        6.580047   0.006923 950.435  < 2e-16 ***
Occupancy.StatusP -0.612035   0.007203 -84.966  < 2e-16 ***
Occupancy.StatusS -0.104440   0.013574  -7.694 1.43e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8143 on 186264 degrees of freedom
Multiple R-squared:  0.0448,    Adjusted R-squared:  0.04479
F-statistic:  4368 on 2 and 186264 DF,  p-value: < 2.2e-16
```

Plot:



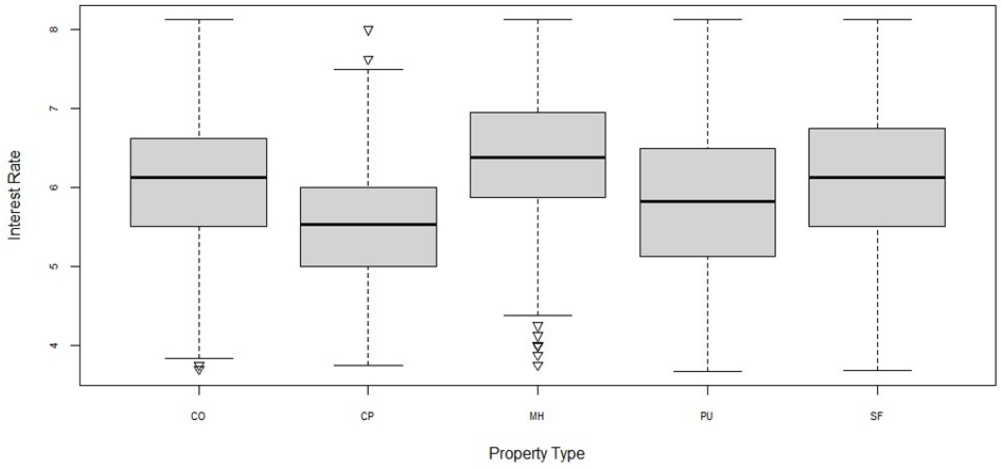
Interpretation:

Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
Occupancy Status (Categorical)	5.39216	For Level Primary Residence, P:-0.612035 with p-val <2e-16 S level it is -0.104440 with very low p -value	No. The high F-statistic 368.7 and the low associated p-value < 2.2e-16 indicate that the model as a whole is statistically significant in explaining variability in interest rate. However, the R squared volumes of 0.03943 indicate insignificant predictive power in explaining the variability in interest rate. Based on the Plot we can say that Primary Residence level in the Occupancy Status variable shows variability with interest rate while Investor and Secondary Home do not show much variability

11.Linear Model for Property Type variable

```
> lmPT <- lm(Original.Interest.Rate ~ Property.Type, data = trainFactors)
> summary(lmPT)

Call:
lm(formula = Original.Interest.Rate ~ Property.Type, data = trainFactors)

Residuals:
     Min       1Q   Median       3Q      Max
-2.58719 -0.61357  0.01143  0.63643  2.40099

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      6.102740   0.006281  971.62   <2e-16 ***
Property.TypeCP -0.503727   0.030071  -16.75   <2e-16 ***
Property.TypeMH  0.234454   0.017273   13.57   <2e-16 ***
Property.TypePU -0.261903   0.007131  -36.73   <2e-16 ***
Property.TypeSF  0.010830   0.006768    1.60     0.11
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8224 on 186262 degrees of freedom
Multiple R-squared:  0.02587,   Adjusted R-squared:  0.02585
F-statistic:  1237 on 4 and 186262 DF,  p-value: < 2.2e-16
```

Plot



Interpretation:

Independent Variable	What is the intercept?	What is the coefficient? (Slope)	Is it a Predictive Feature
Property Type (Categorical)	6.102740	For Level CP:-0.503727 MH:0.234454 PU:-0.261903 SF: 0.010830 of these CP, MH and PU levels have low p-values and therefore are statistically significant	No. The high F-statistic 368.7 and the low associated p-value < 2.2e-16 indicate that the model as a whole is statistically significant in explaining variability in interest rate. However, the R squared volumes of 0.03943 indicate insignificant predictive power in explaining the variability in interest rate. Based on the Plot we can say that except for SF level of the Property Type variable, remaining levels show statistically significant coefficient values, but the values are small.

Most predictive features according to training data: We summarize hereunder the predictive power of all the IVs assessed by building linear models individually with the DV.

Milestone-2 Analysis : Decision if whether the variable is a predictive feature using Linear Regression Model for further analysis

S.No	Linear Model on training dataset Built between	Coefficient of IV	p-value	Multiple R-Squared	Is it a Predictive Feature
1	Original.UPB & OIR	-7.204e-07	<2e-16	0.01536	Statistically significant but weak predictive power
2	Origina LTV. & OIR	0.0046817	<2e-16	0.008093	Statistically significant but very weak predictive power
3	Original CLTV. & OIR	0.0046827	<2e-16	0.008232	Statistically significant but very weak predictive power
4	Number.of.Borrowers & OIR	0.032171	<2e-16	0.000412	No. Though statistically significant overall the model is very weak predictive power
5	Debt.To.Income..DTI. & OIR	0.0064435	<2e-16	0.004629	Statistically significant but very weak predictive power
6	Borrower.Credit.Score.at. OIR	-2.211e-03	<2e-16	0.01224	Statistically significant but very weak predictive power
7	Number.of.Units & OIR	Singularities	NA.	NA.	No. After removal of outliers, this variable is left with only 1 value.

S.No	Linear Model on training dataset Built between	Coefficient of IV	p-value	Multiple R-Squared	Is it a Predictive Feature
8	Property.State as Factor & OIR	vary from 0.006 to 0.21.	14/55 levels have statistically significant coeffs	0.01343	Overall, No. Many levels are Statistically insignificant. R-squared very low indicating the goodness of fit of the model is negligible.
9	Seller.Name as Factor & OIR	range from 0.23 to 1.15	all levels have statistically significant coeffs	0.1144	Stronger than other variables, statistically significant but weak coefficients. R-squared very low indicating the goodness of fit of the model is negligible.
10	Loan.Purpose as Factor & OIR	For Refinance it is -0.24955 and for Purchase it is 0.0175500	<2e-16 for one level and 0.000954 for another both significant	0.003943	No. Though statistically significant overall the model is very weak
11	Occupancy.Status as Factor & OIR	-0.612035 for Primary residence level, -0.104440 for S level	<2e-16 and 1.43e-14	0.0448	Statistically significant but weak predictive power
12	Property.Type as Factor & OIR	CP:-0.503727 MH:0.234454 PU:-0.261903 SF: 0.010830	<2e-16	0.02587	Statistically significant but weak predictive power

At this stage we understood that though few of our IVs have slightly better coefficient values that were statistically significant, the R-squared values of the models indicate they are not a good predictive feature.

We list hereunder variables with slightly better predictive power than others.

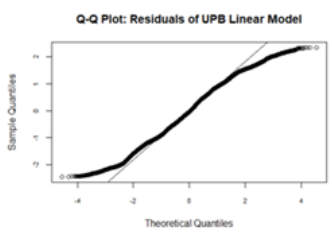
The most predictive features according to Linear Models built on Training Data are :

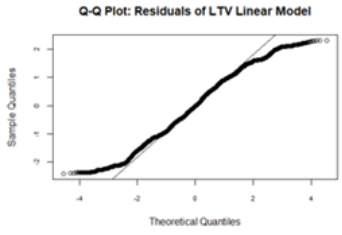
1. Original UPB
2. Borrower Credit Score at Origination
3. Seller Name

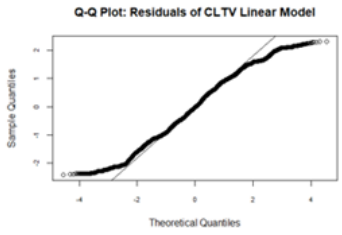
Residuals:

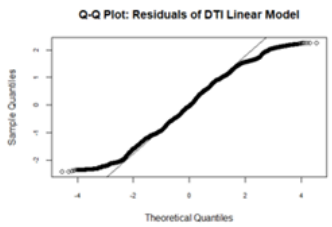
Residuals are the difference between the actual values of the dependent variable and the values predicted using our Linear Model. The residuals in a linear model can be found using the resid() function in R. Linear regressions and Predictions work under the assumption that the Residuals follow a normal distribution. Whether or not they follow a normal distribution can be tested using the qqnorm() and qqline() functions in R. When we plot the residual using qqnorm() and add a line using qqline() functions, if both of them are near, we can say that the residuals are following a Gaussian(normal) distribution. And the underlying assumptions of the linear models built are valid.

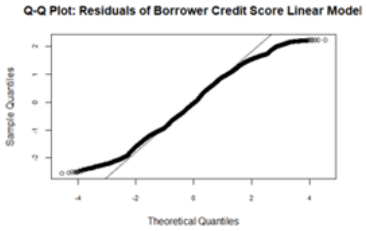
The following plots for residuals if each of the linear models discussed above, (for the dependent Original Interest Rate variable with each of the independent variables (7 numeric and 3 categorical and 2 nominal), follow approximate Gaussian distributions.

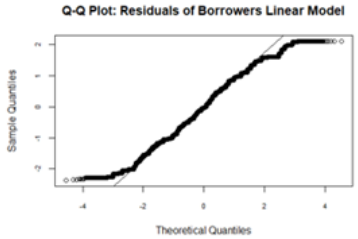


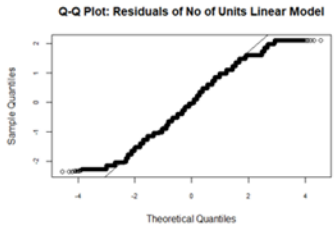


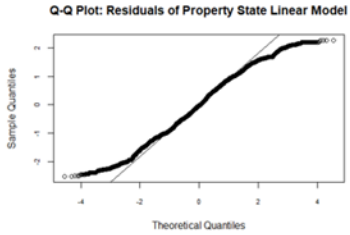


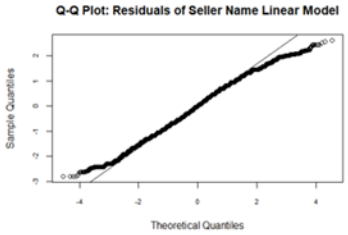


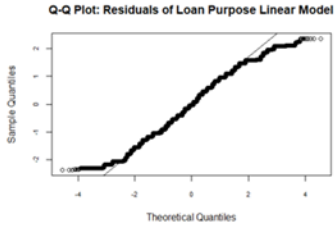


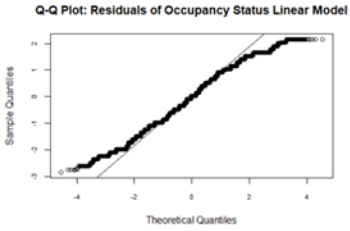


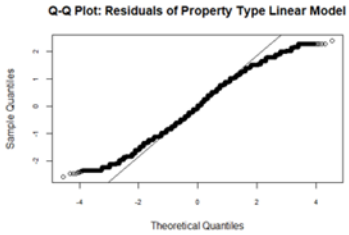












Is linear regression applicable to our problem?

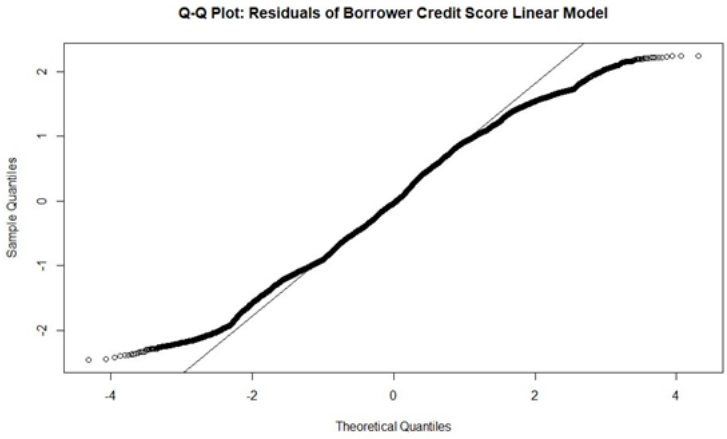
When we are using the lm() function in R, R finds the values of intercept and coefficient that best minimize the squared error. However, the plotting of the residuals indicates whether or not they follow a Guassian distribution, which is the validation of the model. We find that the residual distributions for all the models only follow approximate normal distributions, not exact. Based on the values we found for coefficients for different independent variables(slope) in our dataset and the statistical significance of the slope, as well as the R squared values for the models themselves, we can say that linear regression may not be the best choice for our problem. However, Linear regression has revealed the predictive power of each independent variable and the different levels of categorical and nominal variables with respect to the dependent variable. Because of the results, we are able to identify variables without predictive power and as we move forward in our analysis, we will probably be only using those variables that have better predictive power.

Using the trained model to predict our testing data and showing results together with confidence and prediction bands. Reporting prediction accuracy using (1) the correlation between the predicted and real values and (2) the mean square error between the two.

Using the lmCS model trained on training dataset to predict outcome variable values and comparing them with the actual data in the ‘Test’ dataset:

```
> summary(lmCS)

Call:
lm(formula = Original.Interest.Rate ~ Borrower.Credit.Score.at.Origination,
    data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-2.54853 -0.58972 -0.03241  0.63154  2.23765

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
(Intercept)                       7.693e+00  3.474e-02  221.44   <2e-16 ***
Borrower.Credit.Score.at.Origination -2.211e-03  4.601e-05  -48.05   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8281 on 186265 degrees of freedom
Multiple R-squared:  0.01224,   Adjusted R-squared:  0.01224
F-statistic:  2309 on 1 and 186265 DF,  p-value: < 2.2e-16
```

The distribution of the residuals for this model is approximately Gaussian, as is evident from this plot.



The Correlation coefficient and MSE for this model against predicted values are:

```
> predictionCS <-predict(lmCS, newdata = test)
> cor(predictionCS, test$Original.Interest.Rate)
[1] 0.1090926
> mse <- mean((predictionCS-test$Original.Interest.Rate)^2)
> mse
[1] 0.6834206
>
```

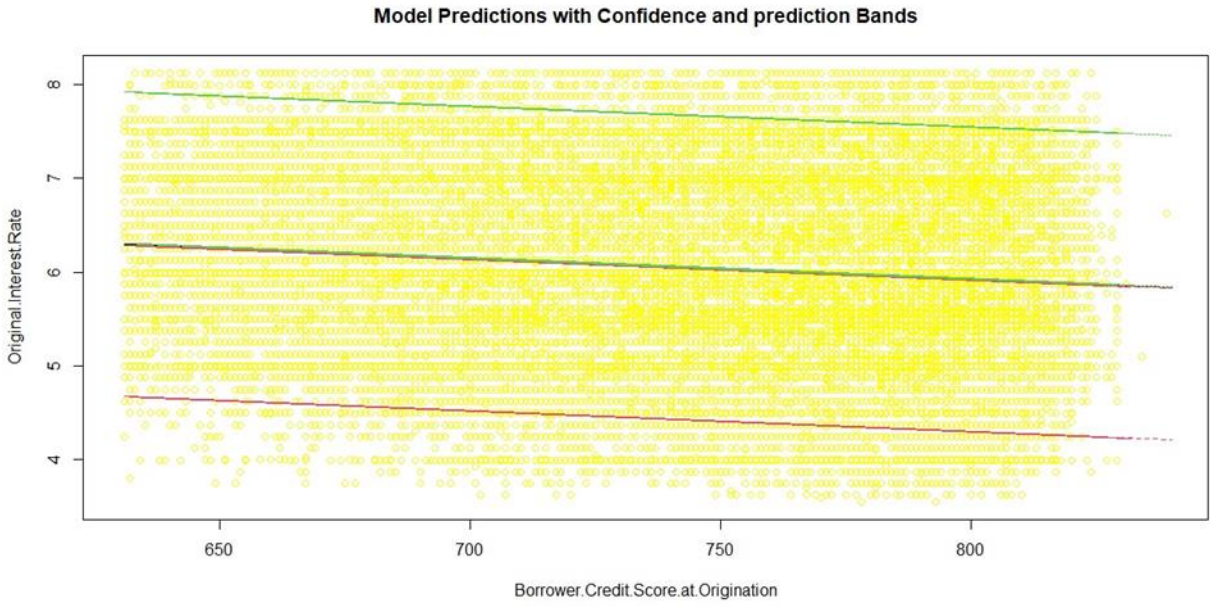
Reporting accuracy of the model trained using entire Training data and testing on Test data:

Validity of the model and Correlation between predicted and actual values: The distribution of the residuals is an approximate Gaussian distribution. The Multiple R-Squared of our linear model is 0.01224 and F-statistic is 2309 on 1 and 186265 DF, with a p-value <2.2e-16. The large F-statistic and the associated low p-value indicate that the results are not by chance. The R-squared value indicates the 'goodness of fit' of the model. Ideally if the value of R-squared is closer to 1, the model is considered a good fit. *We intend to*

improve our R-squared value by eliminating the variables without much predictive power, after multivariate analysis. For now, we proceeded to predict values using test data and find the correlation between predicted and actual values. The correlation between the predicted and actual values of the dependent variable interest rate is 0.1090926 which is poor, indicating a weak positive linear relationship between predicted values of the Interest Rate variable and the actual values from Test data.

The Mean squared error value represents the average squared difference between the predicted values and the actual values and therefore, lower MSE values indicate better convergence between predicted and actual values. In the present model, the MSE is 0.6834206, indicating that on average our model has relatively small, squared errors.

Plot of the model with Prediction and Confidence bands.



Interpretation:

A confidence band in regression analysis shows that the true regression line lies within these bands with a certain level of confidence. Here the 95% CI bands are shown. Prediction bands, on the other hand, account for variability in the data in addition to the uncertainty in the estimated regression coefficients. The Prediction bands indicate the space where the predicted values will lie. The confidence bands are just around the linear regression line in black in the plot. It is evident that the Prediction bands cover a wider area than the confidence bands from the above plot.

b . Multivariate regressions

Exploring whether considering combinations of independent features improves the prediction results. Evaluating different combinations of features as applicable and reporting those that improve the results shown in question (a)

When we performed multivariate analysis, we noticed that combinations of some IVs improved both the coefficients of variables and the R-squared values whereas some did not. We summarize hereunder, our findings.

S.No	Linear Model on training dataset Built with a combination of	Coefficients of IVs	p-value	Multiple R-Squared	Improvement over previous linear models	Is it a Predictive Feature
1	UPB + Credit Score	-6.690e-07 -2.008e-03	1.335e-08	0.02538	No. The magnitudes (absolute values) of the coefficients in the combination model are actually smaller. The combination model R-squared values are slightly better than the values of the individual model but nowhere closer to 1.	No. Statistically significant but small coefficients and R-squared values indicating the model is not having an overall 'goodness of fit'. *Independent models had better values for coefficients for these IVS, but this model has better R-squared*
2	Seller Name + Occupancy Status	Small Coefficient values <1 for all sellers except one. Occupancy status coefficients < 1	<2e-16 for most coeff values	0.1528	Yes. The coefficients are slightly larger than the individual models but not significantly. Linear Models built with seller Name alone had a R-squared value of 0.1144 and with Occupancy status alone had 0.0448. This combination model is a better fit but the size of coefficients is still very small	No. Statistically significant but small coefficients and R-squared values indicating the model is not having an overall 'goodness of fit'. *Independent models had better values for coefficients for these IVS. But this model has better R-squared*

S.No	Linear Model on training dataset Built with a combination of	Coefficients of IVs	p-value	Multiple R-Squared	Improvement over previous linear models	Is it a Predictive Feature
3	Seller Name + Property Type	Small Coefficient values <1 for all sellers except one. Property Type coefficients < 0.01	<2e-16 most coeff values	0.1144	Model built with seller name variable alone had the same R-squared value and Property type alone had 0.02587 R-squared value. The combination model is at the same level of goodness of fit as that of Seller Name model	No.Statistically significant but small coefficients and R-squared values indicating the model is not having an overall 'goodness of fit'.*Independent models had better values for coefficients for these IVS, but this model has better R-squared*
4	UPB+Credit Score+ Seller Name + Occupancy Status (4)	<0.1	<2e-16 most coeff values	0.1703	This model has better R-squared value than individual models but small coefficients	No. Statistically significant but small coefficients and R-squared values indicate the model is not having an overall 'goodness of fit'. *Independent models had better values for coefficients for these IVS, but this model has better R-squared.*
5	UPB+Credit Score+ Seller Name + Occupancy Status + Property state + Property Type (6)	All coefficients in decimals	<2e-16 most coeff values	0.1838	The coefficients are all very small mostly <1 but statistically significant. Of all the models, this combination model has the best R-squared value.. But it is nowhere closer to 1 to surmise that this is the best 'goodness of fit'	No. Statistically significant but small coefficients and R-squared values indicating the model is not having an overall 'goodness of fit'. *Independent models had better values for coefficients for these IVS, but this model has better R-squared*

Conclusion about the most predictive features in our training data:

Revisiting our IVs.

At this point, we understood that our independent variables have statistically significant but magnitude wise insignificant predictive power (because of weak R-squared values of the models) on our dependent variable. This was proven by testing both individual linear models and combination linear models for predictive power. We decided to formulate answers to our research questions using the research done thus far to determine if we needed to infuse new IVs to perform meaningful analysis.

1. Can we accurately predict the 'Interest Rate' applied to a mortgage loan by a lending institution, using criterion such as the Quantum of loan, Loan-To-Value ratio, Debt-To-Income ratio, Loan Purpose, Number of Borrowers and Credit Score of the Borrower/s, using the Fannie Mae Acquisition and Performance 2022Q4 dataset?

Answer : As per linear regression analysis, the models built around variables listed in this research question and few other variables like Seller Name, Occupancy status, Property State and Property type from the existing dataset have very small coefficients and R-squared values. These models do not offer sufficient predictive power to accurately predict our outcome variable 'Interest Rate'. So the answer to this research question is that we cannot determine the answer to this research question with the Fannie Mae dataset alone.

2. Do all the borrower criteria such as the Quantum of loan, Loan-To-Value ratio, Debt-To-Income ratio, Loan Purpose, Number of Borrowers and Credit Score of the Borrower/s have equal influence over 'Interest Rate'? If not, which criteria have more effect on Interest Rate?

Answer: As per linear regression analysis, Quantum of Loan (UPB) and Borrower credit score have statistically significant but mild inverse effect on Interest rate variable, because of very small coefficients and R-squared values of the linear models. The other variables have a much smaller, unequal positive impact on the outcome variable 'Original Interest Rate'. The only IV in the dataset that has a slightly better impact is the Seller Name variable where the coefficients of different levels in a linear regression model range from (0.23 to 1.15) with statistical significance. But the R-squared of the model is 0.11 which is insignificant. So the answer to this research question is that none of the variables in the existing dataset have a good predictive power on the Original interest rate. Therefore we intend to infuse new variables from datasets mentioned below to find an answer to this research question.

3. Original: Do all lending institutions (presumed to be the 'Seller' in the dataset), evaluate the criterion variables similarly and fix the interest rate? (This question was revised because Milestone-1 analysis revealed that most values fall under the 'unknown' category named as 'other'. This aspect may or may not provide for accuracy in training the models.)

Revised: *'Can we accurately predict the class of Occupancy Status in the Loan dataset, based on the IVs, Original interest rate, bondRate, fedFundsRate, Borrower Credit Score and Original UPB (from the revised dataset)?*

We infused 3 variables namely 'bondRate', 'fedFundsRate' and 'benchMarkMortgageRate' from the datasets sourced from the following sites into the Final_FM_AP_FM2022Q4.csv dataset. (We added 3 independent variables and retained the earlier 12 independent focus variables along with the dependent variable.)

1.We downloaded 10-year Treasury Bond rates from the following location to infuse the 'bondRate' variable into our dataset., *https://www.macrotrends.net/2016/10-year-treasury-bond-rate-yield-chart*

2. To infuse 'benchMarkMortgageRate' variable into our dataset, we downloaded this chart from :https://www.macrotrends.net/2604/30-year-fixed-mortgage-rate-chart This is a benchmark chart that shows the average rates of 30 year fixed rate mortgages in the United States since 1971.

3.Fed funds Rate historical chart from: https://www.macrotrends.net/2015/fed-funds-rate-historical-chart The 'Fed funds rate' is the interest rate at which lending institutions lend or borrow funds from other institutions overnight, on an uncollateralized basis. We absorbed it with the view that it represents the 'cost of funds' for lending institutions.

We matched these variables on the 'date' column (common name in all three datasets) from these datasets and the Loan Origination Date column in our dataset. We renamed the 'value' variable (common name in all three datasets) in these datasets to reflect the name of the rate that is described in the details above. We then proceeded to filter out the variables that have not revealed any predictive power so far and rows with Null values. To save time, we used excel to perform vlookup on the Loan Origination Date column to add these 3 variables and retain variables that we have used so far in our research to verify if we can develop models with better predictive power.

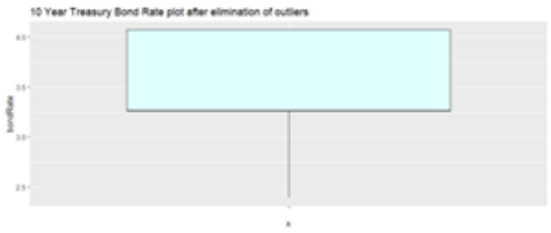
We now had to remove outliers, Nulls and NA values for all the 16 variables all over again!

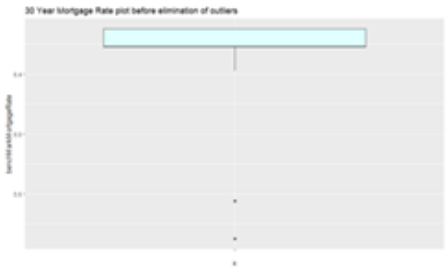
The dimensions of this dataset with various levels of transformations are captured in this table.

Name of the dataset	No of observations	No of variables	Action Performed on this dataset
New_Variables_FM_AP_R_2022Q4.csv is the file created after adding 3 new independent variables to our Final Dataset matching on 'Loan Origination Date'	271368	16	Checking and removing Nulls from Borrower Credit Score and DTI variables to get 270993 observations of 16 variables. Glimpse data reveals 2 of the new variables, bondRate and fedFundsRate have 'chr' data type. View data reveals #NA values

Name of the dataset	No of observations	No of variables	Action Performed on this dataset
data	270993	16	Change datatype from 'chr' to 'dbl' for bondRate and fedFundsRate variables. Removed #NA values to get 184096 obs of 16 variables
NoNullsDataAsFactors.csv	184096	16	Removing outliers from Original Interest Rate variable
data	183582	16	Removing outliers from Original UPB variable
data	181378	16	Removing outliers from LTV variable
data	176521	16	Removing outliers from CLTV variable
data	173966	16	Removing outliers from No of Borrowers variable
data	173625	16	Removing outliers from No of Units variable
data	170960	16	Removing outliers from DTI variable
data	169838	16	Removing outliers from Credit Score variable
data	168760	16	Removed outliers from bondRate variable
data	168312	16	Remove outliers from fedFundsRate variable
data	166778	16	Remove outliers from the benchMarkMortgageRate variable to get 156848 observations of 16 variables. Named this dataset New_FM2022Q4.csv
New_FM2022Q4.csv	144130	16	These are the dimensions of our final dataset. This will be split into Training and Test datasets for analysis

We are only reproducing the box plots for the new variables here, with positions before and after removal of outliers. However, we have removed outliers for the entire set of variables in this new dataset and removed nulls and NA values.





After removal of outliers, our dataset had 144130 observations of 16 variables which we named "New_FM2022Q4.csv". We proceeded to split this dataset into test and training datasets by selecting every 4th row for the test dataset and remaining rows for training dataset using R. We named them "Test_FM2022Q4.csv" and "Training_FM2022Q4.csv".

The dimensions of these datasets are:

```
> data <- read.csv("New_FM2022Q4.csv")
> dim(data)
[1] 144130     16
> data <- read.csv("New_FM2022Q4.csv")
> testIndexNos <- which(1:nrow(data)%%4==0)
> #Write test data using the randomly selected row nos
> testData <-data[testIndexNos, ]
> write.csv(testData, file = "Test_FM2022Q4.csv", row.names = FALSE)
> data <- read.csv("Test_FM2022Q4.csv")
> dim(data)
[1] 36032    16
> data <- read.csv("New_FM2022Q4.csv")
> trainData <- data[-testIndexNos, ]
> write.csv(trainData, file = "Training_FM2022Q4.csv", row.names = FALSE)
> data <- read.csv("Training_FM2022Q4.csv")
> dim(data)
[1] 108098     16
```

Reporting Predictive power of Linear models built with each Independent variable and dependent variable.

Using the New Training dataset, we built linear models with each of the independent variables and combinations of independent variables to ascertain if there is change in the predictive power of the models.

Summary of the Coefficients and R-squared values of Linear models built with each of the IV and DV from the new dataset.

S.No	Linear Model on training dataset Built between	Coefficient of IV (Slope)	p-value	Multiple R-Squared	Remarks
1	UPB & Original Interest Rate	-7.559e-07	<2e-16, statistically highly significant	0.01512	Improvement in decimal value in magnitude of coefficient and R-squared value over previous model with the same IV, but not significant overall improvement in predictive power of the model. This model has No significant predictive power
2	LTV. & Original Interest Rate	0.0046816	<2e-16, statistically highly significant	0.007157	-do-
3	CLTV. & Original Interest Rate	0.0046947	<2e-16, statistically highly significant	0.007288	-do-
4	Number of Borrowers & Original Interest Rate	0.041812	<2e-16, statistically highly significant	0.0006239	-do-
5	DTI. & Original Interest Rate	0.0067850	<2e-16, statistically highly significant	0.004573	-do-
6	Borrower Credit Score & Original Interest Rate	-2.090e-03	<2e-16, statistically highly significant	0.009749	Coefficient magnitude and R-squared values are slightly smaller than previous models with the same variables. The model has No significant predictive power

S.No	Linear Model on training dataset Built between	Coefficient of IV (Slope)	p-value	Multiple R-Squared	Remarks
7	Number of Units & Original Interest Rate	No. After removal of outliers, this variable is left with only 1 value.			Model cannot be trained. Singularity in in IV
8	Property State as Factor & Original Interest Rate	Maximum value is 0.53. 90% of the coefficients <0.2	P-values range from 0.03 to 0.84. Not highly significant	0.0136	Slightly smaller coefficients and larger p-values with insignificantly small R-squared value than previous model with same IV. The model has No significant predictive power
9	Seller Name as Factor & Original Interest Rate	This variable has 25 levels, largest coefficient is 1.11, smallest is 0.02	all < 2e-16 statistically highly significant	0.1211	Slightly better magnitudes of coefficients and R-squared than the previous model. This IV is a weak predictive feature.
10	Loan Purpose as Factor & Original Interest Rate	-0.183567 for Refinance other levels are still smaller.	<2e-16, statistically highly significant	0.002478	Very small R-squared and coefficient values

S.No	Linear Model on training dataset Built between	Coefficient of IV (Slope)	p-value	Multiple R-Squared	Remarks
11	Occupancy Status as Factor & Original Interest Rate	-0.626278 for Primary residence level, other levels much smaller	<2e-16, statistically highly significant	0.04321	-do-
12	Property Type as Factor & Original Interest Rate	Types CP, MH,PU have statistically significant coefficients, smaller than previous model	<2e-16, statistically highly significant	0.02572	-do-
13	*bondRate & Original Interest Rate*	*1.357337*	*<2e-16, statistically highly significant*	*0.3377*	*F-statistic 5.512e+04 on 1 and a better magnitude of coefficient with a statistically significant p-value than other independent variables tested so far. Model has significant predictive power*
14	*fedFundsRate & Original Interest Rate*	*1.043833*	*<2e-16, statistically highly significant*	*0.3515*	*F-statistic: 5.859e+04 on 1 1 and a better magnitude of coefficient with a statistically significant p-value and R-squared values than previously tested independent variable. Model has good predictive power*

S.No	Linear Model on training dataset Built between	Coefficient of IV (Slope)	p-value	Multiple R-Squared	Remarks
15	*benchMarkMortgageRate & Original Interest Rate*	*-5.60808*	*<2e-16, statistically highly significant*	*0.3212*	*F-statistic: 5.116e+04 on 1 and a much greater magnitude of coefficient that varies inversely with the dependent variable. The P-value associated with the coefficient value is also highly significant. Model has high predictive power*

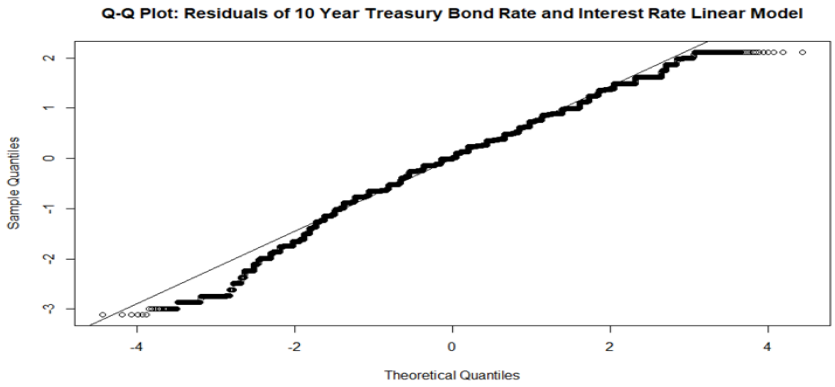
Clearly, the linear models trained on the new bondRate, fedFundsRate and benchMarkMortgageRate variables have coefficients of better magnitude with statistically highly significant p-values and the models have better mean R-squared values. The other models show lower levels of significance than those tested with the previous dataset. Also, we have used those models to predict using testing data and reported prediction accuracy using correlation and mean squared error along with prediction and confidence bands. We are now proceeding to do the same for the linear models built with the new variables.

Validating the linear models with new variables : If the distributions of Residuals of the models follow a Gaussian distribution, we can say that the underlying assumption for the model is valid and therefore the model itself is valid. For each of the linear model with predictive power as described previously we proceeded to validate, predict using test data and report accuracy with correlation, mean square error & plotting with prediction and correlation bands as under:

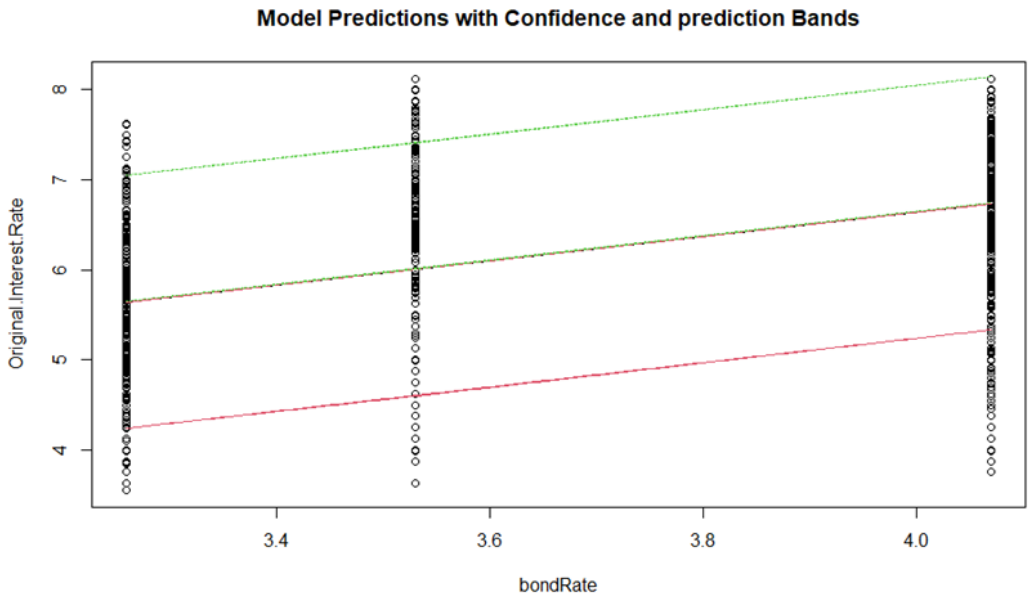
1. Predicting with Linear Model bondRate IV and Original Interest Rate DV: This model has a prediction correlation coefficient of 0.5832867 & MSE of 0.5117943 as shown in the R command output below. The correlation coefficient indicates the correlation between predicted and actual values. A higher value indicates better correlation. Mean squared error on the other hand should be lower suggesting that the model's predictions are closer to actual values, as it measures the average squared difference between actual values and predicted values of the DV . The correlation coefficient of this linear model identifies it as moderately strong (>0.3 and <0.7) in predicting the outcome variable.

```
> train <- read.csv("Training_FM2022Q4.csv")
> test <- read.csv("Test_FM2022Q4.csv")
> prediction <- predict(lmBR, newdata = test)
> View(prediction)
> #Checking for correlation between predicted and actual values of dependent variable
> cor(prediction, test$Original.Interest.Rate)
[1] 0.5832867
> mse <- mean((prediction-test$Original.Interest.Rate)^2)
> mse
[1] 0.5117943
>
```

Plotting residuals: Distribution of Residuals of the model follow approximately a normal distribution but not exactly. . Maybe a linear regression is not the best approximation we can get to the model for this dataset.



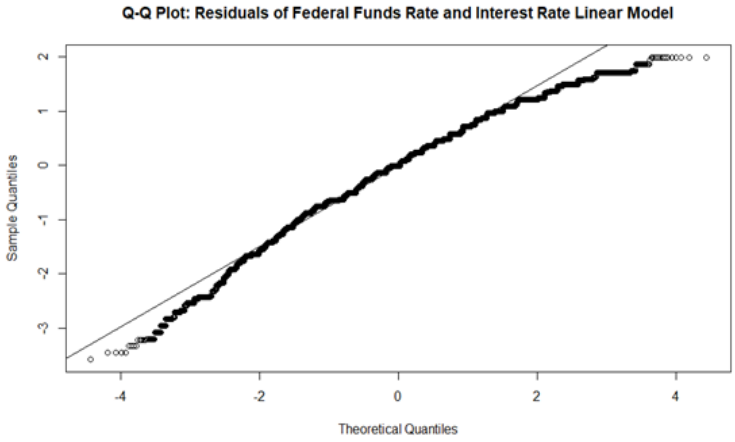
Plotting with prediction and correlation bands shows that all the predictions fall between 4.25% and 7% interest rate. The confidence bands in between them indicate that it can be said with 95% confidence level that the true regression line lies between these two bands



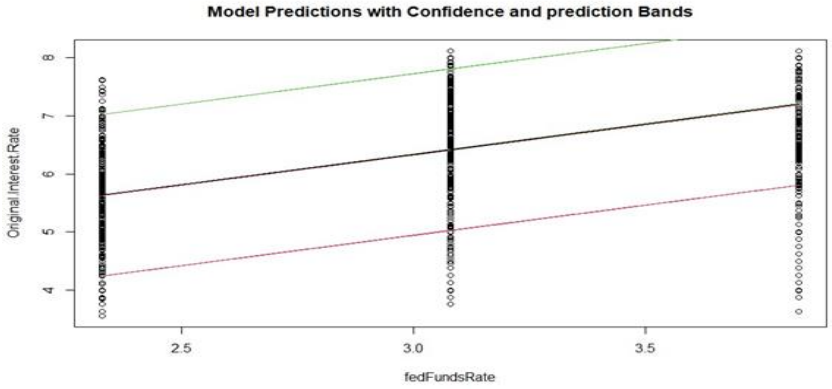
2. Predicting with Linear Model with fedFundsRate IV and Original Interest Rate DV: This model shows better prediction correlation coefficient indicating that for one unit increase in federal funds rate, there is 0.5974069 increase in Loan interest rate. Also the mean square error is smaller indicating that predicted values and actual values are closer to each other than they are for the 10 year treasury bond rate.

```
> #Using the fedFundsRate linear model to predict Interest rate  on Test data
> prediction <- predict(lmFFR, newdata = test)
> #Checking for correlation between predicted and actual values of dependent variable
> cor(prediction, test$Original.Interest.Rate)
[1] 0.5974069
> mse <- mean((prediction-test$Original.Interest.Rate)^2)
> mse
[1] 0.4988715
>
```

Plotting residuals: Distribution of Residuals of the model follows approximately a normal distribution but not exactly. Maybe a linear regression is not the best approximation we can get to the model for this dataset.



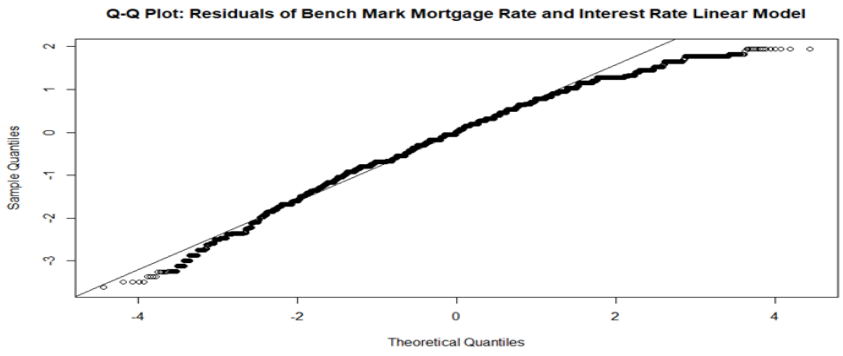
Reporting accuracy with confidence and prediction bands

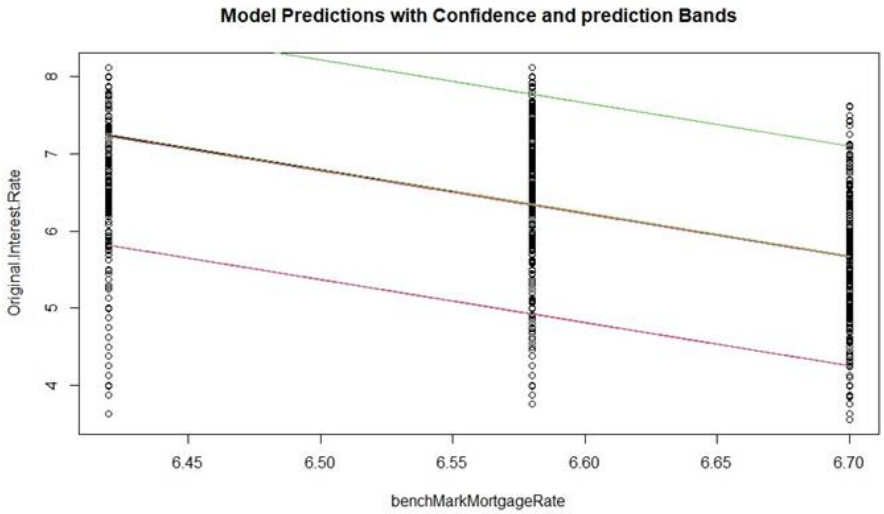


3. Predicting with Linear Model with benchMarkMortgageRate IV and Original Interest Rate DV: This model shows better prediction correlation coefficient indicating that for one unit increase in Benchmark Mortgage rate, there is 0.5713404 increase in Loan interest rate. Also, the mean square error is 0.5225029 indicating that predicted values and actual values are closer to each other. Therefore, this model has good predictive power.

```
> #Using the benchMarkMortgageRate linear model to predict Interest rate  on Test data
> prediction <- predict(lmBMMR, newdata = test)
> #Checking for correlation between predicted and actual values of dependent variable
> cor(prediction, test$Original.Interest.Rate)
[1] 0.5713404
> mse <- mean((prediction-test$Original.Interest.Rate)^2)
> mse
[1] 0.5225029
>
```

Distribution of Residuals of the model follow approximately a normal distribution but not exactly. . Maybe a linear regression is not the best approximation we can get to the model for this dataset.





Multivariate Regression analysis, summary: Clearly the multivariate models with the new variables have better predictive power. We reproduce the important statistics of these models.

MULTIVARIATE REGRESSION MODELS TRAINED WITH NEW TRAINING DATA SET (SORTED IN DESCENDING ORDER OF R-SQUARED VALUES)				
Name of Model (New dataset with 16 variables)	Coefficients of Ivs	P-value significant?	Multiple R-squared	Remarks
bondRate + fedFundsRate+ Borrower Credit Score + UPB+ Seller Name & Original Interest Rate	BR: 8.016e-01 FFR: 5.822e-01 CS: -2.014e-03 UPB: -5.679e-07 SN: 0.0068 to 0.68	Yes	0.4903	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
bondRate + fedFundsRate+ Seller Name & Original Interest Rate	BR: 0.8021608 FFR: 0.5789739 SN: 0.008 to 0.68	Yes	0.4716	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
bondRate + fedFundsRate+ Borrower Credit Score + UPB & Original Interest Rate	BR:7.826e-01 FFR: 6.502e-01 CS: 2.100e-03 UPB: -6.128e-07	Yes	0.4357	Two Coefficients have magnitude and significance. Other two are very small. R-squared can be better. Model can be taken up for further analysis
bondRate + fedFundsRate+benchMarkMortgageRate + Borrower Credit Score+ Origination UPB & Original Interest Rate	Indeterminate for one variable	Yes	0.4357	We cannot train the model because of multicollinearity between fedfundsrate and benchmarkMortgageRate variables

Name of Model (New dataset with 16 variables)	Coefficients of Ivs	P-value significant?	Multiple R-squared	Remarks
bondRate + fedFundsRate+benchMarkMortgageRate + Borrower Credit Score & Original Interest Rate	Indeterminate for one variable	Yes	0.4259	We cannot train the model because of multicollinearity between fedfundsrate and benchmarkMortgageRate variables
bondRate + fedFundsRate & Original Interest Rate	BR: 0.782748 FFR: 0.651628	Yes	0.4194	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
bondRate + Seller Name & Original Interest Rate	BR:1.293098 SN:ranging from 0.004 to 0.82	For 20/25 levelsYes	0.4147	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
fedFundsRate+benchMarkMortgageRate & Original Interest Rate	FFR: 5.72384 BMMR: 26.41776	Yes	0.4142	Coefficients have high magnitude and significance. R-squared can be better. Can be taken up for further analysis
bondRate + benchMarkMortgageRate & Original Interest Rate	BR: 0.883308 BMMR: -3.393897	Yes	0.4142	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis

Name of Model (New dataset with 16 variables)	Coefficients of Ivs	P-value significant?	Multiple R-squared	Remarks
fedFundsRate+ Seller Name & Original Interest Rate	FFR : 0.982703 SN: 0.026 to 0.672	For 20/25 levelsYes	0.4068	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
benchMarkMortgageRate+ Seller Name & Original Interest Rate	BMMR: -5.2414617 SN: 0.008 to 0.667	For 20/25 levelsYes	0.3784	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis

Dividing our dataset into training and testing sets and reporting prediction accuracy using (1) the correlation between the predicted and real values and (2) the mean square error between the two:

For the above Models incorporating new variables, we determined the correlation and mean squared error. Clearly, the Correlation coefficients of predicted and actual values and the mean squared errors are better for these models.

LINEAR MODELS TAKEN UP FOR PREDICTING ON TEST DATA AND REPORTING ACCURACY WITH CORRELATION AND MSE						
Name of Model (New dataset with 16 variables)	Coefficients of Ivs	P-value significant?	Multiple R-squared	corr	mse	Remarks
bondRate + fedFundsRate+ Borrower Credit Score + UPB+ Seller Name & Original Interest Rate	BR: 8.016e-01 FFR: 5.822e-01 CS: -2.014e-03 UPB: -5.679e-07 SN: 0.0068 to 0.68	Yes	0.4903	0.7051072	0.390059	Strong correlation between predicted and actual values of outcome variable interest rate, and low MSE indicate that the predicted and actual values converge better in this model

Name of Model (New dataset with 16 variables)	Coefficients of Ivs	P-value significant?	Multiple R-squared	corr	mse	Remarks
bondRate + fedFundsRate+ Seller Name & Original Interest Rate	BR: 0.8021608 FFR: 0.5789739 SN: 0.008 to 0.68	Yes	0.4716	0.6912357	0.405084	Same as above
bondRate + fedFundsRate+ Borrower Credit Score + UPB & Original Interest Rate	BR:7.826e-01 FFR: 6.502e-01 CS: 2.100e-03 UPB: -6.128e-07	Yes	0.4357	0.6653475	0.43233	Moderatelystrong correlation between predicted and actual values of outcome variable interest rate, and low MSE indicate that the predicted and actual values converge better in this model
bondRate + fedFundsRate & Original Interest Rate	BR: 0.782748 FFR: 0.651628	Yes	0.4194	0.6476303	0.450367	Same as above
bondRate + Seller Name & Original Interest Rate	BR:1.293098 SN:ranging from 0.004 to 0.82	For 20/25 levels Yes	0.4147	0.6469263	0.451069	Same as above
fedFundsRate+benchMarkMortgageRate & Original Interest Rate	FFR: 5.72384 BMMR: 26.41776	Yes	0.4142	0.6476303	0.450367	Same as above
bondRate + benchMarkMortgageRate & Original Interest Rate	BR: 0.883308 BMMR: -3.393897	Yes	0.4142	0.6476303	0.450367	Same as above

Name of Model (New dataset with 16 variables)	Coefficients of Ivs	P-value significant?	Multiple R-squared	corr	mse	Remarks
fedFundsRate+ Seller Name & Original Interest Rate	FFR : 0.982703 SN: 0.026 to 0.672	For 20/25 levels Yes	0.4068	0.642682	0.455321	Same as above
benchMarkMortgageRate+ Seller Name & Original Interest Rate	BMMR: -5.2414617 SN: 0.008 to 0.667	For 20/25 levels Yes	0.3784	0.6199392	0.477593	Same as above

c. Regularization

We may have too many features in our data and that is not helping our prediction model. Through regularization, we can eliminate some variables to improve our model. Regularization helps in preventing overfitting and improves generalization of linear models. There are Ridge and Lasso regularizations we can choose from with alpha parameter. When we use the library called glmnet in R, it produces another linear regression model, but with regularization.

Linear regression reveals that our variables "benchMarkMortgageRate" , "Loan Purpose" and "Number of Units" exhibit multicollinearity. Linear models could not be trained when using these variables that have perfect correlation with other independent variables such as fedFundsRate. Because of these collinearities we could not train a model on the entire dataset. We identified those variables using Pearson's correlation and decided to remove them from the dataset. We can then train a model on the entire dataset and then use regularization techniques to remove some variables and check the improvement.

After removing these 3 variables the dimension of our dataset is 144130 Obs of 13 variables. We then proceeded to split this dataset into training and test datasets for use in regularization.

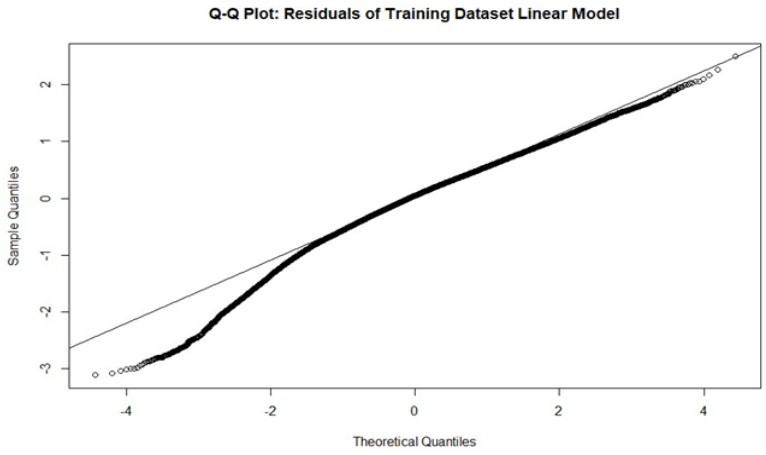
```
> dim(data)
[1] 144130     13
> #ENSURE RANDOMNESS IN THE  DATA FIRST
> #To ensure we get the same set of random nos every time so that our results will be same  We
use the set seed function
> set.seed(123)
> #CREATE RANDOM NOS USING nrow() to create nrows of random nos
> randomNos <- runif(nrow(data))
> #USE randomNos to SHUFFLE DATA,
> datar <- data[order(randomNos), ]
> train <- datar[1:108097, ] #75% of the dataset
> write.csv(train, file = "NC_Train_FM2022Q4.csv", row.names = FALSE)
> data <- read.csv("NC_Train_FM2022Q4.csv")
> dim(data)
[1] 108097     13
> test <- datar[108098:144130, ] #25% of the dataset
> write.csv(test, file= "NC_Test_FM2022Q4.csv", row.names = FALSE)
> data <- read.csv("NC_Test_FM2022Q4.csv")
> dim(data)
[1] 36033    13
```

After this step, we trained a linear model on the entire training dataset to understand the coefficients of IVs and R-squared values of the model. The R-squared of the model is 0.5508 and F-statistic is 1455 on 91 and 108005 DF with p-value < 2.2e-16. Only a few levels of the categorical variable Property state have significant p-values, whereas many levels of the seller name variable have significant p-values associated with the coefficients. Other than bondRate and fedFundsRate which have 0.8022 and 0.5771 magnitude coefficients, other numeric variables have much smaller coefficient magnitudes that have significant p-values for the coefficient values.

The distribution of residuals of the above linear model, does not exactly follow a Gaussian distribution. However the correlation and MSE values are the strongest among all models trained so far.

Cor between predicted and actual outcome variable values using test data r = 0.7429967

And mean square error of the difference between predicted and actual values mse = 0.3469728

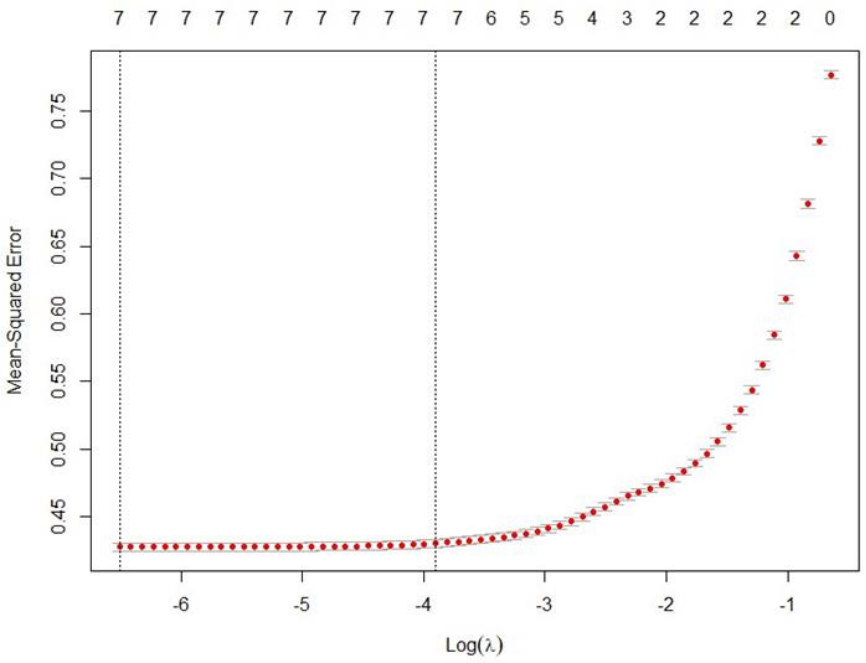



```
> cor(prediction, test$Original.Interest.Rate)
[1] 0.7429967
> mse <- mean((prediction-test$Original.Interest.Rate)^2)
> mse
[1] 0.3469728
>
```

Now we proceed to use Regularization by removing columns that have non-numeric values because regularization only works for numeric values. Therefore, we remove variables ‘Seller Name’, “Property Type”, “Occupancy Status” and “Property State” variables and fit a model, using the “glmnet’ function in R. We also plotted the log(lambda) values and the mean-squared error of the model and listed the coefficients for understanding which are not helping the model.

```
> cv.fit <- cv.glmnet(as.matrix(train[,c(-1,-2,-11,-12,-13)]),
+                                as.vector(train[,2]),
+                                alpha = 1)
> cv.fit

Call:  cv.glmnet(x = as.matrix(train[, c(-1, -2, -11, -12, -13)]), y = as.vector(train[,
2]), alpha = 1)

Measure: Mean-Squared Error

      Lambda Index Measure       SE Nonzero
min 0.001493    64  0.4275 0.003048       7
1se 0.020203    36  0.4303 0.002961       7
> coef(cv.fit)
9 x 1 sparse Matrix of class "dgCMatrix"
                                                s1
(Intercept)                                2.637519e+00
Original.UPB                              -6.245435e-07
bondRate                                   7.463266e-01
fedFundsRate                               6.279284e-01
Original.Loan.to.Value.Ratio..LTV.         .
Original.Combined.Loan.to.Value.Ratio..CLTV. 3.953144e-03
Number.of.Borrowers                        1.642157e-02
Debt.To.Income..DTI.                       1.442769e-03
Borrower.Credit.Score.at.Origination      -1.536119e-03
```

Plotting Lambda and Mean square error values:



The cv.glmnet function performed cross-validation to select the optimal lambda.min value 0.001493. We used Lasso (L1) regularization by setting alpha=1. While making predictions, we provided the lambda.min value in the ‘s’ argument so that predictions are made using that value which is the optimal value.

```
> predictions <- predict(cv.fit, s= 0.001493113,
+                        newx=as.matrix(test[,c(-1,-2,-11,-12,-13)]))
> cor(predictions,as.vector(test[,2]))
          [,1]
s1 0.6703945
> mse <- mean((predictions-test$Original.Interest.Rate)^2)
> mse
[1] 0.4264433
```

Interpretation:

1. Coefficient values: For predicting the output variable, the most important features with highest coefficients are the 'bondRate' variable with the highest magnitude of 0.7463266 and next the 'fedFundsRate' variable with a magnitude of 0.6279284. Other variables have lesser magnitudes of coefficients indicating their predictive power in predicting the outcome variable.

2. Finally with the coefficient expressed as '.' , the LTV variable does not have any effect on the outcome variable according to regularization

3. *Correlation value of 0.6703945 between predicted and output values suggests that this model is not an improvement over the model without regularization.*

4. *The MSE value of 0.4264433 is also higher than the previous model without regularization. The MSE values indicate that the predicted and output values are not converging as well as in the previous model.*

d . Repeating experiments a-c multiple times with randomly selected training and testing datasets

We find that every aspect of analysis in Milestone-2 is designed to reveal answers to our research questions. Towards that end, for this part of the assignment, the question we are trying to answer is whether running linear Regression, multivariate regression and regularization with different training and testing sets will result in differences in Coefficient values or better "goodness of fit' represented by better R-squared values. Will the correlation and MSE values for predictions differ for different runs of training and test datasets? To attempt this question, we summarize the results from the training results of linear and multivariate analyses done so far. From this summary, we know which of our variables exhibit multicollinearity because of which we cannot train linear models using them. We also know which variables offer better predictive power and which have none. This will help us identify which linear models to replicate with the randomly selected datasets for comparison of results that will help us get closer to finding answers to our Research Questions. Therefore, based on the summary,

1. We are proceeding to create multiple training and test datasets,
2. Replicate results of the most predictive models from the list below,
3. Summarize the results and analyze them.

SUMMARY OF ORIGINAL REGRESSION MODELS (Multivariate colored in Blue) TRAINED ON ORIGINAL TRAINING DATASET (SORTED IN DESCENDING ORDER OF R-SQUARED VALUES				
Name of Model (New dataset with 16 variables)	***Coefficients of Ivs***	***P-val significant?***	***Multiple R-squared***	***Remarks***
bondRate + fedFundsRate+ Borrower Credit Score + UPB+ Seller Name & Original Interest Rate	BR: 8.016e-01 FFR: 5.822e-01 CS: -2.014e-03 UPB: -5.679e-07 SN: 0.0068 to 0.68	Yes	0.4903	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
bondRate + fedFundsRate+ Seller Name & Original Interest Rate	BR: 0.8021608 FFR: 0.5789739 SN: 0.008 to 0.68	Yes	0.4716	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
bondRate + fedFundsRate+ Borrower Credit Score + UPB & Original Interest Rate	BR:7.826e-01 FFR: 6.502e-01 CS: 2.100e-03 UPB: -6.128e-07	Yes	0.4357	Two Coefficients have magnitude and significance. Other two are very small. R-squared can be better. Model can be taken up for further analysis
bondRate + fedFundsRate+benchMarkMortgageRate + Borrower Credit Score+ Origination UPB & Original Interest Rate	Indeterminate for one variable	Yes	0.4357	We cannot train the model because of multicollinearity between fedfundsrate and benchmarkMortgageRate variables
bondRate + fedFundsRate+benchMarkMortgageRate + Borrower Credit Score & Original Interest Rate	Indeterminate for one variable	Yes	0.4259	We cannot train the model because of multicollinearity between fedfundsrate and benchmarkMortgageRate variables

Name of Model (New dataset with 16 variables)	***Coefficients of Ivs***	***P-val significant?***	***Multiple R-squared***	***Remarks***
bondRate + fedFundsRate & Original Interest Rate	BR: 0.782748 FFR: 0.651628	Yes	0.4194	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
bondRate + Seller Name & Original Interest Rate	BR:1.293098 SN:ranging from 0.004 to 0.82	For 20/25 levelsYes	0.4147	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
fedFundsRate+benchMarkMortgageRate & Original Interest Rate	FFR: 5.72384 BMMR: 26.41776	Yes	0.4142	Coefficients have high magnitude and significance. R-squared can be better. Can be taken up for further analysis
bondRate + benchMarkMortgageRate & Original Interest Rate	BR: 0.883308 BMMR: -3.393897	Yes	0.4142	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
fedFundsRate+ Seller Name & Original Interest Rate	FFR : 0.982703 SN: 0.026 to 0.672	For 20/25 levelsYes	0.4068	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
benchMarkMortgageRate+ Seller Name & Original Interest Rate	BMMR: -5.2414617 SN: 0.008 to 0.667	For 20/25 levelsYes	0.3784	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
fedFundsRate & Original Interest Rate	1.043833	Yes	0.3515	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis

Name of Model (New dataset with 16 variables)	***Coefficients of Ivs***	***P-val significant?***	***Multiple R-squared***	***Remarks***
bondRate & Original Interest Rate	1.357337	Yes	0.3377	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
benchMarkMortgageRate & Original Interest Rate	-5.60808	Yes	0.3212	Coefficients have magnitude and significance. R-squared can be better. Can be taken up for further analysis
Seller Name as Factor & Original Interest Rate	Max magnitude for 1 seller, 1.11/ 5/25 levels not significant coeffsc	Yes	0.1211	very small R-squared. Model needs improvement
Occupancy Status as Factor & Original Interest Rate	-0.626278	Yes	0.04321	very small R-squared. Model needs improvement
Property Type as Factor & Original Interest Rate	Indeterminate for one class of the variable	Yes	0.02572	We cannot train the model
UPB & Original Interest Rate	-7.56E-07	Yes	0.01512	very small R-squared. Model needs improvement
Property State as Factor & Original Interest Rate	0.53	No	0.0136	very small R-squared. Model needs improvement
Borrower Credit Score & Original Interest Rate	-2.09E-03	Yes	0.00975	very small R-squared. Model needs improvement
CLTV. & Original Interest Rate	0.0046947	Yes	0.00729	very small R-squared. Model needs improvement
LTV. & Original Interest Rate	0.0046816	Yes	0.00716	very small R-squared. Model needs improvement
DTI. & Original Interest Rate	0.006785	Yes	0.00457	very small R-squared. Model needs improvement

Name of Model (New dataset with 16 variables)	***Coefficients of Ivs***	***P-val significant?***	***Multiple R-squared***	***Remarks***
Loan Purpose as Factor & Original Interest Rate	-0.183567	Yes	0.00248	very small R-squared. Model needs improvement
Number of Borrowers & Original Interest Rate	0.041812	Yes	0.00062	very small R-squared. Model needs improvement
Number of Units & Original Interest Rate	Singularities, No regression line	No	0	We cannot train the model

List of non-multicollinear variables in our dataset named 'NC_FM2022Q4.csv" which we are using to create Training and Test sets.

LIST OF NON-MULTICOLLINEAR VARIABLES IN OUR DATASET			
S.No	Variable Name	Data type	Description
1	*Original.Interest.Rate*	*numeric*	*The original interest rate on a mortgage loan as identified in the original mortgage note. It is a numeric continuous variable. This was the outcome variable we used to train models for linear regression, and Random Forests*
2	Original.UPB	numeric	The dollar amount of the loan as stated on the note at the time the loan was originated.
3	Original.Loan.to.Value.Ratio..LTV.	numeric	The ratio, expressed as a percentage, obtained by dividing the amount of the loan at origination by the value of the Property
4	Original.Combined.Loan.to.Value.Ratio..CLTV.	numeric	The ratio, expressed as a percentage, obtained by dividing the amount of all known outstanding loans at origination by the value of the property.
5	Number.of.Borrowers	numeric	The number of individuals obligated to repay the mortgage loan.

S.No	Variable Name	Data type	Description
6	Debt.To.Income..DTI.	numeric	The ratio obtained by dividing the total monthly debt expense by the total monthly income of the borrower at the time the loan was originated.
7	Borrower.Credit.Score.at.Origination	numeric	A numerical value used by the financial services industry to evaluate the quality of borrower's credit. Credit scores are typically based on a proprietary statistical model that is developed for use by credit data repositories. These credit repositories apply the model to borrower credit information to arrive at a credit score. When this term is used by Fannie Mae, it is referring to the "Classic" FICO score developed by Fair Isaac Corporation.
8	Property.State	chr	Property Unpaid Balance. This represents the quantum of loan sanctioned to the Bprrower
9	Seller.Name	chr	The name of the entity that delivered the mortgage loan to Fannie Mae.
10	*Occupancy.Status*	*chr*	*A categorical variable with 3 classes namely "P", "S" and "I". The classification describes the property occupancy status at the time the loan was originated.This variable describes whether the property being purchased with the loan, is the Primary residence of the borrower, Secondary Home or for investment. This is the variable we used as outcome variable for training Logistic Regression, Naïve Bayes and Decision Tree models.*
11	Property.Type	chr	An indicator that denotes whether the property type secured by the mortgage loan is a condominium, co-operative, planned urban development (PUD), manufactured home, or single-family home.
12	bondRate	numeric	10- year Treasury Bond Yield Rate
13	fedFundsRate	numeric	The 'Fed funds rate' is the interest rate at which lending institutions lend or borrow funds from other institutions overnight, on an uncollateralized basis

We proceeded to create multiple randomly selected datasets without setting seed, from our parent dataset with non-collinear variables:

```
> ########################
> # TESTING LINEAR REGRESSION, MULTIVARIATE REGRESSION AND REGULARIZATION WITH DIFFERENT TRAINING AND TESTING DATASETS
> data <- read.csv("NC_FM2022Q4.csv")
> numofsets <- 3
> train_datasets <- list()
> test_datasets <- list()
> for (i in 1:numofsets) {
+
+   sample_size <- floor(0.75 * nrow(data))  # 75% for training
+   train_indices <- sample(1:nrow(data), size = sample_size, replace = FALSE)
+
+   train_data <- data[train_indices, ]
+   test_data <- data[-train_indices, ]
+
+   # Storing the datasets in the lists
+   train_datasets[[i]] <- train_data
+   test_datasets[[i]] <- test_data
+
+   # Writing the datasets to CSV files
+   train_file <- paste0("train_data_", i, ".csv")
+   test_file <- paste0("test_data_", i, ".csv")
+   write.csv(train_data, file = train_file, row.names = FALSE)
+   write.csv(test_data, file = test_file, row.names = FALSE)
+ }
> train_data_1 <- read.csv("train_data_1.csv")
> dim(train_data_1)
[1] 108097     13
>
> train_data_2 <- read.csv("train_data_2.csv")
> dim(train_data_2)
[1] 108097     13
>
> train_data_3 <- read.csv("train_data_3.csv")
> dim(train_data_3)
[1] 108097     13
>
> test_data_1 <- read.csv("test_data_1.csv")
> dim(test_data_1)
[1] 36033    13
> test_data_2 <- read.csv("test_data_2.csv")
> dim(test_data_2)
[1] 36033    13
>
> test_data_3 <- read.csv("test_data_3.csv")
> dim(test_data_3)
[1] 36033    13
>
```

Selecting the most predictive linear and multivariate models from our summary list for deployment on the above datasets and reporting values:

LINEAR MODELS TAKEN UP FOR TRAINING AND PREDICTION USING MULTIPLE TRAINING AND TESTING DATASETS										
	ORIGINAL TRAINING DATASET					*TRAINING DATASET-1*				
Name of Model	***Coefficients of Ivs***	***P-val significant ?***	***Multiple R-squared***	***corr***	***mse***	***Coefficients of Ivs***	***P-val significant ?***	***Multiple R-squared***	***corr***	***mse***
bondRate + fedFundsRate + Borrower Credit Score + UPB+ Seller Name & Original Interest Rate	BR: 8.016e-01 FFR: 5.822e-01 CS: -2.014e-03 UPB: -5.679e-07 SN: 0.0068 to 0.68	Yes <2e-16 for BR, FFR, CS, UPB. For seller Name few levels have no significance	0.4903	0.7051072	0.3900587	BR: 7.981e-01 FFR: 5.858e-01 CS: -2.034e-03 UPB: -5.667e-07 SN: 0.006 to 0.674	Yes <2e-16 for BR, FFR, CS, UPB. For seller Name 5/24 levels have no significance	0.4908	0.7039908	0.390636
fedFundsRate & Original Interest Rate	1.043833	Yes <2e-16	0.3515	0.5832867	0.5117943	1.043034	Yes <2e-16	0.3519	0.5870011	0.5076281
bondRate & Original Interest Rate	1.357337	Yes <2e-16	0.3377	0.5974069	0.4988715	1.35495	Yes <2e-16	0.3363	0.5870011	0.5076281
	TRAINING DATASET-2					*TRAINING DATASET-3*				
Name of Model	***Coefficients of Ivs***	***P-val significant ?***	***Multiple R-squared***	***corr***	***mse***	***Coefficients of Ivs***	***P-val significant ?***	***Multiple R-squared***	***corr***	***mse***
bondRate + fedFundsRate + Borrower Credit Score + UPB+ Seller Name & Original Interest Rate	BR: 7.964e-01 FFR: 5.863e-01 CS: -2.013e-03 UPB: -5.716e-07 SN: 0.0055 to 0.657	Yes <2e-16 for BR, FFR, CS, UPB, SN has 4 levels without significance	0.4918	0.7018611	0.396014	BR: 8.068e-01 FFR: 5.814e-01 CS: -2.066e-03 UPB: -5.621e-07 SN: 0.0055 to 0.657	Yes <2e-16 for BR, FFR, CS, UPB, SN has 4 levels without significance	0.4923	0.7006703	0.3951509
fedFundsRate & Original Interest Rate	1.0439	Yes <2e-16	0.353	0.5936275	0.5054402	1.044682	Yes <2e-16	0.3525	0.594867	0.5015554
bondRate & Original Interest Rate	1.355302	Yes <2e-16	0.3374	0.5841424	0.5141844	1.361393	Yes <2e-16	0.3396	0.5783647	0.516599

Interpretation of above results : It can be observed from the results above that the similarities in results are striking. There are no significant differences in either coefficient values, their statistical significance, or Multiple-R-squared values, or Correlation and MSE values for predictions on test data, in the runs with different randomly selected training and test datasets. Therefore, the learnings from the original analysis holds good.

Reporting results from deploying regularization on the multiple training and test datasets:

REGULARIZATION: STATISTICS FROM DEPLOYING TRAINED MODEL ON TEST DATASETS					
Training Dataset	lambda.min	Coeff Vals	COR	MSE	Most Predictive features
train_data_1	0.001489	BR:7.449254e-01 FFR:6.257046e-01 UPB:--6.183709e-07 LTV: . CLTV:3.894270e-03 No.of Borr:1.640833e-02 DTI:1.732744e-03 CS:-1.566664e-03	0.673734	0.422942	1.bondRate 2.fedFundsRate
train_data_2	0.00149	BR: 7.477525e-01 FFR:6.301071e-01 UPB:-6.644302e-07 LTV: . CLTV:4.178392e-03 No.of Borr:2.818752e-02 DTI:2.256784e-03 CS:-1.594982e-03	0.670359	0.429768	1.bondRate 2.fedFundsRate
train_data_3	0.001489	BR:7.582028e-01 FFR:6.256183e-01 UPB:-6.475223e-07 LTV: . CLTV:4.214750e-03 No.of Borr:2.910613e-02 DTI:1.862368e-03 CS: -1.655554e-03	0.669035	0.428786	1.bondRate 2.fedFundsRate

Training Dataset	lambda.min	Coeff Vals	COR	MSE	Most Predictive features
Original Training dataset	0.001493	BR: 0.7463266 FFR:0.6279284 UPB:-6.245435e-07 LTV: . CLTV:3.953144e-03 No.of Borr:1.642157e-02 DTI:1.442769e-03 CS:-1.536119e-03	0.670395	0.426443	1. bondRate 2.fedFundsRate

Interpretation: As expected, the coefficient values from the original trained dataset, the lambda.min values, the correlation and MSE values from the predictions made on the test dataset, all hold good,

The results from the regularization experiments with multiple datasets are strikingly similar to the original values.

Question 2. Logistic Regression and NB

a. With the knowledge gathered from question 1(b), computing a logistic regression model with respect to different sets of independent features on our training dataset. Reporting various elements:

Logistic regression gives us the opportunity to explore another variable as an outcome variable. The outcome variable that answers our first two research questions was 'Original Interest Rate', which is the loan interest rate on the 'note', also called the 'Note Rate'. We explored the predictive features for this variable using Linear, Multivariate and Regularization methods. Since logistic regression models are linear models that output a nominal label based on a set of independent features, for this part of our analysis, we chose "Occupancy Status' variable that has three classes namely "P" , "I", and "S"(or levels) as the outcome variable. This part of the analysis answers our third research question, "Can we accurately predict the class of 'Occupancy Status' of a loan, using a set of IVs from our Loan dataset?" Since multinomial regression assumes that the levels of the outcome variable are unordered, this set of IVs and DV suits this part of the analysis perfectly. We used multinomial logistic regression models, with which we can classify the output into more than two classes.

Learnings from 1(a), Stage1 with old dataset:

1. Our Number of Units, numeric variable had only one value. A linear model could not be trained using it because of singularities.

2. Our Loan purpose categorical variable was converted to a factor to be deployed in training a model. It had singularities between levels and therefore a model could not be trained on this variable either.

3. All IVs have very small coefficients with low p-values and the models built have low R-squared values and do not exhibit 'goodness of fit' and therefore have very limited predictive power in predicting the outcome variable, interest rate.

Stage2: with new dataset

4. After we infused 3 new variables named bondRate, fedFundsRate and benchMarkMortgageRate, we found that fedFundsRate and benchMarkMortgageRate had multicollinearity (r = -0.995)

5. Linear models built with new variables performed better at predicting the outcome variable.

Learnings from 1(b) multivariate regression analysis :

From training linear models built with each IV and DV separately on the training dataset and using them to predict the outcome variable interest rate using the 'test' dataset, we have learned that multivariate regression models perform better in comparison. Thereafter in answering 1(b), we trained models with different combinations of IVs and DV, to predict the outcome variable. Two models stood out. One model trained with bondRate + fedFundsRate+ Borrower Credit Score + UPB+ Seller Name as IVs & Original Interest Rate as DV, had lower coefficient values for 4 our of 5 variables but an R-squared value of 0.4903 which was the maximum value amongst all models trained. However, the same model without Seller Name had slightly better magnitudes of coefficients for the IVs with a slightly lower R-squared of 0.4357. We therefore chose this model for logistic regression.

Learnings from 1©, Regularization:

Also, while trying to improve our model using regularization, we understood that the LTV variable did not have any impact in predicting the outcome variable.

With all the above learning, we finalized the following list of variables for Logistic Regression analysis:

Logistic Regression Model, IVs and DV			
IVs	Type and what it means	DV	Type and what it means
Original Interest Rate	Numeric, Continuous, The Note rate on the Loan at the time of sanctioning the loan to the borrower	Occupancy Status	Categorical, nominal, whether the underlying property of the loan is the Primary Home of the Borrower (Class "P"), or Secondary Home of the borrower, (Class "S") or if the Property is for investment purposes "I")
bondRate	Numeric, discrete, 10 year Treasury Bond Yield Rate		
fedFundsRate	Numeric, discrete, Federal Funds Rate, this is the rate at which lending institutions borrow funds overnight (Cost of funds)		
Borrower Credit Score	Numeric, discrete, Credit Score of the borrower		
Original UPB	Numeric, discrete, Loan quantum		

We benefited a lot from this article in performing this analysis. We created two datasets called trainLogistic and testLogistic with IVs and DV decided as per results learned from 1(b)

```
> trainLogistic <- train %>%
+   select(Original.Interest.Rate,bondRate,fedFundsRate,Borrower.Credit.Score.at.Origination,Original.UPB,Occupancy.Status)
> trainLogistic$Occupancy.Status <-as.factor(trainLogistic$Occupancy.Status)
> dim(trainLogistic)
[1] 108097      6
> testLogistic <- test %>%
+   select(Original.Interest.Rate,bondRate,fedFundsRate,Borrower.Credit.Score.at.Origination,Original.UPB,Occupancy.Status)
> testLogistic$Occupancy.Status <-as.factor(testLogistic$Occupancy.Status)
> dim(testLogistic)
[1] 36033     6
>
```

The distribution of numerical variables from the dataset

```
> result <- trainLogistic %>%
+   group_by(Occupancy.Status) %>%
+   summarise(
+     M_oir = mean(Original.Interest.Rate),
+     SD_oir = sd(Original.Interest.Rate),
+     M_br = mean(bondRate),
+     SD_br = sd(bondRate),
+     M_ffr = mean(fedFundsRate),
+     SD_ffr = sd(fedFundsRate),
+     M_cs = mean(Borrower.Credit.Score.at.Origination),
+     SD_cs = sd(Borrower.Credit.Score.at.Origination),
+     M_upb = mean(Original.UPB),
+     SD_upb = sd(Original.UPB)
+   )
> result
# A tibble: 3 x 11
  Occupancy.Status M_oir SD_oir  M_br SD_br M_ffr SD_ffr  M_cs SD_cs   M_upb  SD_upb
  <fct>            <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>   <dbl>   <dbl>
1 I                 6.65  0.805  3.59 0.374  2.80  0.526  761.  37.2 230548. 128904.
2 P                 6.01  0.866  3.58 0.378  2.74  0.497  753.  42.0 315443. 142888.
3 S                 6.56  0.880  3.60 0.376  2.79  0.520  767.  39.0 286570. 137335.
>
```

Our outcome variable Occupancy status has 3 levels. We are using the multinom() function from the nnet package in R to estimate a multinomial regression model. R chooses the first level as reference and therefore in the following output, "I" investor level is the reference level.

```
> logisticModel <- multinom(Occupancy.Status~ Original.Interest.Rate+bondRate+
+                             fedFundsRate+Borrower.Credit.Score.at.Origination+Original.UPB,
+                           data = trainLogistic)
# weights:  21 (12 variable)
initial  value 118756.692568
iter  10 value 37023.184531
iter  20 value 35303.635122
final  value 35303.634443
converged
> summary(logisticModel)
Call:
multinom(formula = Occupancy.Status ~ Original.Interest.Rate +
    bondRate + fedFundsRate + Borrower.Credit.Score.at.Origination +
    Original.UPB, data = trainLogistic)

Coefficients:
  (Intercept) Original.Interest.Rate  bondRate fedFundsRate Borrower.Credit.Score.at.Origination Original.UPB
P   12.049758             -1.7672483 1.6524074   0.86991574                         -0.010425841 4.552703e-06
S   -3.817622             -0.1937016 0.2033442   0.08254954                          0.002888868 3.395535e-06

Std. Errors:
   (Intercept) Original.Interest.Rate     bondRate fedFundsRate Borrower.Credit.Score.at.Origination Original.UPB
P 4.905241e-08           3.240558e-07 1.759811e-07 1.368828e-07                        3.721224e-05 1.058745e-07
S 8.632296e-08           5.756839e-07 3.109272e-07 2.428030e-07                        6.661371e-05 1.710643e-07

Residual Deviance: 70607.27
AIC: 70631.27
>
```

Interpretation of other results:

The output shows some iteration values and includes a final negative log-likelihood 35303.634443. This value multiplied by 2 is seen as the 'Residual Deviance' 70607.27. Lower values for Residual deviance and AIC indicate a better model. However, these values are not absolute and are given for comparison with the results of other models. In the table we capture the coefficients and the log odds which are explained in terms of reference level "I" for the outcome variable "Occupancy Status".

Levels of the DV "I" is taken as the reference level.	What is the intercept?	What are the coefficients for each of the features?	SE for the Coefficient	Are they statistically significant?	What are the log-odds of the outcome for a unit increase in each independent variable?
P	12.049758 This intercept represents the log-odds of being in category "P" as opposed to the reference level "I" are 12.049758 times higher.	OIR:-1.7672483 BR:1.6524074 FFR:0.86991574 CS: -0.010425841 UPB:4.552703e-06 The coefficients represent the log-odds of each category ("P" and "S") compared to the reference category ("I") not shown here.	OIR: 3.240558e-07 BR:1.759811e-07 FFR:1.368828e-07 CS: 3.721224e-05 UPB:1.058745e-07	Yes. P values calculated separately using R code show they are all less than alpha(0.05)	OIR: Holding other variables constant, the Coeff estimate of -1.7672483 for this variable means that for one unit increase in this variable, the log odds of being in level "P" Vs. "I", decrease by approximately the same number of units.(1.767) BR: Holding other variables constant, the Coeff of 1.6524074 for this variable means that for one unit increase in this variable, the log odds of being in level "P" Vs. "I", increase by approximately the same number of units.(1.6524074) FFR: Holding other variables constant, the Coeff of 0.86991574 for this variable means that for one unit increase in this variable, the log odds of being in level "P" Vs. "I", increase by approximately the ame number of units.(0.86991574) CS: Holding other variables constant, the Coeff estimate of -0.010425841 for this variable means that for one unit increase in this variable, the log odds of being in level "P" Vs. "I", decrease by approximately the same number of units.(-0.010425841) UPB: Holding other variables constant, the Coeff estimate of 4.552703e-06 for this variable means that for one unit increase in this variable, the log odds of being in level "P" Vs. "I", increase by approximately the same number of units.(4.552703e-06)

Levels of the DV "I" is taken as the reference level.	What is the intercept?	What are the coefficients for each of the features?	SE for the Coefficient	Are they statistically significant?	What are the log-odds of the outcome for a unit increase in each independent variable?
S	-3.817622 This intercept value represents the log-odds of being in category "S" as opposed to the reference level "I" are 3.817622 times lower.	OIR: -0.1937016 BR: 0.2033442 FFR:0.08254954 CS: 0.002888868 UPB: 3.395535e-06	OIR: 5.756839e-07 BR: 3.109272e-07 FFR: 2.428030e-07 CS: 6.661371e-05 UPB: 1.710643e-07	Yes. . P values calculated separately using R code show they are all less than alpha(0.05)	OIR: Holding other variables constant, the Coeff estimate of -0.1937016 for this variable means that for one unit increase in this variable, the log odds of being in level "S" Vs. "I", decrease by approximately the same number of units.(-0.1937016) BR: Holding other variables constant, the Coeff of 0.2033442 for this variable means that for one unit increase in this variable, the log odds of being in level "S" Vs. "I", increase by approximately the same number of units.(0.2033442) FFR: Holding other variables constant, the Coeff of 0.08254954 for this variable means that for one unit increase in this variable, the log odds of being in level "S" Vs. "I", increase by approximately the same number of units.(0.08254954) CS: Holding other variables constant, the Coeff estimate of 0.002888868 for this variable means that for one unit increase in this variable, the log odds of being in level "S" Vs. "I", increase by approximately the same number of units (0.002888868) UPB: Holding other variables constant, the Coeff estimate of 3.395535e-06 for this variable means that for one unit increase in this variable, the log odds of being in level "S" Vs. "I", increase by approximately the same number of units.(3.395535e-06)

Using Odd Ratios: We can also exponentiate coefficients and interpret everything as odd-ratios instead of log-odds which we did using R.

```
> #EXPONENTIATING COEFFS AND INTERPRET AS ODD RATOS
> exp(coef(logisticModel))
   (Intercept) Original.Interest.Rate bondRate fedFundsRate Borrower.Credit.Score.at.Origination Original.UPB
P 1.710580e+05             0.1708023 5.219530     2.386710                           0.9896283     1.000005
S 2.198002e-02             0.8239037 1.225494     1.086052                           1.0028930     1.000003
>
```

Interpretation:

1. *Intercept Values: According to this output, we can say that the odds of being in category "P" are approximately 171,058 times (intercept) higher than the reference category "I". For "S", the odds are 0.022 times higher than the reference category.*

2. *Coefficients :*

Name of IV	*Coeff value of each level of Outcome variable*	*Interpretation*
Original Interest Rate	P: 0.1708023 S: 0.8239037	P coeff: For one unit increase in the interest rate, the odds of being in category "P" are approximately 0.1708023 times the odds of being in reference category "I" S coeff: For one unit increase in the interest rate, the odds of being in category "S" are approximately 0.8239037times the odds of being in reference category "I"
bond Rate	P: 5.219530 S: 1.225494	P coeff: For one unit increase in the interest rate, the odds of being in category "P" are approximately 5.219530 times the odds of being in reference category "I" S coeff: For one unit increase in the interest rate, the odds of being in category "S" are approximately 1.225494times the odds of being in reference category "I"

Name of IV	*Coeff value of each level of Outcome variable*	*Interpretation*
fedFundsRate	P: 2.386710 S: 1.086052	P coeff: For one unit increase in the interest rate, the odds of being in category "P" are approximately 2.386710 times the odds of being in reference category "I" S coeff: For one unit increase in the interest rate, the odds of being in category "S" are approximately 1.086052 times the odds of being in reference category "I"
Borrower Credit Score at Origination	P: 0.9896283 S: 1.0028930	P coeff: For one unit increase in the interest rate, the odds of being in category "P" are approximately 0.9896283 times the odds of being in reference category "I" S coeff: For one unit increase in the interest rate, the odds of being in category "S" are approximately 1.0028930 times the odds of being in reference category "I"
Origination UPB	P: 1.000005 S: 1.000003	P coeff: For one unit increase in the interest rate, the odds of being in category "P" are approximately 1.000005 times the odds of being in reference category "I" S coeff: For one unit increase in the interest rate, the odds of being in category "S" are approximately 1.000003 times the odds of being in reference category "I"

Most predictive features:

According to logistic regression analysis on the training data, the coefficients of the variables Original Interest Rate, Bond Rate and Federal Funds Rate have highest magnitude and statistically significant coefficients for both "P" and "S" levels of the outcome variable. These are the most predictive features according to training data.

Predicting with the trained model on test data and explaining results:

We then proceeded to use the trained model to predict the testing dataset using the type = "class" argument and visualized the results using the confusion matrix. Each row in the confusion matrix displays the actual class and each column represents the predicted class.

```
> classPredictions <- predict(logisticModel, newdata = testLogistic, type = "class")
> library(caret)
> true_labels <- testLogistic$Occupancy.Status
> cMatrix <- confusionMatrix(classPredictions, true_labels)
> cMatrix
Confusion Matrix and Statistics

          Reference
Prediction     I     P     S
         I   189    65    37
         P  2404 32457   881
         S     0     0     0

Overall Statistics

               Accuracy : 0.906
                 95% CI : (0.9029, 0.909)
    No Information Rate : 0.9026
    P-Value [Acc > NIR] : 0.01379

                  Kappa : 0.0975

 Mcnemar's Test P-Value : < 2e-16

Statistics by Class:

                     Class: I Class: P Class: S
Sensitivity          0.072889  0.99800  0.00000
Specificity          0.996950  0.06437  1.00000
Pos Pred Value       0.649485  0.90809      NaN
Neg Pred Value       0.932740  0.77663  0.97452
Prevalence           0.071962  0.90256  0.02548
Detection Rate       0.005245  0.90076  0.00000
Detection Prevalence 0.008076  0.99192  0.00000
Balanced Accuracy    0.534919  0.53119  0.50000
>
```

Interpretation : Our test dataset has 36033 observations. If we take the “P” class of our outcome variable, our model correctly predicted 32457 instances of the P class correctly as P class, 2404 instances from “I” class were incorrectly classified as P and 881 instances from “S” class were incorrectly classified as “P”

The overall accuracy of the model with 95% Confidence Interval is given as 0.906 which is 90.6%. The associated p-value is 0.01379 which is statistically significant.

Sensitivity indicates the True Positive rate: The sensitivity value for class I is 0.072889 which is very low, also for “S” which is 0. The value of “P” is high at 0.998, This indicates that the ability of the model to classify instances of “P” class is high whereas other classes are low.

Specificity indicates the True Negative Rate: This statistic indicates the ability of the model to correctly identify whether or not an instance belongs to a specific class. The values for specificity are high for S and I classes but low for P.

Overall, the model performs better in predicting class “P”. It performs poorly in predicting other classes “I” and “S”.

Run-2:

As explained in the beginning, after completing our report and finding answers to our research questions, we attempted to understand if the Classifiers can be improved by using some more variables from our datasets. Therefore, we add this section for Logistic, NB and Decision Tree analysis.

We started with the larger dataset and split our data into training and test datasets and named them Log_Train_FM2022Q4.csv and Log_Test_FM2022Q4.csv. We then selected Property Type, Loan Purpose variables in addition to the original IVs to find out if our model improves in predictive power.

```
> trainLogistic <- subset(train,
+                         select= c(Original.Interest.Rate,bondRate,fedFundsRa
te,Borrower.Credit.Score.at.Origination,Original.UPB,Loan.Purpose,Property.Typ
e,Occupancy.Status))
> trainLogistic$Occupancy.Status <-as.factor(trainLogistic$Occupancy.Status)
> trainLogistic$Property.Type <- as.factor(trainLogistic$Property.Type)
> trainLogistic$Loan.Purpose <- as.factor(trainLogistic$Loan.Purpose)
> dim(trainLogistic)
[1] 108098      8
> testLogistic <- subset(test,
+                        select=c(Original.Interest.Rate,bondRate,fedFundsRat
e,Borrower.Credit.Score.at.Origination,Original.UPB,Loan.Purpose,Property.Typ
e,Occupancy.Status))
> testLogistic$Occupancy.Status <-as.factor(testLogistic$Occupancy.Status)
> testLogistic$Property.Type <-as.factor(testLogistic$Property.Type)
> testLogistic$Loan.Purpose <- as.factor(testLogistic$Loan.Purpose)
> dim(testLogistic)
[1] 36032     8
> #Using table command to generate a frequency table for the outcome variable.
> with(trainLogistic, table(Occupancy.Status))
Occupancy.Status
    I     P     S
 7924 97384  2790
>
```

This is our new model with additional variables, the coefficients of all the variables in the model show improvement from the model in Run-1.

```
> summary(logisticModel)
Call:
multinom(formula = Occupancy.Status ~ Original.Interest.Rate +
    bondRate + fedFundsRate + Borrower.Credit.Score.at.Origination +
    Original.UPB + Property.Type + Loan.Purpose, data = trainLogistic)

Coefficients:
  (Intercept) Original.Interest.Rate  bondRate fedFundsRate
P   11.928919             -1.8231029 1.6817987   0.90697472
S   -3.696673             -0.1831447 0.1831598   0.02088957
  Borrower.Credit.Score.at.Origination Original.UPB Property.TypeCP Property.TypeMH
P                         -0.010595366 4.815738e-06        12.34802        15.13331
S                          0.002620908 3.319974e-06        11.33165        14.71613
  Property.TypePU Property.TypeSF Loan.PurposeP Loan.PurposeR
P     -0.08776791       0.2727365     0.2386378    -0.4013404
S     -0.43635581      -0.4773772     0.7702860     0.1410660

Std. Errors:
   (Intercept) Original.Interest.Rate     bondRate fedFundsRate
P 5.007782e-08           3.290433e-07 1.797273e-07 1.394185e-07
S 8.742952e-08           5.847385e-07 3.148477e-07 2.459113e-07
  Borrower.Credit.Score.at.Origination Original.UPB Property.TypeCP Property.TypeMH
P                         3.798514e-05 1.078708e-07    4.903526e-10    5.101045e-09
S                         6.760064e-05 1.743815e-07    4.903599e-10    5.101049e-09
  Property.TypePU Property.TypeSF Loan.PurposeP Loan.PurposeR
P    2.941580e-09    4.842456e-08  2.959213e-08  3.047134e-09
S    4.552041e-09    5.530175e-08  7.194564e-08  2.948789e-09

Residual Deviance: 69035.47
AIC: 69083.47
```

We then checked if they were all statistically significant and found them to be.

```
> z <- summary(logisticModel)$coefficients/summary(logisticMode
l)$standard.errors
> z
  (Intercept) Original.Interest.Rate  bondRate fedFundsRate
P   238207640             -5540617.1 9357503.4   6505410.76
S   -42281748              -313207.8  581740.9     84947.58
  Borrower.Credit.Score.at.Origination Original.UPB
P                           -278.93452     44.64359
S                             38.77046     19.03856
  Property.TypeCP Property.TypeMH Property.TypePU
P     25181920490      2966707309       -29836999
S     23108842036      2884922621       -95859378
  Property.TypeSF Loan.PurposeP Loan.PurposeR
P         5632194       8064230    -131710766
S        -8632227      10706501      47838639
> p <- (1 - pnorm(abs(z), 0, 1)) * 2
> p
  (Intercept) Original.Interest.Rate bondRate fedFundsRate
P           0                      0        0            0
S           0                      0        0            0
  Borrower.Credit.Score.at.Origination Original.UPB
P                                    0            0
S                                    0            0
  Property.TypeCP Property.TypeMH Property.TypePU
P               0               0               0
S               0               0               0
  Property.TypeSF Loan.PurposeP Loan.PurposeR
P               0             0             0
S               0             0             0
>
```

We then predicted using the trained model and found that the predictions improved for I and P classes, but the model still could not predict “S’ class.

```
> classPredictions <- predict(logisticModel, newdata = testLogis
tic, type = "class")
> CrossTable(testLogistic$Occupancy.Status, classPredictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |       218 |      2473 |      2691 |
             |     0.006 |     0.069 |           |
-------------|-----------|-----------|-----------|
           P |       110 |     32308 |     32418 |
             |     0.003 |     0.897 |           |
-------------|-----------|-----------|-----------|
           S |        39 |       884 |       923 |
             |     0.001 |     0.025 |           |
-------------|-----------|-----------|-----------|
Column Total |       367 |     35665 |     36032 |
-------------|-----------|-----------|-----------|
```

We therefore attempted to train our model on the entire dataset to see if it can predict S class.

```
> logisticModel <- multinom(Occupancy.Status~ .,
+                           data = train)
# weights:  288 (190 variable)
initial  value 118757.791180
iter  10 value 37512.602430
iter  20 value 33286.554565
iter  30 value 33002.758141
iter  40 value 32371.364389
iter  50 value 32013.469706
iter  60 value 31772.745373
iter  70 value 31673.860227
iter  80 value 31569.691584
iter  90 value 31522.636605
iter 100 value 31485.551432
final  value 31485.551432
stopped after 100 iterations
>
> summary(logisticModel)
Call:
multinom(formula = Occupancy.Status ~ ., data = train)
```

Cross Table showing that the Model is predicting P as P for 89.4% of the time, I as I for 1.2% of the time and S as S, 8 times out of 923 instances (0.11%) available for this class.

```
> classPredictions <- predict(logisticModel, newdata = test, type = "class")
> CrossTable(test$Occupancy.Status, classPredictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))

   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

             | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       423 |      2258 |        10 |      2691 |
             |     0.012 |     0.063 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       261 |     32151 |         6 |     32418 |
             |     0.007 |     0.892 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        92 |       823 |         8 |       923 |
             |     0.003 |     0.023 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |       776 |     35232 |        24 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

Summary for Run-2: With additional variables, the Residual Deviance and AIC parameter and coefficients improved for the variables. Even with the entire dataset, the Logistic Regression model could only predict some values for “S " class of the outcome variable. This indicates that we should look for more defining features for S class from other sources or look for a superior classifier for this dataset.

b. *Reporting classification results using Naïve Bayes:*

For Naive Bayes function, we chose the same set of variables as those for Logistic regression analysis. The reason is, Naive Bayes is also a classifier like the logistic regression model and if we use the same set of variables we can compare and analyze the performance of these classifiers in answering our third research question, "Can we predict accurately, the class of the output variable Occupancy Status from our dataset, using a set of independent variables such as the following?'

Naïve Bayes Model, IVs and DV			
IVs	Type and what it means	DV	Type and what it means
Original Interest Rate	Numeric, Continuous, The Note rate on the Loan at the time of sanctioning the loan to the borrower	Occupancy Status	Categorical, nominal, whether the underlying property of the loan is the Primary Home of the Borrower (Class "P"), or Secondary Home of the borrower, (Class "S") or if the Property is for investment purposes "I")
bondRate	Numeric, discrete, 10 year Treasury Bond Yield Rate		
fedFundsRate	Numeric, discrete, Federal Funds Rate, this is the rate at which lending institutions borrow funds overnight (Cost of funds)		
Borrower Credit Score	Numeric, discrete, Credit Score of the borrower		
Original UPB	Numeric, discrete, Loan quantum		

To train a model using Naïve Bayes classifier, we started with our New_FM2022Q4.csv dataset. We used the sample function to split our dataset into test and training sets and filtered out unpredictive variables.

```
> #Alternative method to create Training and test sets using sample function
> data <-  read.csv("New_FM2022Q4.csv")
> dim(data)
[1] 144130     16
> trainProportion <- 0.75
> indexSet <- 1:nrow(data)
> trainIndices <- sample(indexSet, size = round(trainProportion*length(indexSet)))
> testIndices <- setdiff(indexSet, trainIndices)
> trainNB <- data[trainIndices, ]
> testNB <- data[testIndices, ]
> dimt(trainNB)
Error in dimt(trainNB) : could not find function "dimt"
> dim(trainNB)
[1] 108098     16
> dim(trainNB)
[1] 108098     16
> dim(testNB)
[1] 36032    16
```

```
> trainNB <- trainNB %>%
+   select(Original.Interest.Rate,bondRate,fedFundsRate,Borrower.Credit.Score.at.Origination,O
riginal.UPB,Occupancy.Status)
> trainNB$Occupancy.Status <-as.factor(trainNB$Occupancy.Status)
> dim(trainNB)
[1] 108098      6
> testNB <- testNB %>%
+   select(Original.Interest.Rate,bondRate,fedFundsRate,Borrower.Credit.Score.at.Origination,O
riginal.UPB,Occupancy.Status)
> testNB$Occupancy.Status <-as.factor(testNB$Occupancy.Status)
> dim(testNB)
[1] 36032     6
> levels(trainNB$Occupancy.Status)
[1] "I" "P" "S"
> levels(testNB$Occupancy.Status)
[1] "I" "P" "S"
```

Training our model using Naïve Bayes function with and without Laplace estimator: The naive Bayes function calculates the A-priori probabilities and conditional probabilities for different classes in the outcome variable. We can use the conditional probabilities calculated to make predictions on test data and calculate accuracy of the model. Since we need to analyze the improvement in predictions with and without Laplace estimator, we first created our model without specifying the Laplace estimator. We first reproduce our results and will capture the results in tabular form and interpret them at the end of this part of the analysis.

```
> modelNB <- naiveBayes(Occupancy.Status~ Original.Interest.Rate+bondRate+fedFundsRate+
+                         Borrower.Credit.Score.at.Origination+Original.UPB, data= trainNB)
>
>
> modelNB

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
         I          P          S
0.07330385 0.90088623 0.02580991

Conditional probabilities:
   Original.Interest.Rate
Y       [,1]      [,2]
  I 6.643973 0.8075281
  P 6.012801 0.8657276
  S 6.564030 0.8729830

   bondRate
Y       [,1]      [,2]
  I 3.590413 0.3741246
  P 3.579237 0.3774865
  S 3.590677 0.3756991

   fedFundsRate
Y       [,1]      [,2]
  I 2.794633 0.5248346
  P 2.747119 0.4982266
  S 2.785376 0.5182037

   Borrower.Credit.Score.at.Origination
Y       [,1]     [,2]
  I 761.0463 37.24145
  P 753.1585 41.92172
  S 768.1312 38.84972

   Original.UPB
Y       [,1]     [,2]
  I 231487.8 128668.2
  P 315035.2 142693.6
  S 284950.5 136593.6
```

Making predictions with the above model trained without Laplace estimator and visualizing results as a confusion matrix using the CrossTable() function in R.

```
> predictions <- predict(modelNB, newdata = testNB)
There were 21 warnings (use warnings() to see them)
> CrossTable(testNB$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))

   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

              | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |        13 |      2678 |      2691 |
             |     0.000 |     0.074 |           |
-------------|-----------|-----------|-----------|
           P |         1 |     32417 |     32418 |
             |     0.000 |     0.900 |           |
-------------|-----------|-----------|-----------|
           S |         1 |       922 |       923 |
             |     0.000 |     0.026 |           |
-------------|-----------|-----------|-----------|
Column Total |        15 |     36017 |     36032 |
-------------|-----------|-----------|-----------|
```

Interpretation: For the class I, our model without the Laplace smoothing, has correctly predicted only 13 instances as I and incorrectly predicted 2678 instances as ‘P”. It did not predict class “S”. For Class ‘P”, the model correctly predicted 32417 instances as P but incorrectly predicted two instances as I. For the class S, the model incorrectly predicted 922 instances as P and 1 instance as I. It did not predict any values for “S” class. If we compare this output with the logistic regression output, we find that both these classifiers are better at predicting the class “P” than the other two classes.

Repeating the process above with the Laplace estimator and reporting results:

Laplace smoothing involves adding a small constant, usually 1, to each count of events to ensure that no probability becomes zero. Here we have trained our model using laplace =1

```
> modelNBL <- naiveBayes(Occupancy.Status~ Original.Interest.Rate+bondRate+fedFundsRate+
+                          Borrower.Credit.Score.at.Origination+Original.UPB, data= trainNB,laplace=1)
>
> modelNBL

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
         I          P          S
0.07372939 0.90034968 0.02592092

Conditional probabilities:
   Original.Interest.Rate
Y       [,1]      [,2]
  I 6.640072 0.8161212
  P 6.012234 0.8650815
  S 6.562448 0.8705138

   bondRate
Y       [,1]      [,2]
  I 3.588065 0.3740363
  P 3.579056 0.3774618
  S 3.589454 0.3754153

   fedFundsRate
Y       [,1]      [,2]
  I 2.789975 0.5234772
  P 2.746844 0.4981529
  S 2.784497 0.5185589

   Borrower.Credit.Score.at.Origination
Y       [,1]     [,2]
  I 761.3668 36.92443
  P 753.2381 41.96504
  S 768.2049 38.91015

   Original.UPB
Y       [,1]     [,2]
  I 231650.1 128997.7
  P 315388.6 142840.3
  S 286266.6 138914.1
```

The predictions and values from confusion matrix are :

```
> modelNBL <- naiveBayes(Occupancy.Status~ Original.Interest.Rate+bondRate+fedFundsRate+
+                          Borrower.Credit.Score.at.Origination+Original.UPB, data= trainNB,laplace=1)
> predictionsL <- predict(modelNBL, newdata = testNB)
> CrossTable(testNB$Occupancy.Status, predictionsL,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |         9 |      2636 |      2645 |
             |     0.000 |     0.073 |           |
-------------|-----------|-----------|-----------|
           P |         2 |     32474 |     32476 |
             |     0.000 |     0.901 |           |
-------------|-----------|-----------|-----------|
           S |         2 |       909 |       911 |
             |     0.000 |     0.025 |           |
-------------|-----------|-----------|-----------|
Column Total |        13 |     36019 |     36032 |
-------------|-----------|-----------|-----------|
```

Interpretation : With Laplace smoothing, the model is performing better at predicting P as P, less efficient at identifying I as I. *We can say that the model primarily predicted class 'P" for most of the instances of test data and did not predict any instances belonging to the S class correctly.*

We captured the model results with and without Laplace estimator in the following table.

Laplace Smoothing	Class of Output variable	A-priori probabilities	Conditional Probabilities									
			Original Interest Rate		bondRate		fedFundsRate		Credit Score		UPB	
			Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Yes	I	0.07372939	6.640072	0.8161212	3.588065	0.3740363	2.789975	0.5234772	761.3668	36.92443	231650.1	128997.7
No	I	0.7330385	6.643973	0.8075281	3.590413	0.3741246	2.794633	0.5248346	761.0463	37.24145	231487.8	128668.2
Yes	P	0.90034968	6.012234	0.8650815	3.579056	0.3774618	2.746844	0.4981529	753.2381	41.96504	315388.6	142840.3
No	P	0.90088623	6.012801	0.8657276	3.579237	0.3774865	2.747119	0.4982266	753.1585	41.92172	315035.2	142693.6
Yes	S	0.02592092	6.562448	0.8705138	3.589454	0.3754153	2.784497	0.5185589	768.2049	38.91015	286266.6	138914.1
No	S	0.02580991	6.56403	0.872983	3.590677	0.3756991	2.785376	0.5182037	768.1312	38.84972	284950.5	136593.6

Interpretation: Laplace smoothing ensures that no outcome is left without a probability value. This is effective where values in the dataset are sparse, which is not the case in our dataset. Also, Laplace smoothing produces robustness in the probability estimates in that it helps prevent events that were observed only a few times from having disproportionately high or low probabilities. The A-priori probabilities were lower for I class and S class without smoothing. They were higher for P class without smoothing. With smoothing, in almost all the cases, the conditional probabilities for all Classes are slightly lower. But all the variations are miniscule.

Reason for under performance of the model in predicting I and S classes correctly : Naïve Bayes assumes independence between predictor variables. But three of our predictor variables have a high degree of correlation as explained by Pearson's coefficient here. We can see that the underlying assumption of independence between predictor variables is violated in the dataset we used.

```
> cor(Original.Interest.Rate,fedFundsRate)
[1] 0.5941333
> cor(Original.Interest.Rate,bondRate)
[1] 0.5822616
> cor(bondRate,fedFundsRate)
[1] 0.6643931
```

Run2:

In Run-1, we realized that our Classifiers did not have enough features to detect “S” class correctly. We therefore attempted to improve our Classifiers by adding a Property Type and Loan Purpose variables from the dataset.

```
> trainNB <- read.csv("NB_Train_FM2022Q4.csv")
> testNB <- read.csv("NB_Test_FM2022Q4.csv")
> trainNB <- subset(trainNB,
+                   select=c(Original.Interest.Rate,bondRate,fed
FundsRate,Borrower.Credit.Score.at.Origination,Original.UPB,Prop
erty.Type, Loan.Purpose,Occupancy.Status))
> trainNB$Occupancy.Status <-as.factor(trainNB$Occupancy.Status)
> trainNB$Property.Type <- trainNB$Property.Type
> trainNB$Loan.Purpose <- as.factor(trainNB$Loan.Purpose)
> dim(trainNB)
[1] 108098      8
> testNB <- subset(testNB,
+                  select=c(Original.Interest.Rate,bondRate,fedF
undsRate,Borrower.Credit.Score.at.Origination,Original.UPB,Prope
rty.Type, Loan.Purpose,Occupancy.Status))
> testNB$Occupancy.Status <-as.factor(testNB$Occupancy.Status)
> testNB$Loan.Purpose <- as.factor(testNB$Loan.Purpose)
> testNB$Property.Type <- as.factor(testNB$Property.Type)
> dim(testNB)
[1] 36032     8
> levels(trainNB$Occupancy.Status)
[1] "I" "P" "S"
> levels(testNB$Occupancy.Status)
[1] "I" "P" "S"
>
```

We proceeded to train the models on the new set of IVs with and without Laplace estimator and found that the coefficients for all the IVs showed improvement in this model. But the model does not yet predict the “S” class with the addition of these two new variables.

Crosstable for the model (modelNB) without laplace estimator

```
>
> predictions <- predict(modelNB, newdata = testNB)
>
> CrossTable(testNB$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |        32 |      2659 |      2691 |
             |     0.001 |     0.074 |           |
-------------|-----------|-----------|-----------|
           P |         9 |     32409 |     32418 |
             |     0.000 |     0.899 |           |
-------------|-----------|-----------|-----------|
           S |         3 |       920 |       923 |
             |     0.000 |     0.026 |           |
-------------|-----------|-----------|-----------|
Column Total |        44 |     35988 |     36032 |
-------------|-----------|-----------|-----------|
```

Crosstable for the model (modelNBL) with laplace estimator:

```
> predictionsL <- predict(modelNBL, newdata = testNB)
>
> CrossTable(testNB$Occupancy.Status, predictionsL,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |        33 |      2658 |      2691 |
             |     0.001 |     0.074 |           |
-------------|-----------|-----------|-----------|
           P |        10 |     32408 |     32418 |
             |     0.000 |     0.899 |           |
-------------|-----------|-----------|-----------|
           S |         3 |       920 |       923 |
             |     0.000 |     0.026 |           |
-------------|-----------|-----------|-----------|
Column Total |        46 |     35986 |     36032 |
-------------|-----------|-----------|-----------|
```

We then tried our model with the entire dataset with and without Laplace estimator and found that the model starts predicting “S” class.:

CrossTable for model trained without Laplace smoothing shows the instances of correct predictions for I as I as 486, for “P” as “P” as 31782 and “S” as “S” as 14. These are called True Positives. The model predicts True positives 89.4% of the time.:

```
> modelNB <- naiveBayes(Occupancy.Status~ ., data= train)
>
> predictions <- predict(modelNB, newdata = test)
> CrossTable(test$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       486 |      2197 |         8 |      2691 |
             |     0.013 |     0.061 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       606 |     31782 |        30 |     32418 |
             |     0.017 |     0.882 |     0.001 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        82 |       827 |        14 |       923 |
             |     0.002 |     0.023 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |      1174 |     34806 |        52 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

Cross Table for model trained with Laplace smoothing indicates that the class predictions for “S” did not improve with smoothing but for I it improved marginally from 486 to 488 and decreased for “P” from 31782 to 31770. The model predicts True Positives 89.56% of the time.

```
> modelNBL <- naiveBayes(Occupancy.Status~ ., data= train, laplace = 1)
>
> predictions <- predict(modelNBL, newdata = test)
> CrossTable(test$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       488 |      2195 |         8 |      2691 |
             |     0.014 |     0.061 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       616 |     31770 |        32 |     32418 |
             |     0.017 |     0.882 |     0.001 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        82 |       827 |        14 |       923 |
             |     0.002 |     0.023 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |      1186 |     34792 |        54 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

Summary for Run-2 of Naive Bayes: When trained on the entire dataset, the model is able to predict all the classes in the outcome variable. However, the prediction quality is still poor.

Question 3. Decision Trees and Random Forests:

a . Splitting our dataset into training and testing sets assuming that samples are not randomly ordered and reporting distribution.

We split the data using random numbers into training and test datasets. The dimensions of our dataset are before split : 144130 obs of 16 variables. Training dataset: 108098 obs of 16 variables. Test Dataset : 36032 obs of 16 variables

```
> ## SPLITTING THE DATASET INTO TRAININ AND TESTING SETS USING RANDOM NUMBERS
> set.seed(12345)
> data <- read.csv("New_FM2022Q4.csv")
> #CREATE RANDOM NOS USING nrow() to create nrows of random nos
> randomNos <- runif(nrow(data))
> #USE randomNos to SHUFFLE DATA,
> datar <- data[order(randomNos), ]
> dim(datar)
[1] 144130     16
> train <- datar[1:108098, ] #75% of the dataset
> write.csv(train, file = "DT_Train_FM2022Q4.csv", row.names = FALSE)
> data <- read.csv("DT_Train_FM2022Q4.csv")
> dim(data)
[1] 108098     16
> test <- datar[108099:144130, ] #25% of the dataset
> write.csv(test, file= "DT_Test_FM2022Q4.csv", row.names = FALSE)
> data <- read.csv("DT_Test_FM2022Q4.csv")
> dim(data)
[1] 36032    16
> trainDT <- read.csv("DT_Train_FM2022Q4.csv")
> dim(trainDT)
[1] 108098     16
> testDT <- read.csv("DT_Test_FM2022Q4.csv")
> dim(testDT)
[1] 36032    16
```

We used the R summary() function to verify if the distributions are similar.

Distribution of Dataset before split

```
> data <- read.csv("New_FM2022Q4.csv")
> #### showing that the distributions are the same before and after split
> summary(data)
 Seller.Name        Original.Interest.Rate  Original.UPB       bondRate     fedFundsRate
 Length:144130      Min.   :3.550          Min.   : 20000   Min.   :3.26   Min.   :2.330
 Class :character   1st Qu.:5.375          1st Qu.:200000   1st Qu.:3.26   1st Qu.:2.330
 Mode  :character   Median :6.000          Median :289000   Median :3.26   Median :2.330
                    Mean   :6.075          Mean   :308346   Mean   :3.58   Mean   :2.751
                    3rd Qu.:6.875          3rd Qu.:402000   3rd Qu.:4.07   3rd Qu.:3.080
                    Max.   :8.125          Max.   :716000   Max.   :4.07   Max.   :3.830
 benchMarkMortgageRate Original.Loan.to.Value.Ratio..LTV.
 Min.   :6.420         Min.   :29.00
 1st Qu.:6.580         1st Qu.:70.00
 Median :6.700         Median :80.00
 Mean   :6.629         Mean   :77.97
 3rd Qu.:6.700         3rd Qu.:93.00
 Max.   :6.700         Max.   :97.00
 Original.Combined.Loan.to.Value.Ratio..CLTV. Number.of.Borrowers Debt.To.Income..DTI.
 Min.   : 34.00                                Min.   :1.000       Min.   :13.00
 1st Qu.: 70.00                                1st Qu.:1.000       1st Qu.:32.00
 Median : 80.00                                Median :1.000       Median :39.00
 Mean   : 78.21                                Mean   :1.464       Mean   :37.82
 3rd Qu.: 94.00                                3rd Qu.:2.000       3rd Qu.:45.00
 Max.   :105.00                                Max.   :3.000       Max.   :62.00
 Borrower.Credit.Score.at.Origination Loan.Purpose       Property.Type
 Min.   :631.0                        Length:144130      Length:144130
 1st Qu.:726.0                        Class :character   Class :character
 Median :761.0                        Mode  :character   Mode  :character
 Mean   :754.2
 3rd Qu.:788.0
 Max.   :839.0
 Number.of.Units Occupancy.Status   Property.State
 Min.   :1       Length:144130      Length:144130
 1st Qu.:1       Class :character   Class :character
 Median :1       Mode  :character   Mode  :character
 Mean   :1
 3rd Qu.:1
 Max.   :1
```

Distribution of Training Dataset (after split)

```
- -
> summary(trainDT)
 Seller.Name        Original.Interest.Rate  Original.UPB       bondRate     fedFundsRate   benchMarkMortgageRate
 Length:108098      Min.   :3.550          Min.   : 20000   Min.   :3.26   Min.   :2.330   Min.   :6.420
 Class :character   1st Qu.:5.375          1st Qu.:200000   1st Qu.:3.26   1st Qu.:2.330   1st Qu.:6.580
 Mode  :character   Median :6.000          Median :289000   Median :3.26   Median :2.330   Median :6.700
                    Mean   :6.074          Mean   :308481   Mean   :3.58   Mean   :2.752   Mean   :6.629
                    3rd Qu.:6.875          3rd Qu.:403000   3rd Qu.:4.07   3rd Qu.:3.080   3rd Qu.:6.700
                    Max.   :8.125          Max.   :716000   Max.   :4.07   Max.   :3.830   Max.   :6.700
 Original.Loan.to.Value.Ratio..LTV. Original.Combined.Loan.to.Value.Ratio..CLTV. Number.of.Borrowers Debt.To.Income..[
 Min.   :29.00                      Min.   : 34.00                               Min.   :1.000       Min.   :13.00
 1st Qu.:70.00                      1st Qu.: 70.00                               1st Qu.:1.000       1st Qu.:32.00
 Median :80.00                      Median : 80.00                               Median :1.000       Median :39.00
 Mean   :77.98                      Mean   : 78.22                               Mean   :1.465       Mean   :37.83
 3rd Qu.:93.00                      3rd Qu.: 94.00                               3rd Qu.:2.000       3rd Qu.:45.00
 Max.   :97.00                      Max.   :105.00                               Max.   :3.000       Max.   :62.00
 Borrower.Credit.Score.at.Origination Loan.Purpose       Property.Type      Number.of.Units Occupancy.Status
 Min.   :631.0                        Length:108098      Length:108098      Min.   :1       Length:108098
 1st Qu.:726.0                        Class :character   Class :character   1st Qu.:1       Class :character
 Median :761.0                        Mode  :character   Mode  :character   Median :1       Mode  :character
 Mean   :754.2                                                              Mean   :1
 3rd Qu.:788.0                                                              3rd Qu.:1
 Max.   :839.0                                                              Max.   :1
 Property.State
 Length:108098
 Class :character
 Mode  :character
```

Distribution of Test Dataset after (split)

```
> summary(testDT)
 Seller.Name        Original.Interest.Rate  Original.UPB       bondRate
 Length:36032       Min.   :3.625          Min.   : 20000   Min.   :3.260
 Class :character   1st Qu.:5.375          1st Qu.:200000   1st Qu.:3.260
 Mode  :character   Median :6.000          Median :289000   Median :3.260
                    Mean   :6.076          Mean   :307940   Mean   :3.582
                    3rd Qu.:6.875          3rd Qu.:401000   3rd Qu.:4.070
                    Max.   :8.125          Max.   :713000   Max.   :4.070
  fedFundsRate   benchMarkMortgageRate Original.Loan.to.Value.Ratio..LTV.
 Min.   :2.330   Min.   :6.420         Min.   :30.00
 1st Qu.:2.330   1st Qu.:6.580         1st Qu.:70.00
 Median :2.330   Median :6.700         Median :80.00
 Mean   :2.751   Mean   :6.629         Mean   :77.94
 3rd Qu.:3.080   3rd Qu.:6.700         3rd Qu.:93.00
 Max.   :3.830   Max.   :6.700         Max.   :97.00
 Original.Combined.Loan.to.Value.Ratio..CLTV. Number.of.Borrowers Debt.To.Income..DTI.
 Min.   : 34.00                               Min.   :1.000       Min.   :13.00
 1st Qu.: 70.00                               1st Qu.:1.000       1st Qu.:32.00
 Median : 80.00                               Median :1.000       Median :39.00
 Mean   : 78.19                               Mean   :1.461       Mean   :37.78
 3rd Qu.: 94.00                               3rd Qu.:2.000       3rd Qu.:45.00
 Max.   :105.00                               Max.   :3.000       Max.   :51.00
 Borrower.Credit.Score.at.Origination Loan.Purpose       Property.Type
 Min.   :631.0                        Length:36032       Length:36032
 1st Qu.:726.0                        Class :character   Class :character
 Median :761.0                        Mode  :character   Mode  :character
 Mean   :754.3
 3rd Qu.:788.0
 Max.   :834.0
 Number.of.Units Occupancy.Status   Property.State
 Min.   :1       Length:36032       Length:36032
 1st Qu.:1       Class :character   Class :character
 Median :1       Mode  :character   Mode  :character
 Mean   :1
 3rd Qu.:1
 Max.   :1
```

Interpretation : The distribution of the datasets is similar.

We then checked whether the proportion of Outcome variable is similar in both Training and Test datasets using the prop.table() function in R and found that the proportions are similar.

```
> prop.table(table(trainDT$Occupancy.Status))

         I          P          S
0.07372014 0.90045144 0.02582841
> prop.table(table(testDT$Occupancy.Status))

         I          P          S
0.07343472 0.90100466 0.02556061
```

b . Training a decision tree and interpreting the main resulting if-then rules together with their corresponding plots:

Since the previous classifiers did not predict I and S classes of our output variable correctly, which would have answered our 3rd research question conclusively, we decided to stick with the same analysis for Decision Tree analysis also, to observe if this model performs better. We therefore decided to use the same IVs and DV that we used in Logistic regression and Naïve Bayes function for the Decision Tree model also so as to be able to compare the performance of these classifiers on the same set of variables.

Decision Tree Model, IVs and DV			
IVs	Type and what it means	DV	Type and what it means
Original Interest Rate	Numeric, Continuous, The Note rate on the Loan at the time of sanctioning the loan to the borrower	Occupancy Status	Categorical, nominal, whether the underlying property of the loan is the Primary Home of the Borrower (Class "P"), or Secondary Home of the borrower, (Class "S") or if the Property is for investment purposes "I")
bondRate	Numeric, discrete, 10 year Treasury Bond Yield Rate		
fedFundsRate	Numeric, discrete, Federal Funds Rate, this is the rate at which lending institutions borrow funds overnight (Cost of funds)		
Borrower Credit Score	Numeric, discrete, Credit Score of the borrower		
Original UPB	Numeric, discrete, Loan quantum		

We trained the Decision Tree over the training dataset using the C5.0() function in R.

```
> modelDT <- C5.0(Occupancy.Status~ Original.Interest.Rate+bondRate+fedFundsRate+
+                   Borrower.Credit.Score.at.Origination+Original.UPB, data= trainDT)
> summary(modelDT)

Call:
C5.0.formula(formula = Occupancy.Status ~ Original.Interest.Rate + bondRate + fedFundsRate
 + Borrower.Credit.Score.at.Origination + Original.UPB, data = trainDT)


C5.0 [Release 2.07 GPL Edition]        Mon Oct 16 21:41:37 2023
-------------------------------

Class specified by attribute `outcome'

Read 108098 cases (6 attributes) from undefined.data
```

These are our main nodes and branches of the Decision Tree model which we explained using If -then rules.

```
Decision tree:

Original.Interest.Rate <= 5.973: P (49844/2180)
Original.Interest.Rate > 5.973:
:...Original.Interest.Rate > 7.577:
    :...Borrower.Credit.Score.at.Origination > 737:
    :   :...Original.Interest.Rate > 7.99: I (294/98)
    :   :   Original.Interest.Rate <= 7.99:
    :   :   :...Original.UPB <= 210000: I (751/338)
    :   :       Original.UPB > 210000: P (1178/515)
    :   Borrower.Credit.Score.at.Origination <= 737:
    :   :...Original.Interest.Rate <= 7.99: P (1702/343)
    :       Original.Interest.Rate > 7.99:
    :       :...Borrower.Credit.Score.at.Origination <= 678: P (67/9)
    :           Borrower.Credit.Score.at.Origination > 678:
    :           :...Borrower.Credit.Score.at.Origination > 716: I (72/41)
    :               Borrower.Credit.Score.at.Origination <= 716:
    :               :...Original.UPB <= 117000: I (16/3)
    :                   Original.UPB > 117000:
    :                   :...Original.Interest.Rate <= 7.999: I (9/3)
    :                       Original.Interest.Rate > 7.999: P (72/30)
    Original.Interest.Rate <= 7.577:
    :...bondRate > 3.26: P (38772/3323)
        bondRate <= 3.26:
        :...Original.Interest.Rate <= 6.624: P (11569/1958)
            Original.Interest.Rate > 6.624:
            :...Borrower.Credit.Score.at.Origination <= 728:
                :...Borrower.Credit.Score.at.Origination <= 694: P (815/127)
                :   Borrower.Credit.Score.at.Origination > 694:
                :   :...Original.UPB > 196000: P (528/114)
                :       Original.UPB <= 196000:
                :       :...Original.Interest.Rate > 6.995:
                :           :...Original.Interest.Rate <= 7.186: I (69/26)
                :           :   Original.Interest.Rate > 7.186: P (9/1)
                :           Original.Interest.Rate <= 6.995:
                :           :...Original.UPB <= 127000:
                :               :...Original.Interest.Rate <= 6.878: I (82/41)
                :               :   Original.Interest.Rate > 6.878: P (13/4)
                :               Original.UPB > 127000:
                :               :...Original.UPB <= 188000: P (109/32)
                :                   Original.UPB > 188000: I (8/2)
                Borrower.Credit.Score.at.Origination > 728:
                :...Original.UPB <= 180000:
                    :...Original.Interest.Rate > 7.186: P (18/4)
                    :   Original.Interest.Rate <= 7.186:
                    :   :...Original.Interest.Rate <= 6.644: I (303/117)
                    :       Original.Interest.Rate > 6.644:
                    :       :...Original.Interest.Rate <= 6.829: P (49/21)
                    :           Original.Interest.Rate > 6.829: I (312/101)
                    Original.UPB > 180000:
                    :...Borrower.Credit.Score.at.Origination > 775:
                        :...Original.UPB <= 402000: I (522/299)
                        :   Original.UPB > 402000: P (151/66)
                        Borrower.Credit.Score.at.Origination <= 775:
                        :...Original.Interest.Rate > 6.919:
                            :...Original.Interest.Rate > 7.186: P (11/2)
                            :   Original.Interest.Rate <= 7.186: [S1]
                            Original.Interest.Rate <= 6.919: [S2]
```

Part of the output showing how this model evaluated the training data

```
Evaluation on training data (108098 cases):

           Decision Tree
         ----------------
         Size      Errors

           36 10078( 9.3%)   <<


          (a)   (b)   (c)    <-classified as
         ----  ----  ----
         1429  6540          (a): class I
          746 96591          (b): class P
          364  2428          (c): class S


        Attribute usage:

        100.00% Original.Interest.Rate
         50.04% bondRate
          7.32% Borrower.Credit.Score.at.Origination
          4.59% Original.UPB
```

Interpretation: Decision trees have branches that are one side of a split and leaves that represent terminal nodes that return values. The IVs are tested in the conditions and the class of the outcome variable is returned in the terminal node. The entire decision tree can be explained as a series of IF-THEN rules where "if" splits inputs into categories and "then" assigns value. At the root of the decision tree

IF OIR value is <= 5.973 returns TRUE, THEN CLASS:P

IF OIR value is <=5.973 returns FALSE, i.e, IF OIR value is > 5.973 the tree splits into a branch. Here

IF OIR > 7.577 returns TRUE the tree splits into branch and checks

IF Borrower Credit Score is > 737 is TRUE tree splits checks

IF OIR > 7.99 return TRUE THEN class is "I"

IF OIR >7.99 returns FALSE i.e, OIR <= 7.99 tree splits Checks

IF Original UPB <= 210000 returns TRUE THEN CLASS I

IF Original UPB <= 210000 returns FALSE THEN CLASS P

IF Borrower Credit Score is > 737 is FALSE i.e., Bor.Cr.Sc <=737tree splits checks

IF OIR <= 7.99 returns TRUE THEN CLASS P

IF OIR <= 7.99 returns FALSE i.e. OIR >7.99, tree splits checks… so on.

IF OIR > 7.577 returns FALSE, i.e., OIR < =7.577 tree splits into a branch and checks

IF bondRate > 3.26 returns TRUE THEN class P

IF bondRate > 3.26 returns FALSE tree Splits and so on.

Summary: The size of the decision tree is 36. There are 10078 errors in classification (misclassifications) in this model which accounts for 9.3% of the total number of observations (108098).

The classes of our Output variables into which classifications were made are I, P and S and the model made correct and erroneous decisions in classification as under:

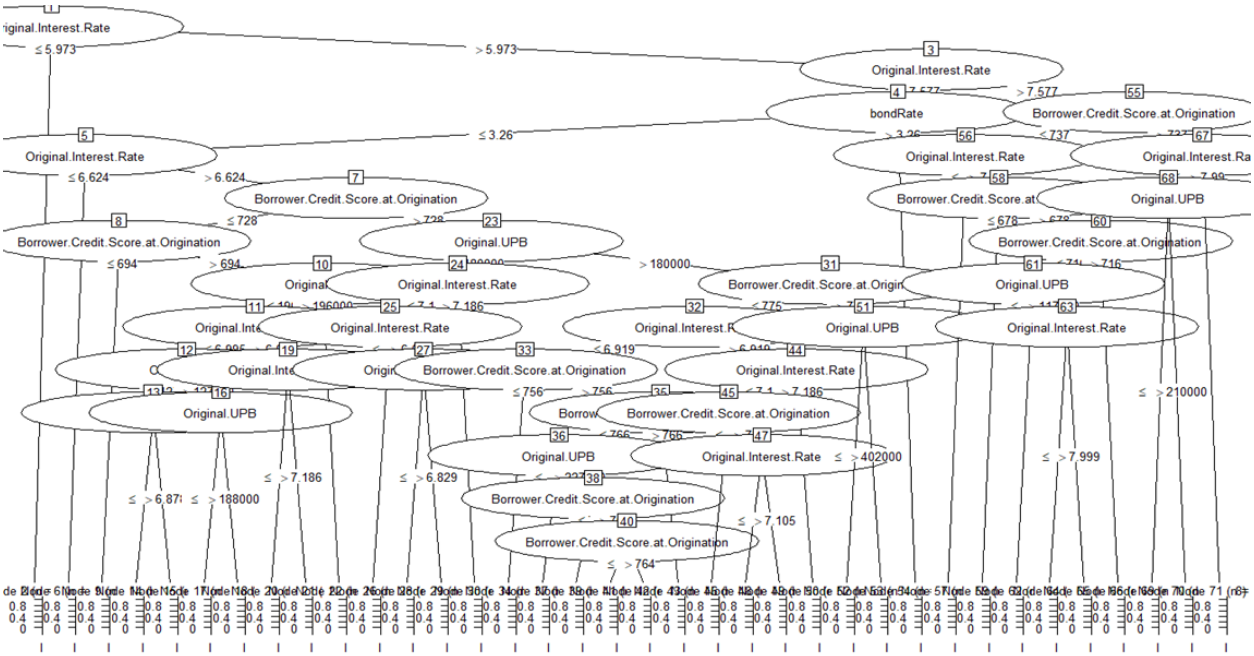
For Class I, it made the correct classification in 1429 cases. Misclassified as P in 746 cases and misclassified as S in 364 cases.

For Class P, the model erroneously classified obs. as belonging to I in 6540 cases and correctly classified 96591 observations as class P. It misclassified 2428 observations as class S.

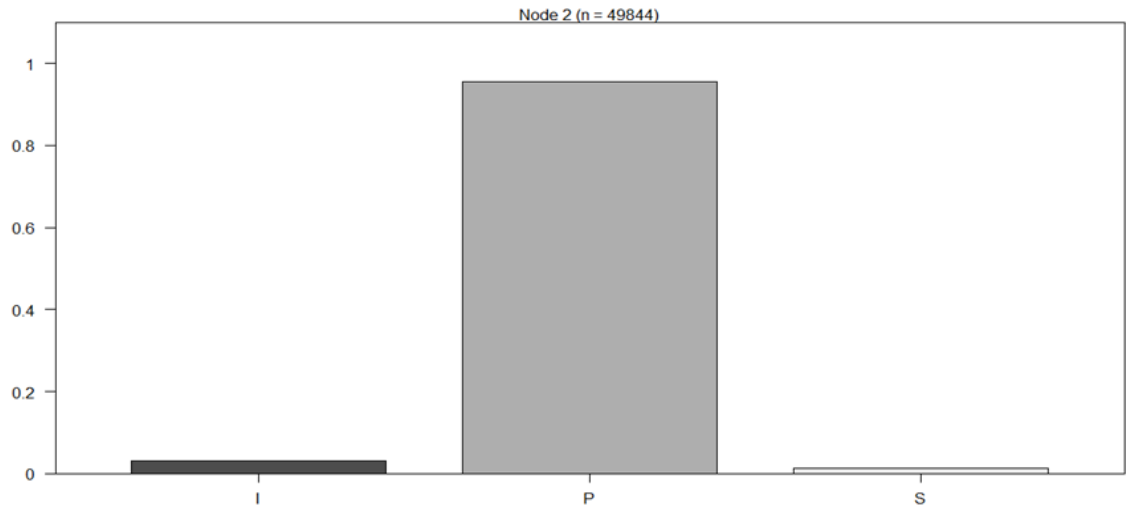
Overall the model is 90.7% accurate in classifying the obs.

Attribute usage: In making decisions, the model used OIR 100% of the time, bondRate for 50.04% of the time, Borrower Credit Score 7.32% of the time and Original UPB 4.59% of the time. We will further analyze the model's performance after making predictions with it.

We also plotted the decision tree, but because we have more IVs and consequently more conditions to be checked, we got a busy picture but the nodes and conditions that are checked at these nodes are clear.



We also visualized parts of the decision tree to visualize the class distributions. Here we find the class distribution for the subtree defined by node2. As per this plot, the highest bar is the "P" class which is the predicted class for the majority of instances in this subtree.



Testing the trained tree with our testing dataset and comparing the confusion matrices obtained during training and testing: computing percentages of correctly and incorrectly classified samples and comparing results in training and testing:

We predicted on a test dataset using the trained decision tree model which we named modelDT and then created the confusion matrix and statistics to analyze the results.

```
> predictions <- predict(modelDT, newdata = testDT)
> library(caret)
> confusionMatrix(predictions, testDT$Occupancy.Status)
Confusion Matrix and Statistics

          Reference
Prediction     I     P     S
         I   455   251   102
         P  2191 32214   819
         S     0     0     0

Overall Statistics

               Accuracy : 0.9067
                 95% CI : (0.9036, 0.9097)
    No Information Rate : 0.901
    P-Value [Acc > NIR] : 0.0001482

                  Kappa : 0.206

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: I Class: P Class: S
Sensitivity           0.17196   0.9923  0.00000
Specificity           0.98943   0.1562  1.00000
Pos Pred Value        0.56312   0.9145      NaN
Neg Pred Value        0.93780   0.6894  0.97444
Prevalence            0.07343   0.9010  0.02556
Detection Rate        0.01263   0.8940  0.00000
Detection Prevalence  0.02242   0.9776  0.00000
Balanced Accuracy     0.58069   0.5742  0.50000
```

Analysis of the above result:

The accuracy of the model is 90.67% which is almost the same as the trained model. (error rate was given as 9.3% meaning, 90.7% accuracy of the trained model). The 95% confidence interval is from 90.36% to 90.97% which indicates the confidence interval within which the accuracy of this model can be found. NIR is an important factor to understand in this model. If the model predicted "P" for every observation (without explicit decision making) it would still be 90.1% accurate. The kappa value of 0.206 indicates the agreement between predicted and actual values. It is low to fair for this model. The p-value <2.2e-16 indicates that the model predictions are not by chance.

Sensitivity specifies that the model is able to predict P class better than other classes. It is very bad at identifying S class. The specificity is 0.98943 for I, which means that the model is correct in identifying I class more. It is pertinent to recollect that the Naïve Bayes model predicted 'P" class correctly and performed poorly on S. At this point, we were concerned whether there were adequate values for class "S' in our test data for the model to train on and found that we did have 921 observations for class S in the test data. The total number of observations in the test data is 36032 of which there are 2646 obs belonging to I class, 32465 belonging to P class and 921 belonging to S class.

In the training phase of the model, for Class I, it made correct classification in 1429 cases. Misclassified as P in 746 cases and misclassified as S in 364 cases. For Class P, the model erroneously classified obs as belonging to I in 6540 cases and correctly classified 96591 observations as class P. It misclassified 2428 observations as class S. We repeated the confusion matrix with the CrossTable() function to visualize prediction results and create an accuracy % table.

```
> modelDT <- C5.0(Occupancy.Status~ Original.Interest.Rate+bondRate+fedFundsRate+
+                   Borrower.Credit.Score.at.Origination+Original.UPB, data= trainDT)
> predictions <- predict(modelDT, newdata = testDT)
> CrossTable(testDT$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))
```

```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

              | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |       455 |      2191 |      2646 |
             |     0.013 |     0.061 |           |
-------------|-----------|-----------|-----------|
           P |       251 |     32214 |     32465 |
             |     0.007 |     0.894 |           |
-------------|-----------|-----------|-----------|
           S |       102 |       819 |       921 |
             |     0.003 |     0.023 |           |
-------------|-----------|-----------|-----------|
Column Total |       808 |     35224 |     36032 |
-------------|-----------|-----------|-----------|
```

The cross-table analysis confirms the same. It says that P was correctly classified as P for 89.4% of the time, I as I only for 1.3% of the time. S was not correctly predicted at all. All S classes were identified as either I or P. Therefore, we can say that the True Positives are 32214+455 = 32669/36032 = 90.66%. False Positives are 2191+251 = 2442 . False Negatives are 102+819 = 921

We put these details in a tabular format and compiled the % of correct and wrong classifications and found that both are almost the same. In our case, the prediction accuracy on test data is at the same level as training.

Computing Percentages of correctly and incorrectly classified observations during training and testing					
Model trained on	Obs in the Dataset	Number of correct classifications	% Correct classifications for all classes	Number of incorrect classifications	% of incorrect classifications for all classes put together
Training Data	108098	1429+96591= 98020	90.68%	746+364+2428+6540=10078	9.32%
Deployed on Test data	36032	455+32214=32669	90.67%	2191+251+102+819=3363	9.33%

Note: We performed Run-2 for Decision Trees after infusing new variables from the existing dataset. This part of the analysis is presented after Boosting with this set of variables and is discussed below.

c . Apply boosting with different numbers of trees and analyze the impact on the prediction results: what is the impact of the number of trees in the accuracy of the classifier?

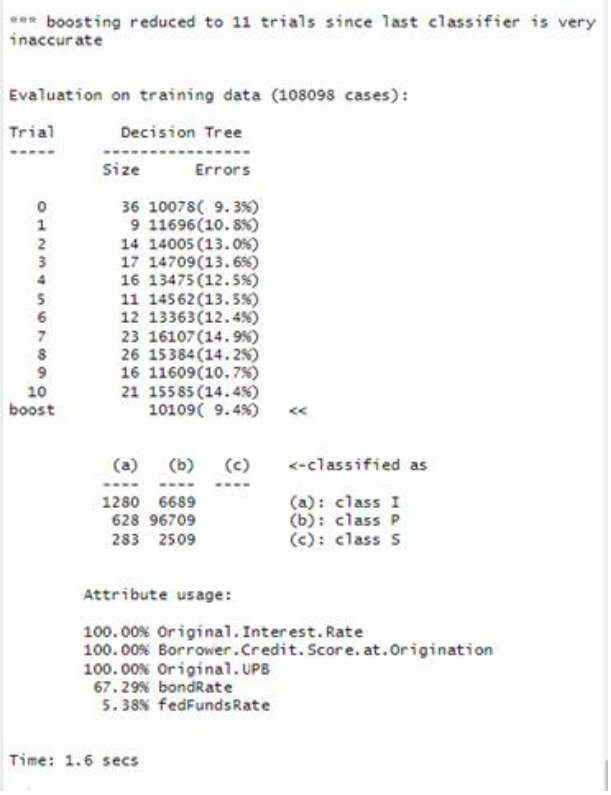
Applying Boosting on the training model with 10 trees: (We are producing the results which we will capture in tabular form and analyze

```
Evaluation on training data (108098 cases):

Trial        Decision Tree
-----      ----------------
           Size      Errors

   0         36 10078( 9.3%)
   1          9 11696(10.8%)
   2         14 14005(13.0%)
   3         17 14709(13.6%)
   4         16 13475(12.5%)
   5         11 14562(13.5%)
   6         17 13521(12.5%)
   7         20 11844(11.0%)
   8         21 11104(10.3%)
   9          8 10365( 9.6%)
boost           10183( 9.4%)   <<


            (a)   (b)   (c)    <-classified as
           ----  ----  ----
           1008  6961          (a): class I
            430 96907          (b): class P
            211  2581          (c): class S


         Attribute usage:

         100.00% Original.Interest.Rate
          68.35% Borrower.Credit.Score.at.Origination
          68.35% Original.UPB
          64.22% bondRate
           5.38% fedFundsRate
```

Boosting with 20 trees:

```
Evaluation on training data (108098 cases):

Trial        Decision Tree
-----      ----------------
           Size      Errors

   0         36 10078( 9.3%)
   1          9 11696(10.8%)
   2         14 14005(13.0%)
   3         17 14709(13.6%)
   4         16 13475(12.5%)
   5         11 14562(13.5%)
   6         12 13363(12.4%)
   7         23 16107(14.9%)
   8         26 15384(14.2%)
   9         16 11609(10.7%)
  10         21 15585(14.4%)
  11         15 11743(10.9%)
  12         30 17614(16.3%)
  13         28 13863(12.8%)
  14         20 12808(11.8%)
  15         31 11021(10.2%)
  16         17 10925(10.1%)
  17         22 10667( 9.9%)
  18         24 10227( 9.5%)
  19         21 10299( 9.5%)
boost           10181( 9.4%)   <<


            (a)   (b)   (c)    <-classified as
           ----  ----  ----
           1003  6966          (a): class I
            423 96914          (b): class P
            238  2554          (c): class S


         Attribute usage:

         100.00% Original.Interest.Rate
         100.00% Borrower.Credit.Score.at.Origination
         100.00% Original.UPB
          68.97% bondRate
          14.19% fedFundsRate


Time: 2.3 secs
```

Boosting with 100 trees output with error saying boosting reduced to 11 trials since last classifier was inaccurate. Therefore we tried with 30 trees but again the trial got terminated after 11 with the same error.



```
*** boosting reduced to 11 trials since last classifier is very
inaccurate


Evaluation on training data (108098 cases):

Trial       Decision Tree
-----     ----------------
          Size      Errors

   0        36 10078( 9.3%)
   1         9 11696(10.8%)
   2        14 14005(13.0%)
   3        17 14709(13.6%)
   4        16 13475(12.5%)
   5        11 14562(13.5%)
   6        12 13363(12.4%)
   7        23 16107(14.9%)
   8        26 15384(14.2%)
   9        16 11609(10.7%)
  10        21 15585(14.4%)
boost          10109( 9.4%)   <<


          (a)   (b)   (c)    <-classified as
          ----  ----  ----
          1280  6689          (a): class I
           628 96709          (b): class P
           283  2509          (c): class S


        Attribute usage:

        100.00% Original.Interest.Rate
        100.00% Borrower.Credit.Score.at.Origination
        100.00% Original.UPB
         67.29% bondRate
          5.38% fedFundsRate


Time: 1.6 secs
```

We summarize the effect of boosting based on trial data at this point.

ON TRAINING DATA	Actual	Predicted		
No Boosting (1 Decision Tree)		I	P	S
	I	1429	6540	
	P	746	96591	
	S	364	2428	
Boosting with 10 Trees	Actual	Predicted		
		I	P	S
	I	1008	6961	
	P	430	96907	

	S	211	2581	
Boosting with 20 Trees	Actual	Predicted		
		I	P	S
	I	1003	6966	
	P	423	96914	
	S	238	2554	
Boosting terminating after 11 trials	Actual	Predicted		
		I	P	S
	I	1280	6689	
	P	628	96709	
	S	283	2509	

Interpretation, of above results:

Without Boosting, the model was correctly interpreting I as I in 1429 cases, P as P in 96591 cases. It is not predicting S at all. With boosting using 10 trials, the model still does not predict S class. It underperformed in predicting I and improved in predicting P. With 20 trees, it became still better at predicting P, but worse at predicting I. It is not predicting S at all. So one understanding we had was boosting techniques seemed to produce better and better results for the largest class 'P" as the models had more observations to train on.

We also understood that our classifier models right from Logistic, Naïve Bayes and decision trees are all not able to predict S class.

Our research showed that the possible reasons for this are:

1. Lack of identifying features (IVs) for S class.
2. The choice of our package C5.0() being basic.

So we decided to try another package called 'rpart' for Decision Tree analysis and boosting and found that the tree returned by rpart() has only one node and basically is not testing any conditions.

```
> modelDT <- rpart(Occupancy.Status ~ Original.Interest.Rate +
bondRate + fedFundsRate +
+                    Borrower.Credit.Score.at.Origination + Ori
ginal.UPB, data = trainDT)
> modelDT
n= 108098

node), split, n, loss, yval, (yprob)
      * denotes terminal node

1) root 108098 10761 P (0.07372014 0.90045144 0.02582841) *
>
```

Model summary revealed the probabilities for each class. From this result we understood that one reason for our Decision Tree not being able to detect S class is because of the low probability of finding the set of observations for class S and training on them.

```
> library(rpart)
> modelDT <- rpart(Occupancy.Status ~ Original.Interest.Rate + bondRate + fedFundsRate +
+                    Borrower.Credit.Score.at.Origination + Original.UPB, data = trainDT,
+                  method = "class")
> summary(modelDT)
Call:
rpart(formula = Occupancy.Status ~ Original.Interest.Rate + bondRate +
    fedFundsRate + Borrower.Credit.Score.at.Origination + Original.UPB,
    data = trainDT, method = "class")
  n= 108098

          CP nsplit rel error xerror xstd
1 0.008239631      0         1      0    0

Node number 1: 108098 observations
  predicted class=P  expected loss=0.09954856  P(node) =1
    class counts:  7969 97337  2792
   probabilities: 0.074 0.900 0.026
```

We proceeded to use boosting techniques on predictions made on test data. We produce the results here, which we summarized in a table for analysis.

Our predictions on the Decision Tree model without boosting:

```
> modelDT <- C5.0(Occupancy.Status~ Original.Interest.Rate+bondRate+fedF
undsRate+
+                    Borrower.Credit.Score.at.Origination+Original.UPB, d
ata= trainDT)
> predictions <- predict(modelDT, newdata = testDT)
> CrossTable(testDT$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |       455 |      2191 |      2646 |
             |     0.013 |     0.061 |           |
-------------|-----------|-----------|-----------|
           P |       251 |     32214 |     32465 |
             |     0.007 |     0.894 |           |
-------------|-----------|-----------|-----------|
           S |       102 |       819 |       921 |
             |     0.003 |     0.023 |           |
-------------|-----------|-----------|-----------|
Column Total |       808 |     35224 |     36032 |
-------------|-----------|-----------|-----------|
```

Our predictions on the Decision Tree model with boosting, trials 10:

```
> predictions10 <- predict(boost10ModelDT, newdata = testDT)
> CrossTable(testDT$Occupancy.Status, predictions10,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |       318 |      2328 |      2646 |
             |     0.009 |     0.065 |           |
-------------|-----------|-----------|-----------|
           P |       142 |     32323 |     32465 |
             |     0.004 |     0.897 |           |
-------------|-----------|-----------|-----------|
           S |        48 |       873 |       921 |
             |     0.001 |     0.024 |           |
-------------|-----------|-----------|-----------|
Column Total |       508 |     35524 |     36032 |
-------------|-----------|-----------|-----------|
```

Predicting with 20 Boosting Decision Tree model:

```
> boost20ModelDT <- C5.0(trainDT[-6], trainDT$Occupancy.Status, trial =2
0)
> predictions20 <- predict(boost20ModelDT, newdata = testDT)
> CrossTable(testDT$Occupancy.Status, predictions20,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  36032


             | predicted Class
actual Class |         I |         P | Row Total |
-------------|-----------|-----------|-----------|
           I |       308 |      2338 |      2646 |
             |     0.009 |     0.065 |           |
-------------|-----------|-----------|-----------|
           P |       136 |     32329 |     32465 |
             |     0.004 |     0.897 |           |
-------------|-----------|-----------|-----------|
           S |        60 |       861 |       921 |
             |     0.002 |     0.024 |           |
-------------|-----------|-----------|-----------|
Column Total |       504 |     35528 |     36032 |
-------------|-----------|-----------|-----------|
```

When we predicted a boosting of 50 trials, we got the predictions for S class!

```
> boost50ModelDT <- C5.0(trainDT[-6], trainDT$Occupancy.Status, trial =5
0)
>
> predictions50 <- predict(boost50ModelDT, newdata = testDT)
>
> CrossTable(testDT$Occupancy.Status, predictions50,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))

   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

              | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       395 |      2250 |         1 |      2646 |
             |     0.011 |     0.062 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       210 |     32255 |         0 |     32465 |
             |     0.006 |     0.895 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        73 |       848 |         0 |       921 |
             |     0.002 |     0.024 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |       678 |     35353 |         1 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

But when we increased boosting to 100, we got the exactly the same numbers indicating that the model has reached optimal performance with 50 trials.

```
> boost100ModelDT <- C5.0(trainDT[-6], trainDT$Occupancy.Status, trial =
100)
> predictions100 <- predict(boost100ModelDT, newdata = testDT)
> CrossTable(testDT$Occupancy.Status, predictions100,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))

   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

              | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       395 |      2250 |         1 |      2646 |
             |     0.011 |     0.062 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       210 |     32255 |         0 |     32465 |
             |     0.006 |     0.895 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        73 |       848 |         0 |       921 |
             |     0.002 |     0.024 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |       678 |     35353 |         1 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

We summarize hereunder, the improvement in the model with increasing number of trials till it reached optimal performance.

PREDICTIONS WITH TEST DATA	Actual	Predicted		
No Boosting (1 Decistion Tree)		I	P	S
	I	455	2191	
	P	251	32214	
	S	102	819	
Boosting with 10 Trees	Actual	Predicted		
		I	P	S
	I	318	2328	
	P	142	32323	
	S	48	873	
Boosting with 20 Trees	Actual	Predicted		
		I	P	S
	I	308	2338	
	P	136	32329	
	S	60	861	
Boosting with 50 and 100 Trees shows same result	Actual	Predicted		
		I	P	S
	I	395	2250	1
	P	210	32255	0
	S	73	848	0

Summary : When boosting was performed on test data, the model improved its performance in predicting class ‘P” and deteriorated in predicting Class “I”. It started predicting values for S class for trials = 50 but did not show improvement in predicting S class after that. *This indicates that Boosting techniques in*

Decision Tree models show marked improvement in learning based on the most occurring class in the dataset.

Run-2: For this run, we first added the "Property Type" and "Loan Purpose" variables the IV used in the previous run to train our model. In the previous run, our classifier could not predict "S" class at all without boosting. In this run, we find that even without boosting, our model could predict "S" class.

Model and results:

```
> modelDT <- C5.0(Occupancy.Status~ Original.Interest.Rate+bondRate+fedF
undsRate+ Loan.Purpose+Property.Type+
+                    Borrower.Credit.Score.at.Origination+Original.UPB, d
ata= trainDT)
> summary(modelDT)

Call:
C5.0.formula(formula = Occupancy.Status ~ Original.Interest.Rate
 + Borrower.Credit.Score.at.Origination + Original.UPB, data
 = trainDT)


C5.0 [Release 2.07 GPL Edition]         Sun Oct 22 11:44:40 2023
-------------------------------

Class specified by attribute `outcome'

Read 108098 cases (8 attributes) from undefined.data
```

Evaluation on Training data shows that the model predicts I as I for 1588 times, P as P 96719 times and S as S 92 times.

```
Evaluation on training data (108098 cases):

           Decision Tree
         ----------------
         Size      Errors

          130 9699( 9.0%)   <<


           (a)   (b)   (c)    <-classified as
          ----  ----  ----
          1588  6353    28    (a): class I
           582 96719    36    (b): class P
           294  2406    92    (c): class S


        Attribute usage:

        100.00% Original.Interest.Rate
         50.13% bondRate
         17.67% Borrower.Credit.Score.at.Origination
         16.35% Property.Type
         12.58% Original.UPB
          4.54% Loan.Purpose
```

.

Predictions on test data and the confusion matrix reveal that the model predicts True positives for 90.75% of the time, which is an improvement over both Logistic and Naive Bayes models.:

```
> predictions <- predict(modelDT, newdata = testDT)
>
> CrossTable(testDT$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))

   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

             | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       475 |      2141 |        30 |      2646 |
             |     0.013 |     0.059 |     0.001 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       240 |     32209 |        16 |     32465 |
             |     0.007 |     0.894 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        90 |       817 |        14 |       921 |
             |     0.002 |     0.023 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |       805 |     35167 |        60 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

We tried boosting with trial =10 and found that the model improved in predicting P class but deteriorated in predicting “I” and “S” classes.

```
> boost10ModelDT <- C5.0(trainDT[-8], trainDT$Occupancy.Status, trial =10)
> predictions <- predict(boost10ModelDT, newdata = testDT)
> CrossTable(testDT$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))

   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

             | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       466 |      2169 |        11 |      2646 |
             |     0.013 |     0.060 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       219 |     32242 |         4 |     32465 |
             |     0.006 |     0.895 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        87 |       827 |         7 |       921 |
             |     0.002 |     0.023 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |       772 |     35238 |        22 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

We tried boosting with 20 Trees and found that the model was less effective in predicting “S” class and also deteriorated in predicting “P” class.

```
> predictions <- predict(boost20ModelDT, newdata = testDT)
> CrossTable(testDT$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))

   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

             | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       468 |      2169 |         9 |      2646 |
             |     0.013 |     0.060 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       235 |     32219 |        11 |     32465 |
             |     0.007 |     0.894 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        91 |       827 |         3 |       921 |
             |     0.003 |     0.023 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |       794 |     35215 |        23 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

Summary: In run-2 with the new addition of variables, the Decision Tree is more efficient in Predicting all classes of the outcome variable. However, with boosting it improved in predicting “P” class to an extent but not in predicting other classes. This confirms the analysis from Run-1 that the classifier learns better on the most occurring class of the Outcome variable.

d **. Using bagging and Random Forest techniques :**

Perform the analysis with bagging and random forests: train with bagging and then train with random forests using at least four different numbers of trees. Compare prediction results over the testing sets. Which are the most important features in each random forest?

We have reached the final part of this Milestone, and in the analysis so far, we have only fitted Linear Models to answer our first and second research questions to find out the best predictive features for our ‘Original Interest Rate’ variable. The distributions of these linear models were not exactly Gaussian. Also, the best performing model’s R-squared was 0.70 which is just hitting the threshold of being a ‘good fit’. Therefore, in this part of our analysis, we intend to find out conclusively, which of our variables have the most importance in predicting the ‘original Interest rate’ of the loan.

Bagging and Random Forests are ensembles of Decision Trees . Bagging stands for Bootstrap aggregating and is used to improve the performance and reduce the variance of a predictive model. The bagging model creates multiple subsets of the training dataset using bootstrap sampling, to train on a separate instance of the model. Then it aggregates the predictions from individual models to make a final class prediction. The key idea behind bagging is to increase robustness of the predictive model by reducing overfitting. It tends to have better generalization performance on unseen data because of which the accuracy of the model is improved. Random Forests add feature randomization technique to bagging. At each node of a decision tree, instead of considering all features to make a split a random subset of features is considered for splitting. This

introduces diversity among trees and enhances prediction accuracy. Therefore, in Bagging we use all the IVs in the dataset and in Random Forest technique, we specify the number of variable to be used using the 'mtry' argument. Here is our list of IVs and DV for this part of the analysis.

Bagging and Random Forest Model, IVs and DV			
IVs	Type and what it means	DV	Type and what it means
bondRate	Numeric, discrete, 10-year Treasury Bond Yield Rate	Original Interest Rate	Numeric, Continuous, The Note rate on the Loan at the time of sanctioning the loan to the borrower.
fedFundsRate	Numeric, discrete, Federal Funds Rate, this is the rate at which lending institutions borrow funds overnight (Cost of funds)		
Borrower Credit Score	Numeric, discrete, Credit Score of the borrower		
Original UPB	Numeric, discrete, Loan quantum		

We have 144130 observations in our dataset which took an inordinately long time to compile and run the Random Forest. After two or three attempts to run the techniques with the entire dataset and having to terminate R after 20 minutes, we decided to create a random sample of our dataset and proceed with this part of the analysis. *At this point we also understood that to have a graph with clearer patterns, a random sample of the dataset is the best option.* We created a random sample of our dataset with 1000 observations and proceeded to verify the effect of bagging and random forests on our dataset and identify the most important features of our dataset that help predict the outcome variable.

```
> data <- read.csv("New_FM2022Q4.csv")
> dim(data)
[1] 144130     16
> library(dplyr)
> data <-  subset(data,
+   select=c(Original.Interest.Rate,bondRate,fedFundsRate,Borrower.Credit.Score.at.Origination,Original.UPB))
>
> write.csv(data, file = "RF_FM2022Q4.csv", row.names = FALSE)
> dataRF <- read.csv("RF_FM2022Q4.csv")
> dim(dataRF)
[1] 144130      5
>
> #########SAMPLING FROM ABOVE DATA#########
> set.seed(123)
> sample_size <- 1000
> sampledataRF <- dataRF[sample(nrow(dataRF), sample_size, replace = FALSE), ]
> dim(sampledataRF)
[1] 1000    5
>
> # attach(Boston)
> # View(Boston)
> set.seed(1)
> trainRowIndices <-sample(1:nrow(sampledataRF),nrow(sampledataRF)/2)
> trainRF <- sampledataRF[trainRowIndices, ]
> dim(trainRF)
[1] 500   5
> testRF <- sampledataRF[-trainRowIndices,]
> dim(testRF)
[1] 500   5
>
```

Bagging:

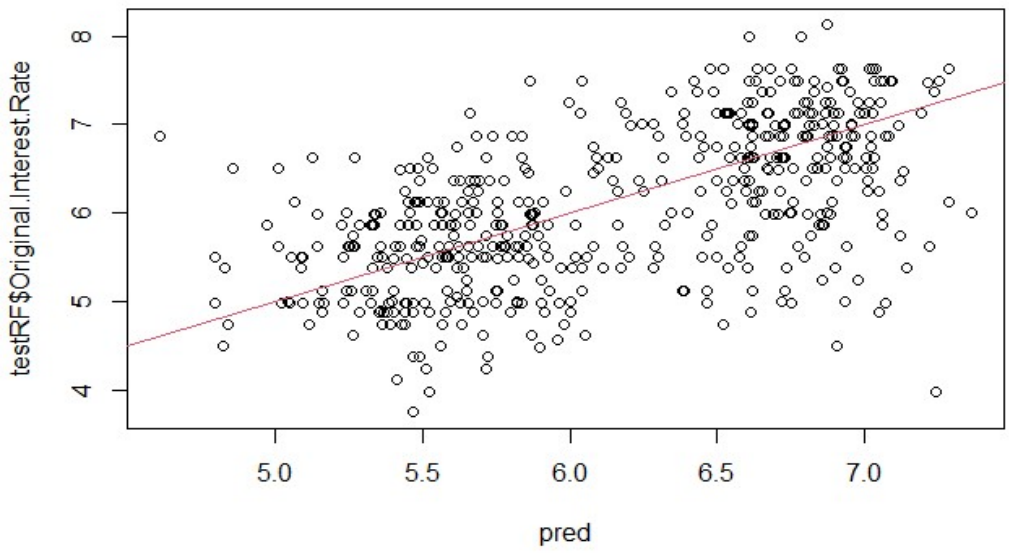
Bagging involves using all the variables in the dataset to train the model and predict using it, whereas the Random Forest method uses fewer variables. We first produced the 'baggingModel' and then the 'ModelRF' with default values for tree number which is '500'. to check the improvement.

The number of trees created was 500. The number of variables used by the model was 4 (IVs). We found correlation was 0.5707298 between the 4 IVs we selected and a very small mean of squared residuals and the % of variability explained was 33.38% with this model.

```
> baggingModel <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=5,impor
tance=TRUE)
Warning message:
In randomForest.default(m, y, ...) :
  invalid mtry: reset to within valid range
> baggingModel

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,
mtry = 5, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 0.479158
                    % Var explained: 33.38
> pred <- predict(baggingModel, newdata= testRF)
> cor(pred, testRF$Original.Interest.Rate)
[1] 0.5707298
> mean((pred- testRF$Original.Interest.Rate)^2)
[1] 0.5312899
```

Plotting the predicted and actual values, we find that there appears a general pattern in the values which is good. We will first present the data and then summarize the results in the end in a tabular form for analysis.

Plotting predicted and actual values indicates that the values are more dispersed in bagging models.

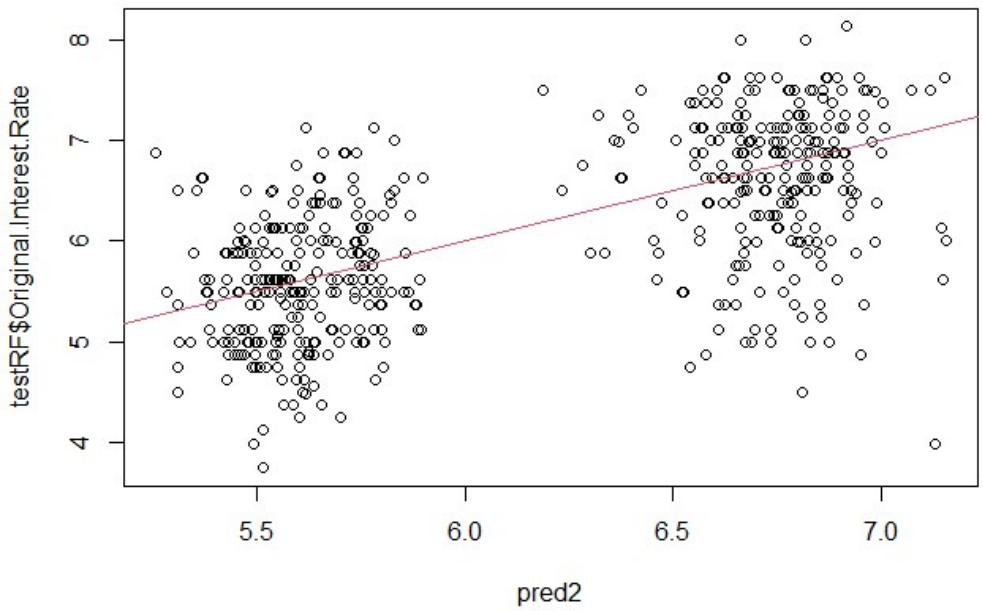


Random Forest Models:

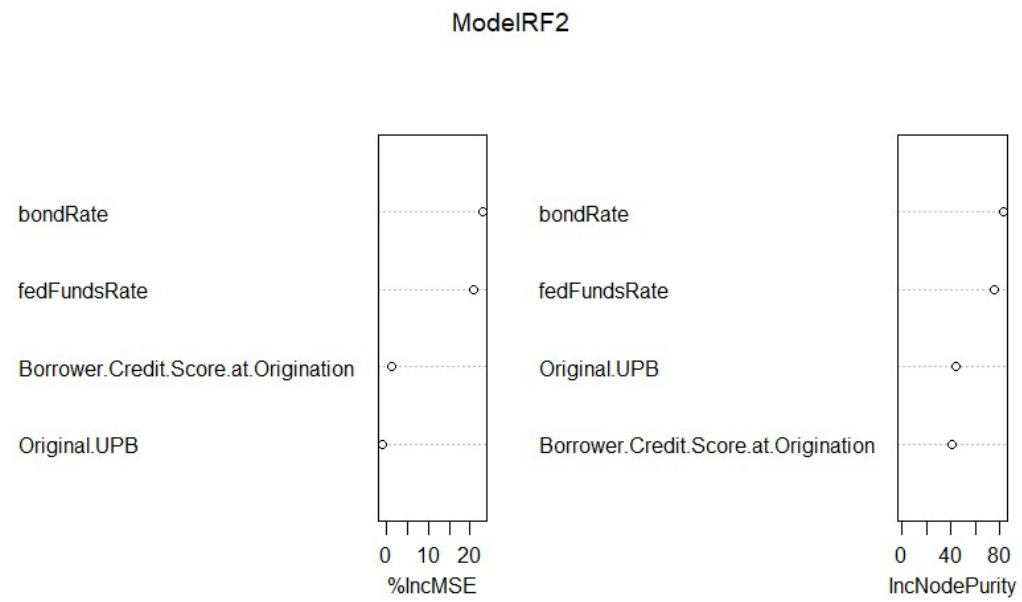
Our Model for training on Random Forest (2 variables) and the output, shows that the correlation is higher and the MSE lower for this model:

```
> ModelRF2 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=2,importance=TRUE)
> ModelRF2

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mtry
= 2, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 0.4351489
                    % Var explained: 39.49
>
> pred2 <- predict(ModelRF2, newdata= testRF)
> cor(pred2, testRF$Original.Interest.Rate)
[1] 0.6100299
> mean((pred2- testRF$Original.Interest.Rate)^2)
[1] 0.4796074
>
```

Plotting predicted and actual values with the Random Forest (2 variables) Model, we find that the values are closer with this approach.



Interpretation: there is significant improvement in the Random Forest model both in terms of MSE and variability explained. The plot between predicted and actual values shows a clearer positive linear relationship. We will plot the important features for this Random Forest model using varImpPlot() function. In R.



Interpretation: The above plot shows the importance of IVs in two parameters namely increase in MSE and increase in Node Purity. The names of the IVs in the model are shown on the y-axes and the % increase in MSE and increase in node purity are shown on the x-axes. For both criteria, variables with higher important scores are identified by the model as more influential in improving the model's performance, which of course can be equated to its ability to predict the outcome variable accurately. The left bar in the above plot determines the extent to which randomly permuting those IVs increases MSE in the order of values on x-axis. According to the left Bar plot, the bondRate variable has the highest effect on increase of MSE and decrease in accuracy when randomly permuted. The bar plot on the right side measures the level of node purity. The bondRate variable has the highest value and is the most important in this parameter also, indicating its ability to maintain node purity by splitting the data into more pure subsets. Next comes bondRate and UPB and Borrower credit score are next in this order.

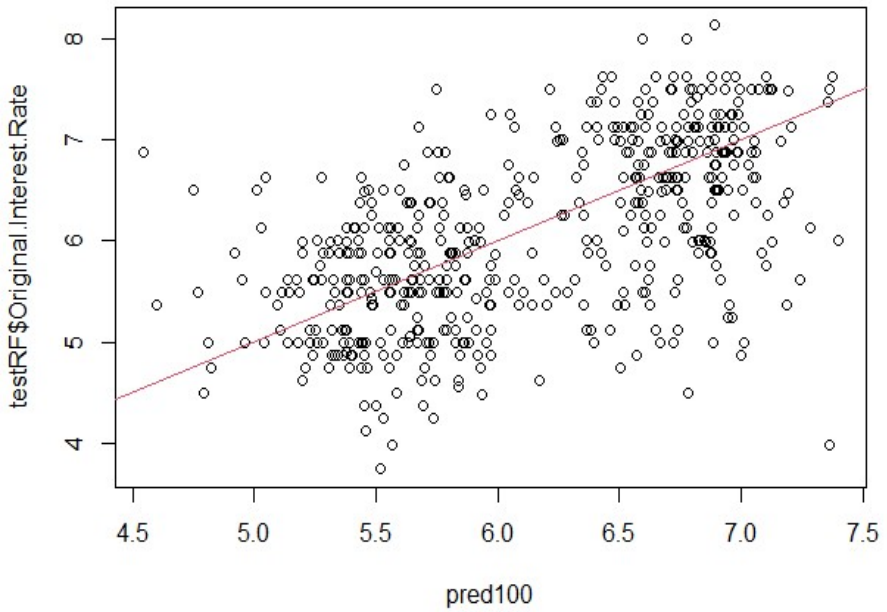
If the same pattern is observed in all the subsequent variable importance plots for all the models of Random Forest we created, we can interpret that the 'bondRate' is the most predictive feature for predicting the values of the outcome variable 'Original interest rate' in our dataset closely followed by 'fedFundsRate'. Interestingly, none of the variables in our original dataset, including Borrower Credit Score and Original UPB, are as important as these two variables we infused after phase- multivariate regression analysis.

We repeat the Bagging and Random Forest experiments with the number of trees = 100.

Bagging:

```
> baggingModel100 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=5,importance=TR
UE, ntree=100)
Warning message:
In randomForest.default(m, y, ...) :
  invalid mtry: reset to within valid range
> baggingModel100

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mt
ry = 5, importance = TRUE, ntree = 100)
               Type of random forest: regression
                     Number of trees: 100
No. of variables tried at each split: 4

          Mean of squared residuals: 0.4890123
                    % Var explained: 32.01
>
> pred100 <- predict(baggingModel100, newdata= testRF)
> cor(pred100, testRF$Original.Interest.Rate)
[1] 0.5662334
> mean((pred100- testRF$Original.Interest.Rate)^2)
[1] 0.5378488
>
> plot(pred100,testRF$Original.Interest.Rate)
> abline(c(0,1),col=2)
```

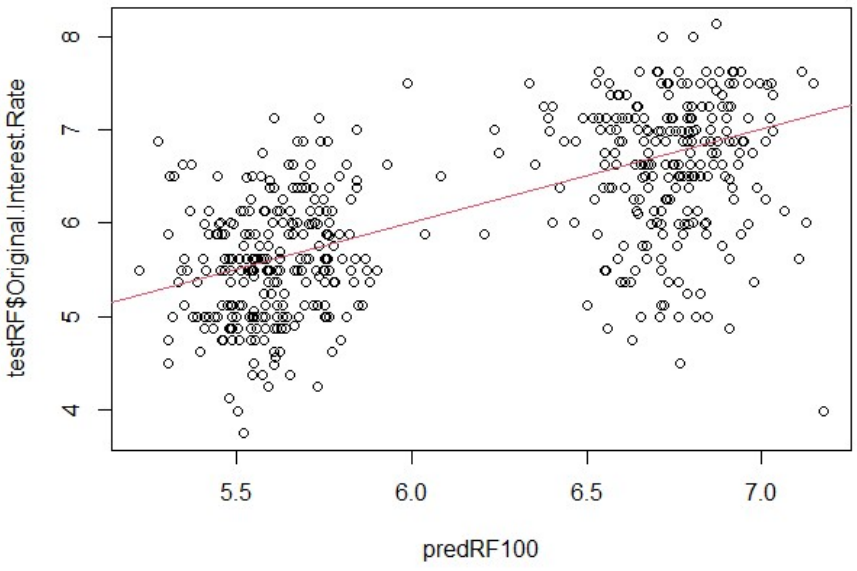
Plotting predicted and actual values indicates that the values are more dispersed in bagging models.



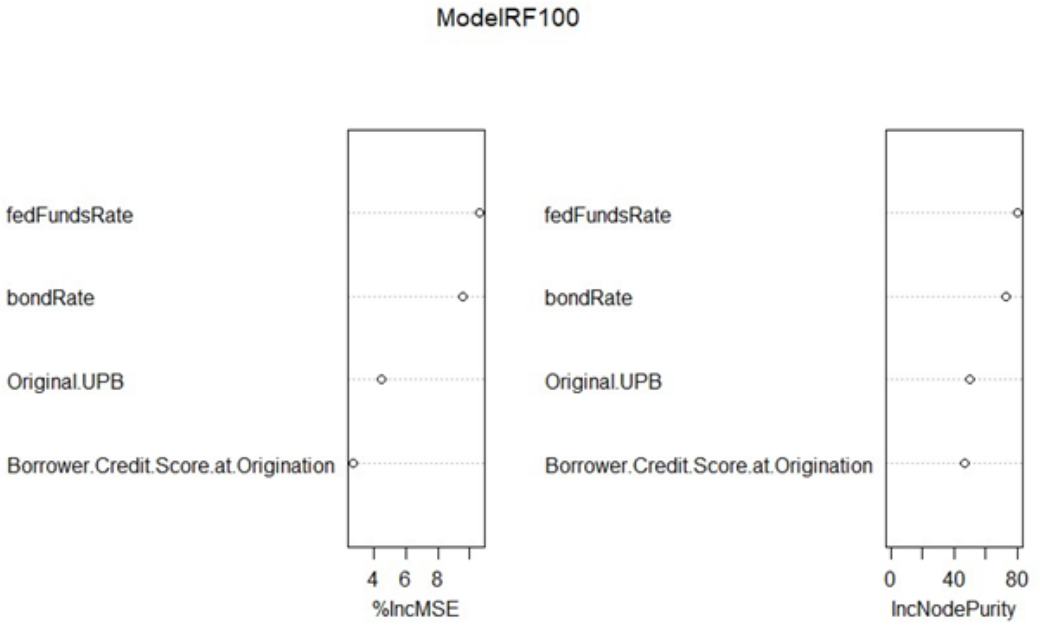
RandomForest with 2 variables, Number of trees = 100:

```
> ModelRF100 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=2,importance=TR
UE, ntree=100)
> ModelRF100

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mt
ry = 2, importance = TRUE, ntree = 100)
               Type of random forest: regression
                     Number of trees: 100
No. of variables tried at each split: 2

          Mean of squared residuals: 0.4386137
                    % Var explained: 39.01
>
> predRF100 <- predict(ModelRF100, newdata= testRF)
> cor(predRF100, testRF$Original.Interest.Rate)
[1] 0.6087091
> mean((predRF100- testRF$Original.Interest.Rate)^2)
[1] 0.4796896
>
> plot(predRF100, testRF$Original.Interest.Rate)
> abline(c(0,1),col=2)
> varImpPlot(ModelRF100)
```

Plotting predicted and actual values in the test dataset shows the values are closer in the RF model compared to Bagging model.



Important Variables Plot for random forest model trees =100



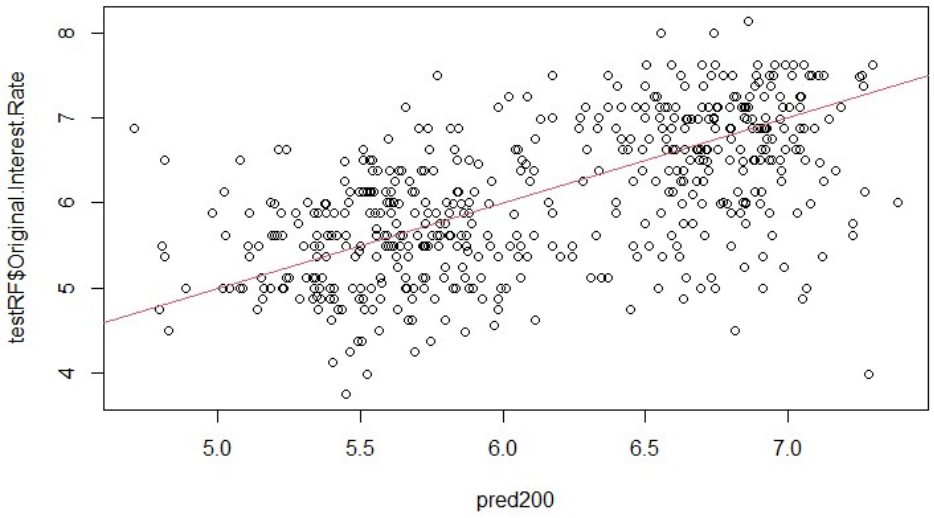
The plot shows the order of importance of variables as fedFundsRate followed by bondRate in predicting the outcome variable.

We repeat this experiment with the number of trees = 200.

Bagging:

```
> baggingModel200 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=5,importance=TRUE, ntree=200)
Warning message:
In randomForest.default(m, y, ...) :
  invalid mtry: reset to within valid range
> baggingModel200

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mtry = 5, import
ance = TRUE, ntree = 200)
               Type of random forest: regression
                     Number of trees: 200
No. of variables tried at each split: 4

          Mean of squared residuals: 0.487408
                    % Var explained: 32.23
>
> pred200 <- predict(baggingModel200, newdata= testRF)
> cor(pred200, testRF$Original.Interest.Rate)
[1] 0.5719019
> mean((pred200- testRF$Original.Interest.Rate)^2)
[1] 0.5303927
```

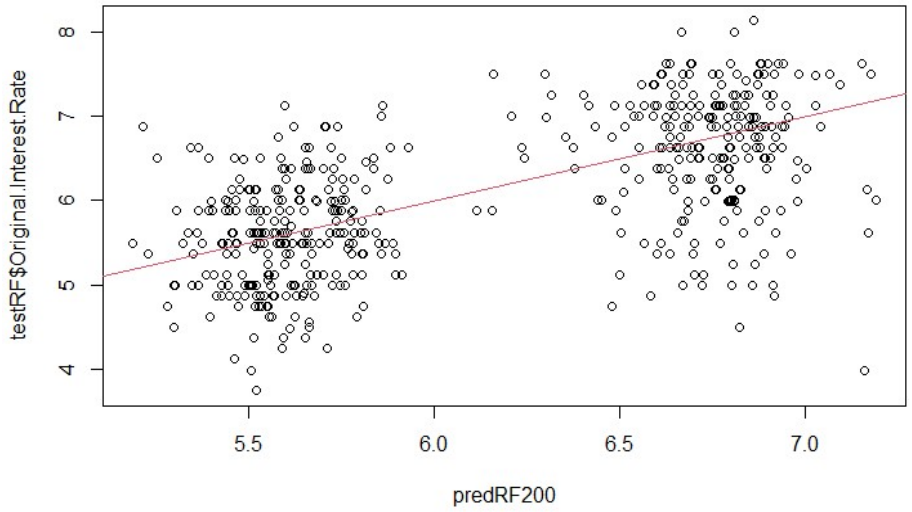
Plotting predicted and actual values indicates that the values are more dispersed in bagging models.



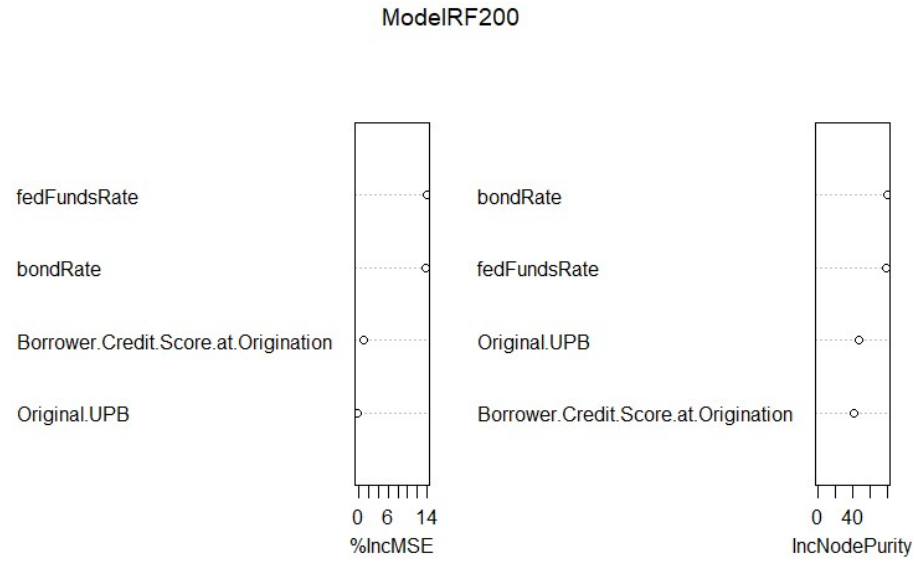
Random Forest, Number of Trees = 200

```
> ModelRF200 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=2,importance=TRUE, ntree=200)
> ModelRF200

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mtry = 2, import
ance = TRUE, ntree = 200)
               Type of random forest: regression
                     Number of trees: 200
No. of variables tried at each split: 2

          Mean of squared residuals: 0.434862
                    % Var explained: 39.53
>
> predRF200 <- predict(ModelRF200, newdata= testRF)
> cor(predRF200, testRF$Original.Interest.Rate)
[1] 0.6061396
> mean((predRF200- testRF$Original.Interest.Rate)^2)
[1] 0.4833997
>
> plot(predRF200, testRF$Original.Interest.Rate)
> abline(c(0,1),col=2)
>
```

Plotting predicted and actual values in the test dataset shows the values are closer in the RF model compared to Bagging model.



Important Variables Plot for random forest model trees =200

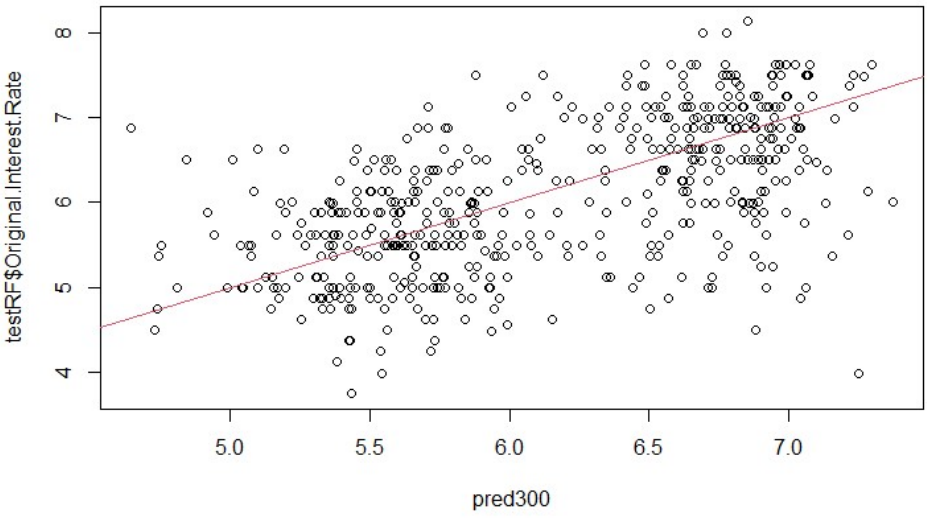


Experiment with Number of Trees = 300

Bagging:

```
> baggingModel300 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=5,importance=TRUE, ntree=300)
Warning message:
In randomForest.default(m, y, ...) :
  invalid mtry: reset to within valid range
> baggingModel300

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mtry = 5, import
ance = TRUE, ntree = 300)
               Type of random forest: regression
                     Number of trees: 300
No. of variables tried at each split: 4

          Mean of squared residuals: 0.4840344
                    % Var explained: 32.7
>
> pred300 <- predict(baggingModel300, newdata= testRF)
> cor(pred300, testRF$Original.Interest.Rate)
[1] 0.5757962
> mean((pred300- testRF$Original.Interest.Rate)^2)
[1] 0.5273337
>
> plot(pred300,testRF$Original.Interest.Rate)
> abline(c(0,1),col=2)
>
```

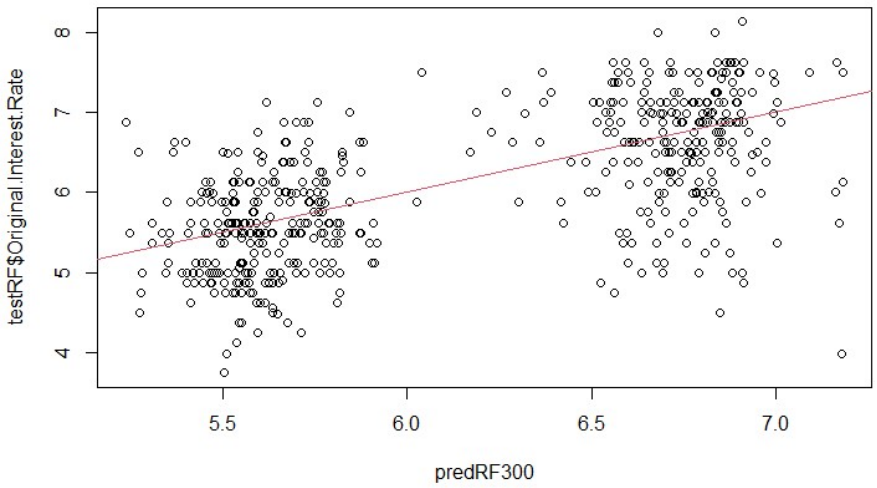
Plotting predicted and actual values indicates that the values are more dispersed in bagging models.



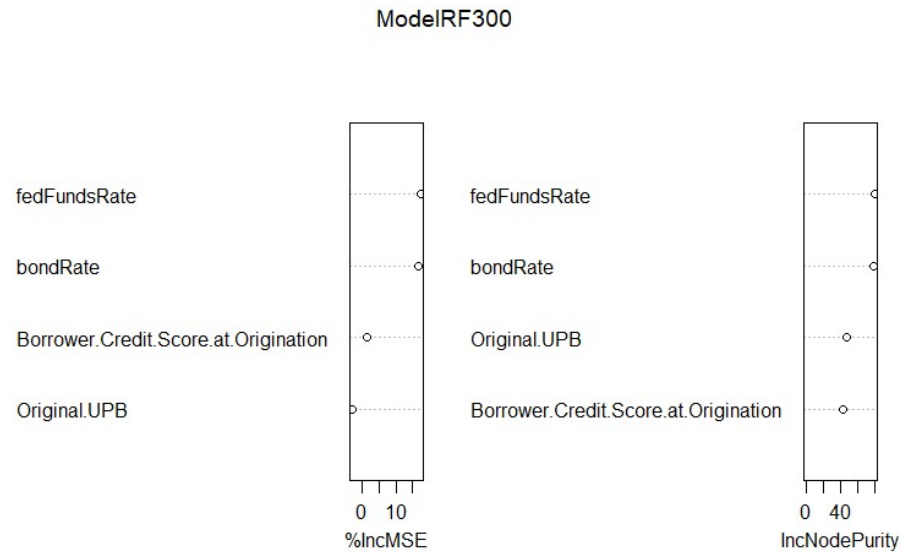
Random Forests, number of trees = 300:

```
> ModelRF300 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=2,importance=TRUE, ntree=300)
> ModelRF300

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mtry = 2, import
ance = TRUE, ntree = 300)
               Type of random forest: regression
                     Number of trees: 300
No. of variables tried at each split: 2

          Mean of squared residuals: 0.4387499
                    % Var explained: 38.99
>
> predRF300 <- predict(ModelRF300, newdata= testRF)
> cor(predRF300, testRF$Original.Interest.Rate)
[1] 0.6054383
> mean((predRF300- testRF$Original.Interest.Rate)^2)
[1] 0.4836421
>
> plot(predRF300, testRF$Original.Interest.Rate)
> abline(c(0,1),col=2)
>
```

Plotting predicted and actual values in the test dataset shows the values are closer in the RF model compared to Bagging model.



Important variables plot number of trees = 300, is ths same as that of trees = 200

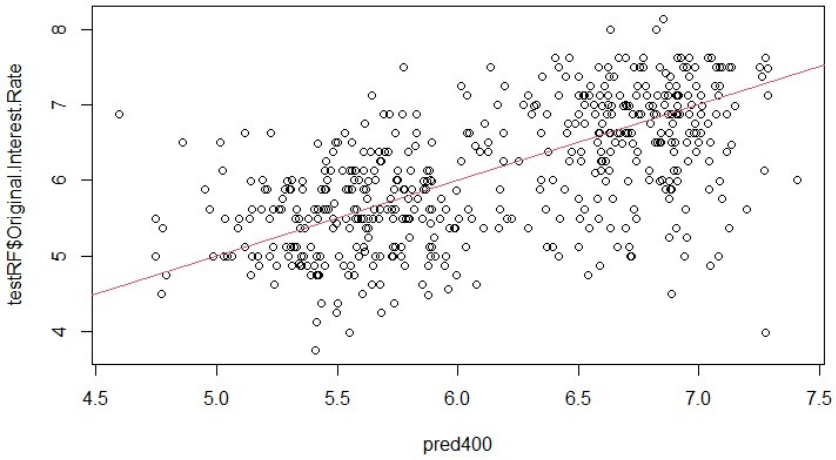


Experiment with Number of Trees = 400:

Bagging:

```
> baggingModel400 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=5,importance=TRUE, ntree=400)
Warning message:
In randomForest.default(m, y, ...) :
  invalid mtry: reset to within valid range
> baggingModel400

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mtry = 5, import
ance = TRUE, ntree = 400)
               Type of random forest: regression
                     Number of trees: 400
No. of variables tried at each split: 4

          Mean of squared residuals: 0.4788854
                    % Var explained: 33.41

>
> pred400 <- predict(baggingModel400, newdata= testRF)
> cor(pred400, testRF$Original.Interest.Rate)
[1] 0.5738547
> mean((pred400- testRF$Original.Interest.Rate)^2)
[1] 0.5296372
>
> plot(pred400,testRF$Original.Interest.Rate)
> abline(c(0,1),col=2)
```

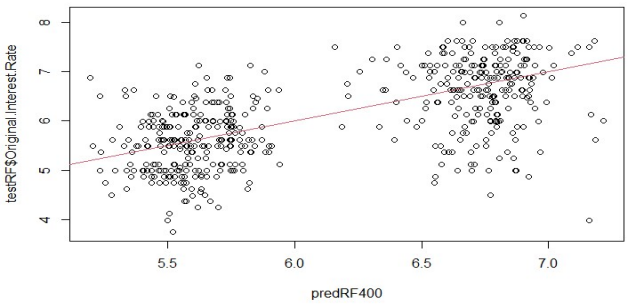
Plotting predicted and actual values:



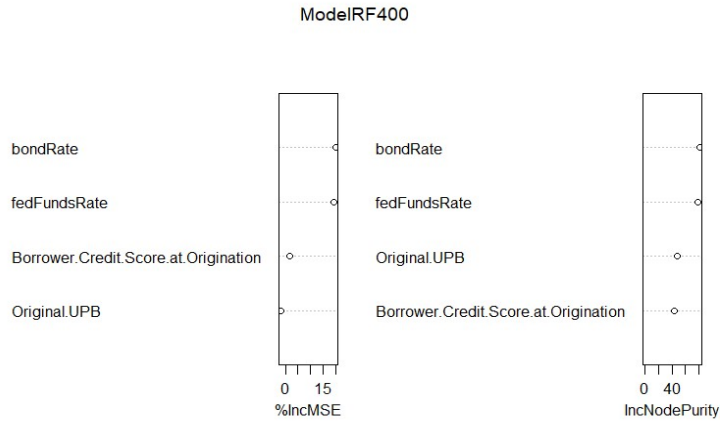
Random Forest, number of trees=400

```
> ModelRF400 <-
+   randomForest(Original.Interest.Rate~.,data=trainRF,mtry=2,importance=TRUE, ntree=400)
> ModelRF400

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = trainRF,      mtry = 2, import
ance = TRUE, ntree = 400)
               Type of random forest: regression
                     Number of trees: 400
No. of variables tried at each split: 2

          Mean of squared residuals: 0.4371114
                    % Var explained: 39.22
>
> predRF400 <- predict(ModelRF400, newdata= testRF)
> cor(predRF400, testRF$Original.Interest.Rate)
[1] 0.6091867
> mean((predRF400- testRF$Original.Interest.Rate)^2)
[1] 0.4803481
>
> plot(predRF400, testRF$Original.Interest.Rate)
> abline(c(0,1),col=2)
```

Plotting predicted and actual values in the test dataset shows the values are closer to some values in the RF model compared to Bagging model.



Important variables plot num of trees = 400



Summary Table:

SUMMARIZING RESULTS BAGGING VS. RANDOM FORESTS FOR ANALYSIS								
NO OF TREES	BAGGING			RF			Improvement in Correlation in Random Forest Models from Bagging models	Most Preditive Features as per Random Forest Analysis(Imp Variables Plot) in Order of importance
	No of Ivs	Cor	MSE	No of Ivs	Cor	MSE		
100	4	0.5662334	0.5378488	2	0.6087091	0.4796896	0.0424757	1.fedFundsRate 2.bondRate 3. Original UPB 4. Borrower Credit Score
200	4	0.5719019	0.5303927	2	0.6061396	0.4833997	0.0342377	same as above
300	4	0.5757962	0.5273337	2	0.60544383	0.4836421	0.02964763	same as above

NO OF TREES	BAGGING			RANDOM FORESTS			Improvement in Correlation in Random Forest Models from Bagging models	Most Preditive Features as per Random Forest Analysis(Imp Variables Plot) in Order of importance
	No of Ivs	Cor	MSE	No of Ivs	Cor	MSE	Diff in Cor	
400	4	0.5738547	0.5296372	2	0.6091867	0.4803481	0.035332	1.bondRate 2.fedFundsRate 3. Borrower Credit Score 4. Original UPB
500	4	0.5707298	0.5312899	2	0.6100299	0.4796074	0.0393001	same as above

Interpretation:

1. As is clearly visible from the above chart, from Bagging to Random Forest, there is a clear improvement in Correlation and MSE values.

2. The order of importance of predictive features, for some runs of Random Forest is slightly different.

    a. In general, bondRate and fedFundsRate have best values for both effect on MSE and impact on purity of subsets at each node.

    b.These two are followed by UPB and Borrower credit Score as variables with predictive power on the outcome variable. While UPB and CS have almost the same score for ability to maintain node purity, CS falls behind in its impact on reduction of MSE.

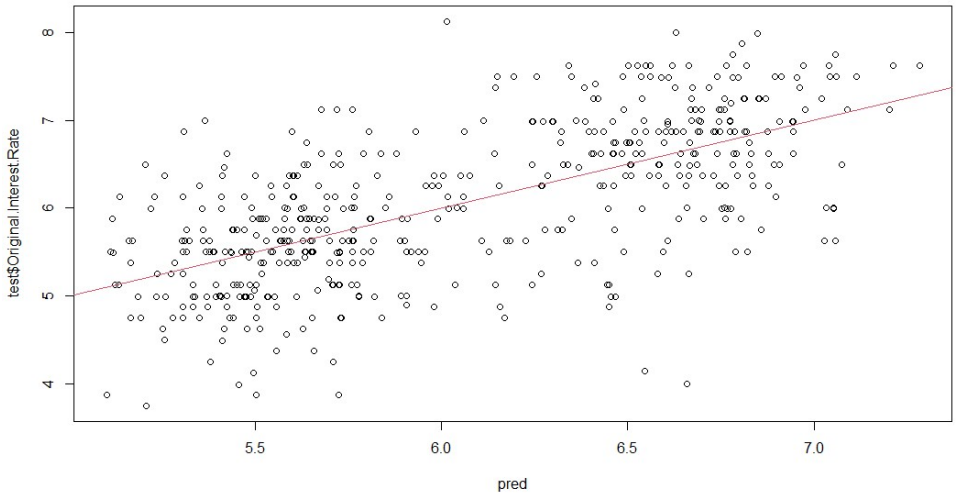
Run2: Bagging and Random Forests: As explained earlier, we intend to find out if adding all the IVs from the new dataset to the mix will improve prediction accuracy for the outcome variable Original Interest Rate. The list of variables in the entire dataset is given at the beginning of this research and is not being repeated here to maintain brevity.

We have 15 IVs and 1DV in the dataset, all of which were used to train the Bagging model.

```
> baggingModel <-
+   randomForest(Original.Interest.Rate~.,data=train,mtry=15,importance=TRUE)
> baggingModel

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = train,      mtry = 15, impo
rtance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 15

          Mean of squared residuals: 0.4321708
                    % Var explained: 38.28
> pred <- predict(baggingModel, newdata= test)
> cor(pred, test$Original.Interest.Rate)
[1] 0.6558487
> mean((pred- test$Original.Interest.Rate)^2)
[1] 0.4365761
```

The correlation coefficient and MSE values are 0.6558487 and 0.4365761 respectively.

Plot for Bagging model, Run-2



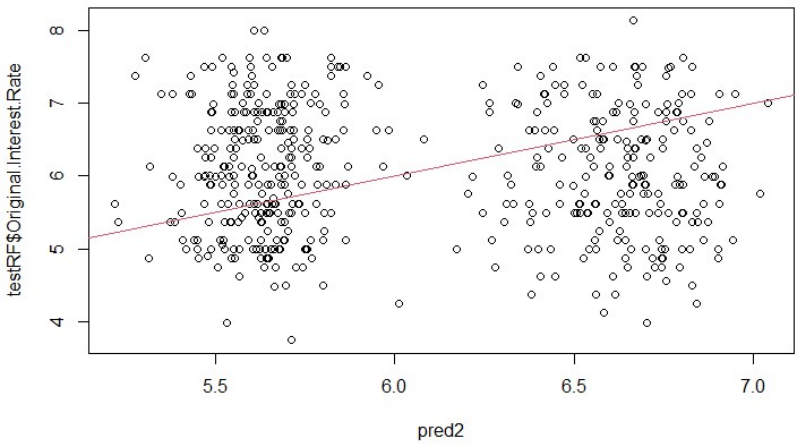
We then trained a Random Forest Model with 4 variables and tested its predictive power to find that it shows superior performance with a correlation of 0.6777353 and MSE value of 0.4160561 over the Bagging model.

```
> ModelRF4 <-
+   randomForest(Original.Interest.Rate~.,data=train,mtry=4,importance=
TRUE)
> ModelRF4

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = train,
mtry = 4, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 0.420266
                    % Var explained: 39.98
>
> pred4 <- predict(ModelRF4, newdata= test)
> cor(pred4, test$Original.Interest.Rate)
[1] 0.6777353
> mean((pred4- test$Original.Interest.Rate)^2)
[1] 0.4160561
```

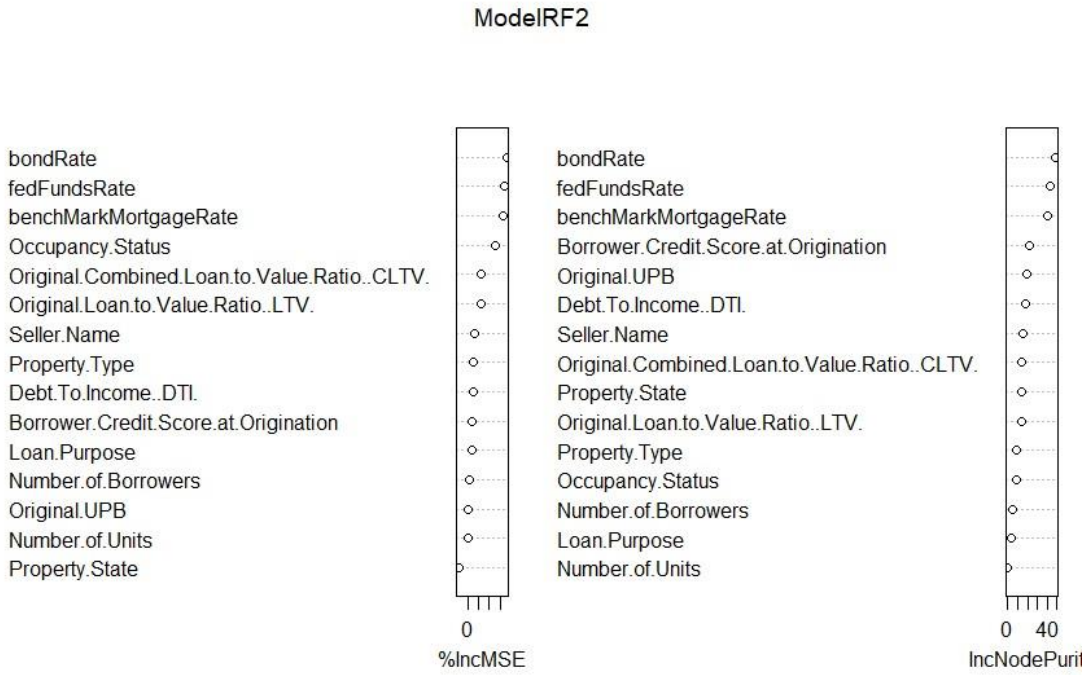
We trained 3 more Random Forest Models with mtry = 2/ 5/ 8 to find out that the model with 2 variables shows the best values for correlation coefficient and MSE.

```
> ModelRF2 <-
+   randomForest(Original.Interest.Rate~.,data=train,mtry=2,importance=
TRUE)
> ModelRF2

Call:
 randomForest(formula = Original.Interest.Rate ~ ., data = train,
mtry = 2, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 0.4123602
                    % Var explained: 41.11
>
> pred2 <- predict(ModelRF2, newdata= test)
> cor(pred2, test$Original.Interest.Rate)
[1] 0.6823664
> mean((pred2- test$Original.Interest.Rate)^2)
[1] 0.4156908
```

Plot for Random Forest Model-2:



We then visualized the important features for predicting the outcome variable 'Original Interest Rate' for the best performing RF model which is RF2. The following plot provides the predictive power of each of the 15 IVs in predicting the outcome variable "Original Interest Rate'. The most predictive feature is bondRate closely followed by fedFundsRate. Since we retained all variables, those that exhibited multicollinearity also in this dataset, we find that benchMarkMortgageRate variable is the next predictive. We had eliminated this variable from most of our analyses because it was multicollinear with fedfundsrate variable.



Question 4. Comparative Analysis.

A summary of all classifiers, their predictive quality and our choices to answer our research question(s).

We reproduce our research questions and analyze which of the models helped us find better quality answers for them.

1. Can we accurately predict the 'Interest Rate' applied to a mortgage loan by a lending institution, using criterion such as the Quantum of loan, Loan-To-Value ratio, Debt-To-Income ratio, Loan Purpose, Number of Borrowers and Credit Score of the Borrower/s, using the Fannie Mae Acquisition and Performance 2022Q4 dataset?

Using Linear Regression : In this part of the analysis, we trained linear models on training datasets for each of the IV and the DV original interest rate, to learn the model's predictive power. We used

metrics such as coefficient magnitudes and the associated p-values, correlation between predicted and actual values using test data and corresponding MSE parameters to assess the predictive power of the models. As per linear regression analysis, the models built around variables listed in this research question and few other variables like Seller Name, Occupancy status, Property State and Property type from the Fannie Mae dataset have very small coefficients and R-squared values. They also had low values for correlation between actual and predicted values and high MSE. These models do not offer sufficient predictive power to accurately predict our outcome variable 'Interest Rate'. So, the answer to this research question at this stage is that we cannot determine the answer to this research question with the Fannie Mae dataset alone. So, we infused 3 new variables bondRate, fedFundsRate and benchMarkMortgageRate and performed linear regression. Unfortunately, we found that benchMarkMortgageRate variable and fedFundsRate variables had multicollinearity due to which linear models could not be trained using both of them. So, in subsequent analyses, we used only fedFundsRate and removed benchMarkMortgageRate.

Multivariate Regression: This part of the analysis included trying combinations of IVs with the same DV Original interest rate, to train models on training data and predict using test data and analyzing the correlation and MSE of the predictions using a variety of functions and graphs. In this part of the analysis, we found that the best model was that which was trained with a combination of bondRate, fedFundsRate,Borrower Credit score, UPB and Seller Name variables. It had the best values for coefficients with statistical significance and also the best R-squared value of 0.7 among all models. Out of this set of variables, the bondRate and fedFundsRate variables had the highest magnitude of coefficients which were statistically significant followed by Credit score and UPB variables.

Bagging and Random Forest Models:

The Bagging and Random Forest models were easier to deploy than the linear and multivariate regression models to draw conclusions from. Bagging involved using all the 4 IVs we identified as most predictive from previous learnings at a time while training the model, whereas we used 2 variables for the parameter 'mtry', to train the Random Forest version of the model. Random Forests are generally more accurate than Bagging models, which was proven in our analysis. In this part of the analysis, we hoped to find conclusive answers to our quest regarding the most predictive features in our dataset, to accurately predict 'Original Interest Rate' fixed on a Loan by a lending institution. We used the new variables bondRate and fedFundsRate along with Borrower Credit Score and Original UPB from the Fannie Mae dataset. These models provided conclusive and better-quality answers that were easier to observe, to our research question in identifying that the fedFundsRate variable, followed by the bondRate variable are the most significant in predicting the 'Interest Rate' fixed by a lending institution on the loan to a borrower rather than features like the Borrowers Credit Score or the quantum of the loan.

Run-2: This time round, instead of using the most predictive non- collinear IVs, we used the entire 15 IVs with the 'mtry' parameter for Bagging and used different values for 'mtry' such as 2/5/8 for training Random Forest models. The best results were obtained with mtry =2 from the Random Forest model. The correlation and MSE values for predictions vs. actual values when deployed on test data were 0.6823664 and 0.4156908 respectively. The best multivariate regression model trained with bondRate + fedFundsRate+ Borrower Credit Score+ UPB+ Seller Name had 0.70510

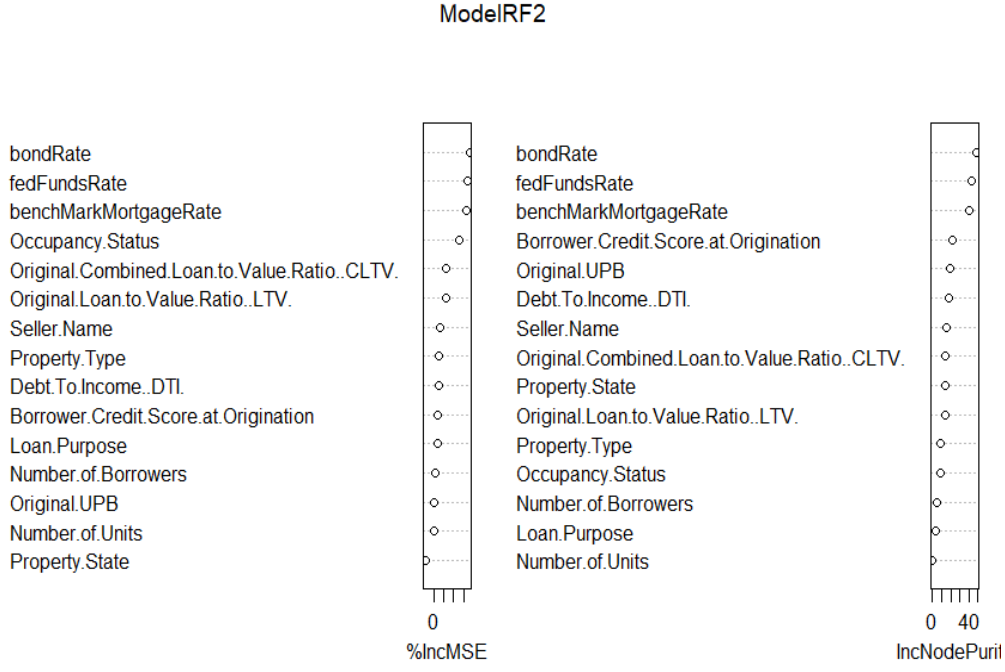
correlation value and 0.690059 MSE values. At this point, since we already tried running Random Forest with 5 variables and did not get the same results, we concluded this analysis.

Conclusion: Random Forest models can deal with multi-collinear, categorical and nominal variables and still deliver superior performance whereas multivariate linear models cannot deal with multicollinearity. Therefore, Random Forests are the model of choice for this kind of analysis.

2. Do all the borrower criteria such as the Quantum of loan, Loan-To-Value ratio, Debt-To-Income ratio, Loan Purpose, Number of Borrowers and Credit Score of the Borrower/s have equal influence over 'Interest Rate'? If not, which criteria have more effect on Interest Rate?

Linear regression with univariate and multivariate models: During this part of the analysis, we find that the Quantum of Loan (UPB) and Borrower credit score variables have statistically significant but mild inverse effect on Interest rate variable, because of very small coefficients and R-squared values of the linear models. The other variables have a much smaller, unequal positive impact on the outcome variable 'Original Interest Rate'. The only IV in the dataset that has a slightly better impact is the Seller Name variable where the coefficients of different levels in a linear regression model range from (0.23 to 1.15) with statistical significance. But the R-squared of the model is 0.11 which is insignificant. So, the answer to this research question is that each of the variables have a statistically significant but very mild predictive power in predicting the outcome variable 'original interest rate' of the loan.

Random Forest Model: This model offered confirmation for our second research question in a more intuitive manner. We found that after infusing new variables into our dataset and deploying Bagging and Random Forest techniques, we could clearly determine the features with most predictive power as those that were not in the Fannie Mae dataset but the bondRate, Federal Funds Rate and benchMarkMortgageRate variables.



Conclusion: The Random Forest model with different numbers of trees predicted different features are most predictive. But since the best model was ModelRF2, the most predictive features for evaluating the outcome variable are bondRate, followed by fedFundsRate, followed by benchMarkMortgageRate.

3. Original: Do all lending institutions (presumed to be the 'Seller' in the dataset), evaluate the criterion variables similarly and fix the interest rate? (This question was revised because Milestone-1 analysis revealed that most values fall under the 'unknown' category named as 'other'. This may or may not provide accuracy in training the models.) Therefore, we revised our research question to,

Revised: *'Can we accurately predict the class of Occupancy Status in the Loan dataset, based on the IVs, Original interest rate, bondRate, fedFundsRate, Borrower Credit Score and Original UPB (from the revised dataset)?*

Classifiers:

Our third research question requires that we find a classifier that, when trained on the training dataset, can accurately predict the class of observations using the test dataset. The accuracy of the classifier is determined by the summary output for the model in which we find metrics such as coefficients, significance values, attribute usage and other metrics relevant to each classifier and the Confusion Matrix provides the number of True Positives, True Negatives, False Positives and False negatives classified by the model. In explaining how to use the different classifiers we used for analysis in this Milestone, the class teachings, which were our primary source for completing this analysis, used different types of IVs and DVs. By closely following them, we understood which classifier works best in predicting which kind of output variable. Therefore, we are happy to report that in this Milestone, besides finding conclusive answers to all our research questions, we also could estimate which of the models we learned to use have better predictive quality in answering different types of research questions.

We therefore used the same set of IVs and DV (given below) from our new dataset to train models using Logistic regression, Naive Bayes function and Decision Tree method with and without Boosting in the first run. We made two runs with different datasets and benefited from the extra effort by being able to train our Classifiers to predict the elusive "S" class.

Variables set for Classifiers, Logistic Regression, Naive Bayes, Decision Tree Models, Run-1			
IVs	Type and what it means	DV	Type and what it means
Original Interest Rate	Numeric, Continuous, The Note rate on the Loan at the time of sanctioning the loan to the borrower	Occupancy Status	Categorical, nominal, whether the underlying property of the loan is the Primary Home of the Borrower (Class "P"), or Secondary Home of the borrower, (Class "S") or if the Property is for investment purposes "I")
bondRate	Numeric, discrete, 10 year Treasury Bond Yield Rate		
fedFundsRate	Numeric, discrete, Federal Funds Rate, this is the rate at which lending institutions borrow funds overnight (Cost of funds)		
Borrower Credit Score	Numeric, discrete, Credit Score of the borrower		
Original UPB	Numeric, discrete, Loan quantum		

For Run-2, we used the entire dataset and could successfully train our models to predict all three classes.

Logistic Regression: Using Models trained with this method, we explained the values of coefficients using log odds and odd ratios and identified that "Original Interest Rate" , "bondRate" and fedFundsRate" variables had the most predictive power in identifying the correct class for the outcome variable. The results from the model's performance on the training dataset and the predictions results were compared using Confusion Matrix. When this model was deployed on a test dataset to predict values, it could only predict I and P class correctly. *It was not good at predicting S class. In the second run, we added all the variables from the dataset to find that the model could predict the "S" class also. Also in this run, the summary of the model revealed that various levels of the Seller Name variable had significant coefficients and therefore, this turned out to be another important predictive feature besides the ones found in the previous run. the accuracy of the model is not good for predicting the I and S classes.*

Naive Bayes Classifier: Naive Bayes classifier compiles conditional probabilities for the set of features associated with the observations for each class during the training phase and uses these to predict the class when deployed on the test dataset. We used this classifier with and without Laplace estimator and compared results using Confusion Matrices. We found that this model also was getting better at predicting the

predominant class “P” than other classes. The Laplace estimator did not have a significant effect on the outcome because our dataset was not sparse. We analyzed the reasons for underperformance and found that it could be because of the fact that Naive Bayes model assumes independence of variables. The strong correlation between our variables could be one reason for the underperformance of this model in predicting I and S classes correctly. We performed Run-2 adding all the variables to the model while training. When used for prediction, the model could predict “S” class but not a significant number. Laplace smoothing did not show any significant improvement.

Decision Trees with and without Boosting: We trained Decision Tree models and analyzed the If-then rules to understand how the model functions. We learned how to understand the accuracy of the model and the attribute usage. We compared the model statistics on the training set and then the prediction values when the model was deployed on the test dataset. We visualized the nodes using plots and Confusion Matrices and found that with the set of variables used in Run-1, our model could not predict the “S " class of our outcome variable. In Run-2 we deployed all the variables and found that this model could predict all three classes better than the Logistic and Naive Bayes models.

```
> predictions <- predict(modelDT, newdata = testDT)
>
> CrossTable(testDT$Occupancy.Status, predictions,
+            prop.chisq = FALSE, prop.c = FALSE,
+            prop.r = FALSE,
+            dnn = c('actual Class', 'predicted Class'))
```

```
   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  36032

              | predicted Class
actual Class |         I |         P |         S | Row Total |
-------------|-----------|-----------|-----------|-----------|
           I |       475 |      2141 |        30 |      2646 |
             |     0.013 |     0.059 |     0.001 |           |
-------------|-----------|-----------|-----------|-----------|
           P |       240 |     32209 |        16 |     32465 |
             |     0.007 |     0.894 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
           S |        90 |       817 |        14 |       921 |
             |     0.002 |     0.023 |     0.000 |           |
-------------|-----------|-----------|-----------|-----------|
Column Total |       805 |     35167 |        60 |     36032 |
-------------|-----------|-----------|-----------|-----------|
```

Conclusion: Among all classifiers, the one with the most predictive quality is the Decision Tree with Boosting. In predicting a class of outcome therefore, our best option is to use this model.

References:

1. Code and insights were used from class Youtube videos for this analysis
2. Multinomial Logistic Regression | R Data Analysis Examples (ucla.edu)
3. User's guide to correlation coefficients - PMC (nih.gov)
4. *https://www.macrotrends.net/2016/10-year-treasury-bond-rate-yield-chart*
5. *https://www.usinflationcalculator.com/inflation/historical-inflation-rates/*
6. https://www.macrotrends.net/2604/30-year-fixed-mortgage-rate-chart
7. https://data.bls.gov/timeseries/LNS14000000
8. https://www.macrotrends.net/2015/fed-funds-rate-historical-chart

Contributions:

All the 3 team members made equal contributions for research, analysis, code creation, PPT and Video presentation for Milestone-2