Group 10

Jun, Wu

Vishwa, Shah

Avinita, Shetty

MidTerm Project

Advances in Data Science/Architecture

Table of Contents

1. **West Roxbury**

1.1 Exploratory Data Analysis2

1.2 Building Predictive Model15

1.3 Building Model and Evaluation18

1.4 Model Recommendation29

**2. Mortgage Defaults**

2.1 Exploratory Data Analysis30

2.2 Building Classification Model33

2.3 Performance Evaluation34

2.4 Model Selection and Recommendation39

**3. Spam base**

3.1 Exploratory Data Analysis40

3.2 Building Classification Model42

3.3 Model Evaluation43

3.4 Model Comparison50

**4. Blog Feedback**

4.1 Building Predictive Model51

4.2 Performance Evaluation52

4.3 Model Selection and Recommendation53

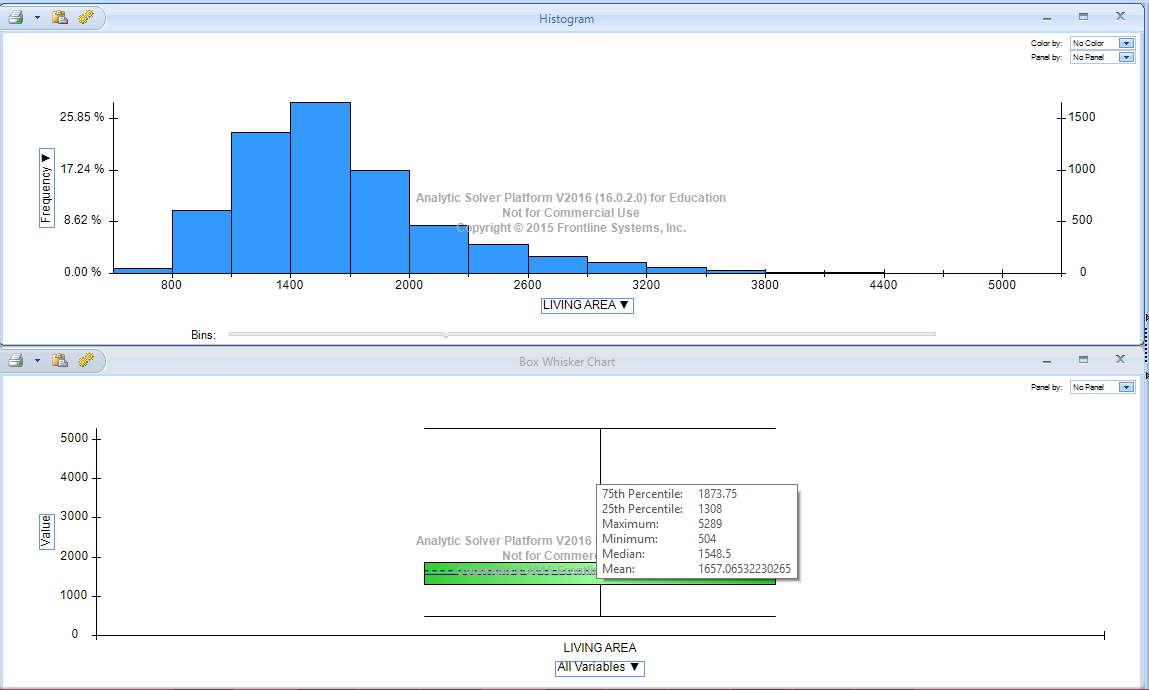
1. **West Roxbury**

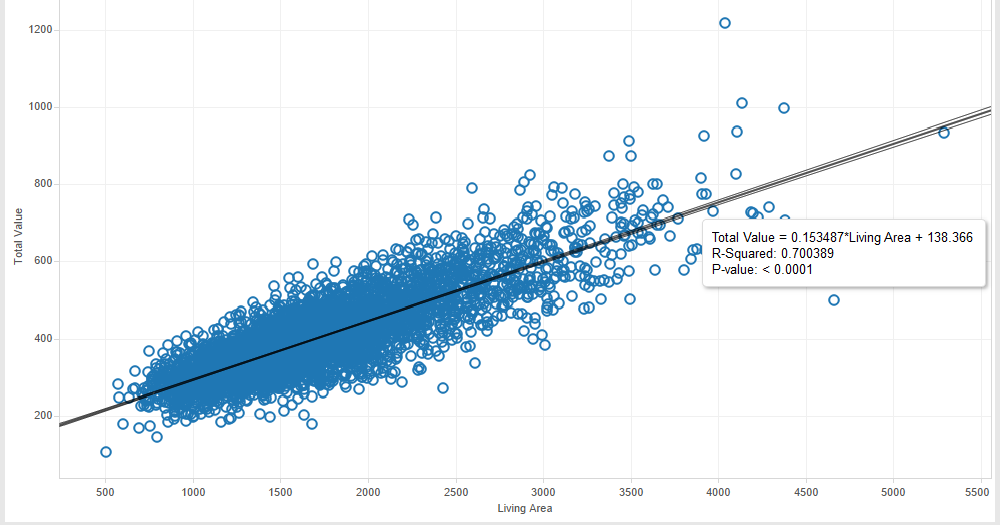
Problem: West Roxbury is a dataset about the houses in the West Roxbury neighborhood and we need to compute the expected prices of a home in future from this given dataset.

* 1. **Exploratory data analysis using TABLEAU and XLMiner**

Tool Used: Tableau and Excel

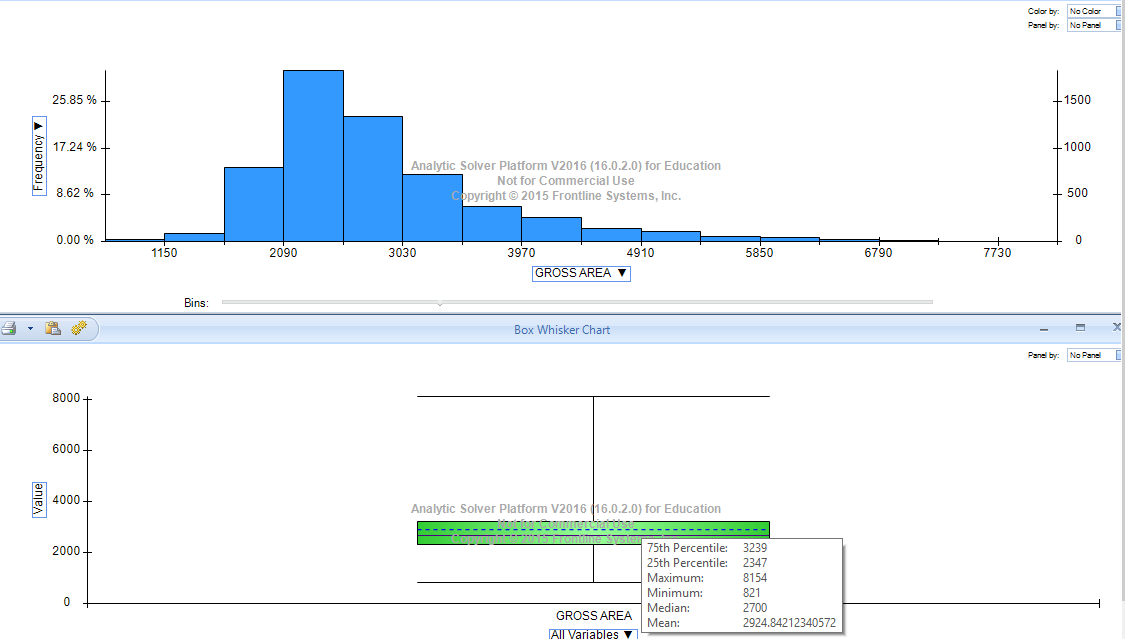
Living Area:



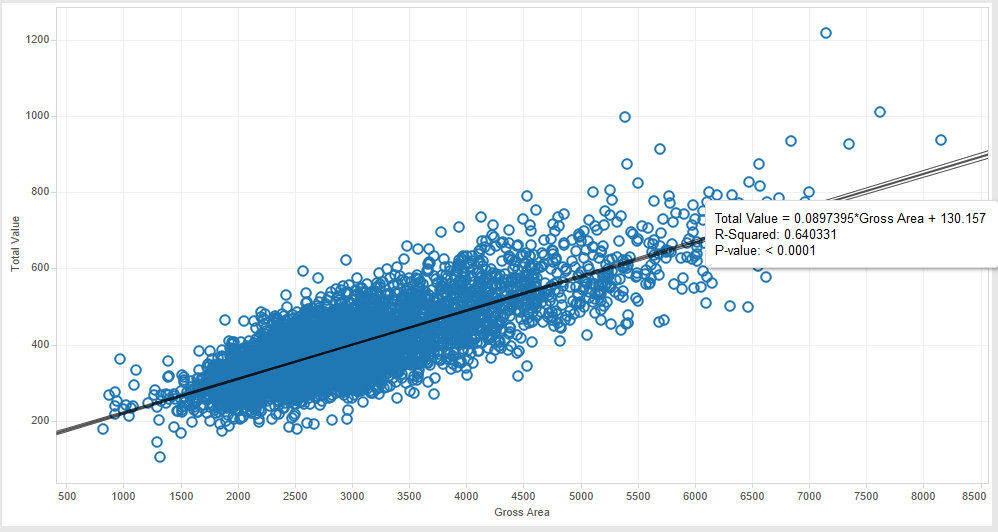
Scatter Plot in Tableau for Living Area

This Chart shows how the total price of the house depends on the Living area. The more the Living area of the house the price is higher.

Gross Area:

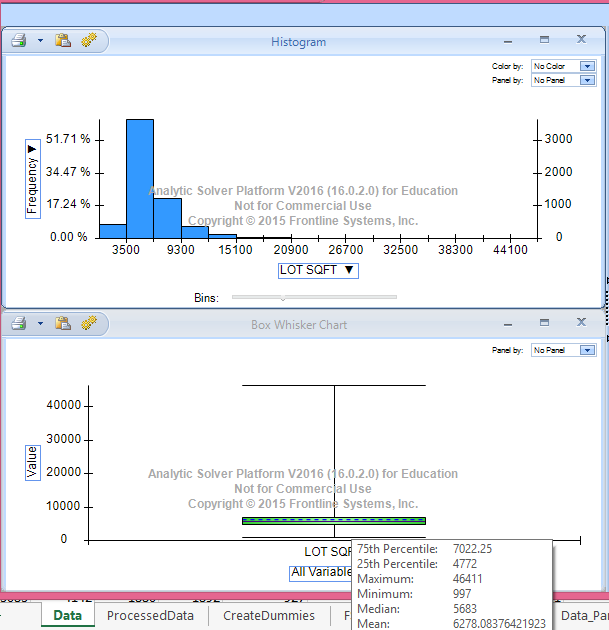


Scatter Plot in Tableau for Gross Area:

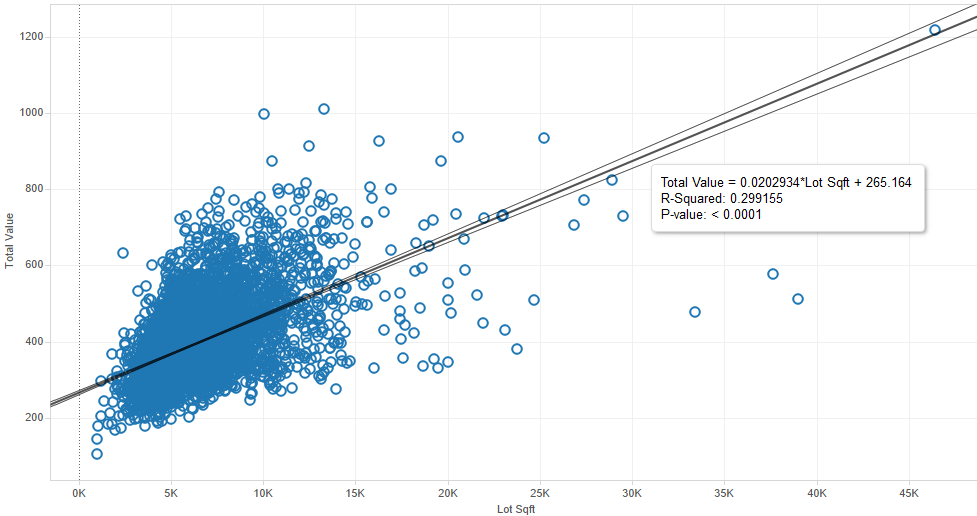


This Chart shows how the total price of the house depends on the Gross area. The more the gross area of the house the price is higher.

Lot Sqft:



Scatter Plot in Tableau for Lot Sqft

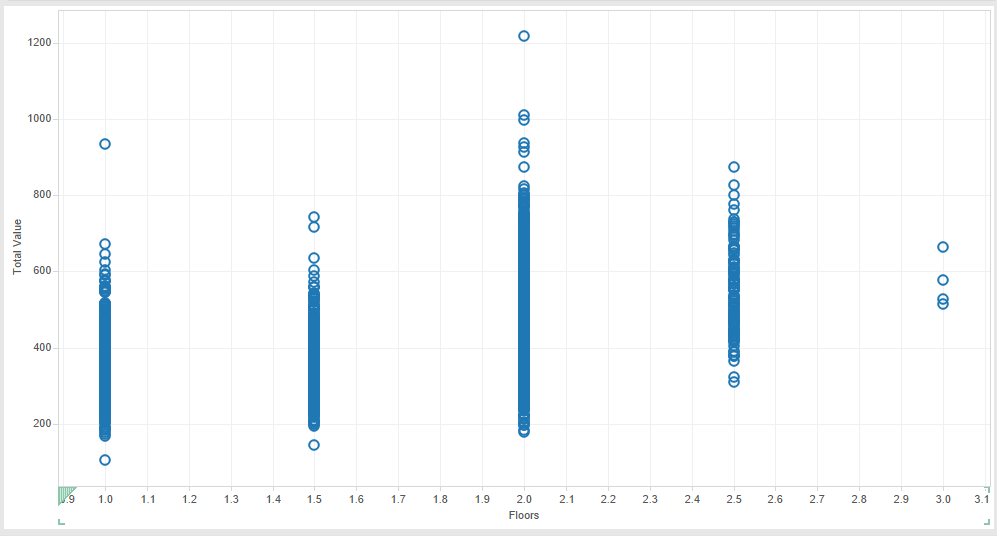


This chart shows how the Lot Sqft effect the pricing of the house. There are a lot of houses of 1000-15K sqft and they are priced between 100$ to 800$. As the size of the Lot increases the price increases too.

Floors:

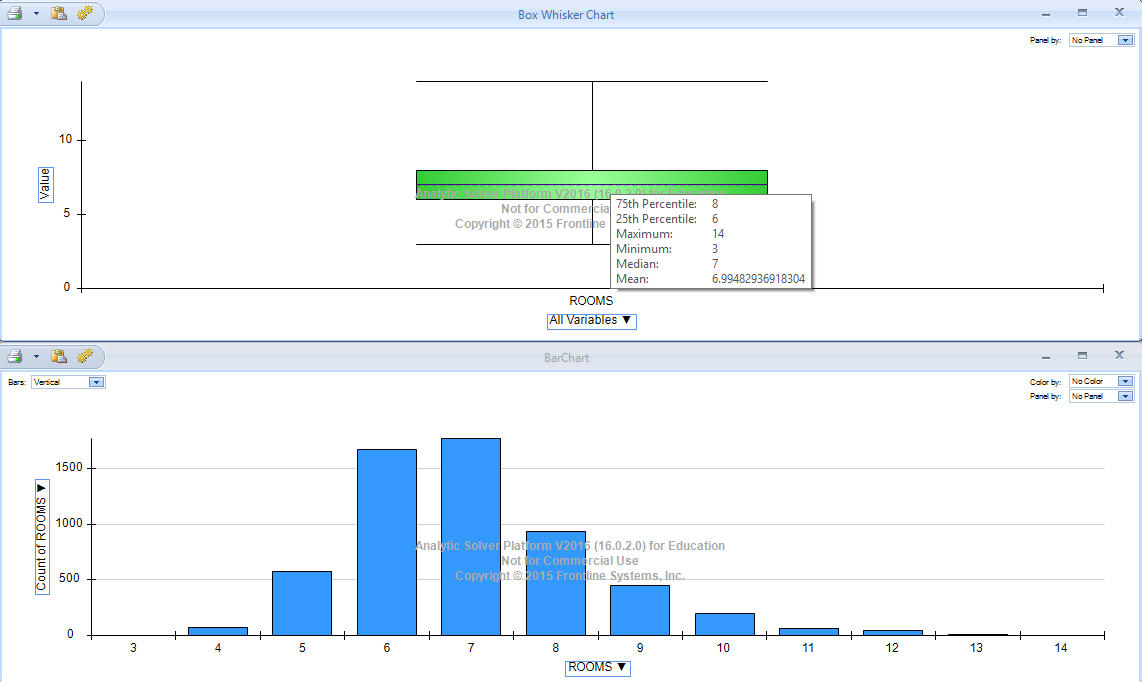


Scatter Plot on Tableau for Floor:

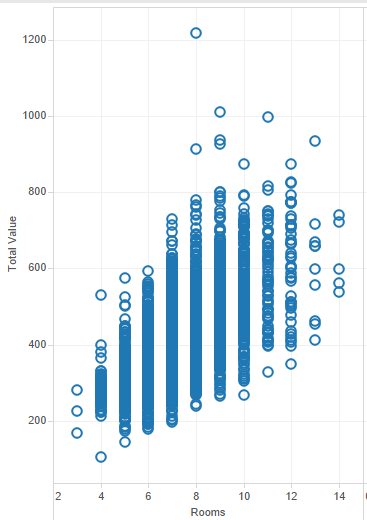


A very few houses in this dataset have 3 floors less than 1%. So this is an outlier.

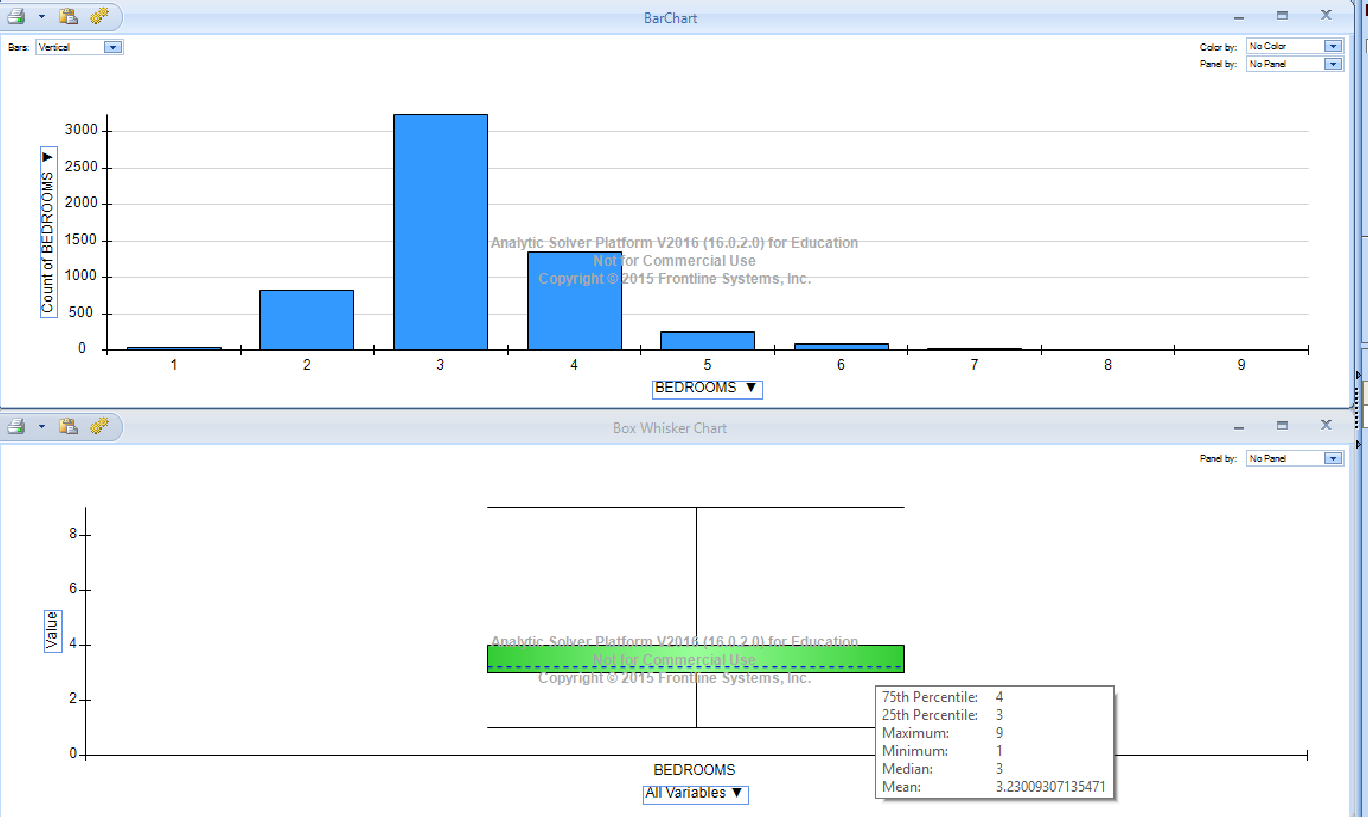
Rooms:



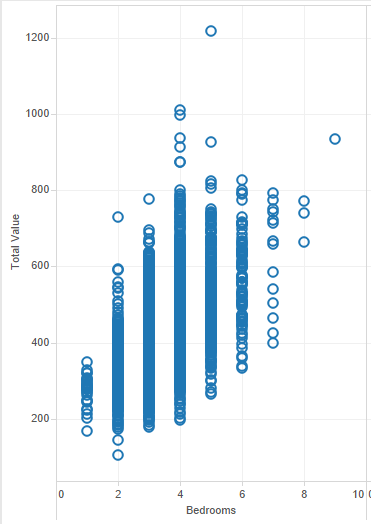
Scatter Plot on tableau for Room:



We can observe here that very few houses have 14 rooms. Most of the houses in this neighborhood has around 3-8 rooms in total.

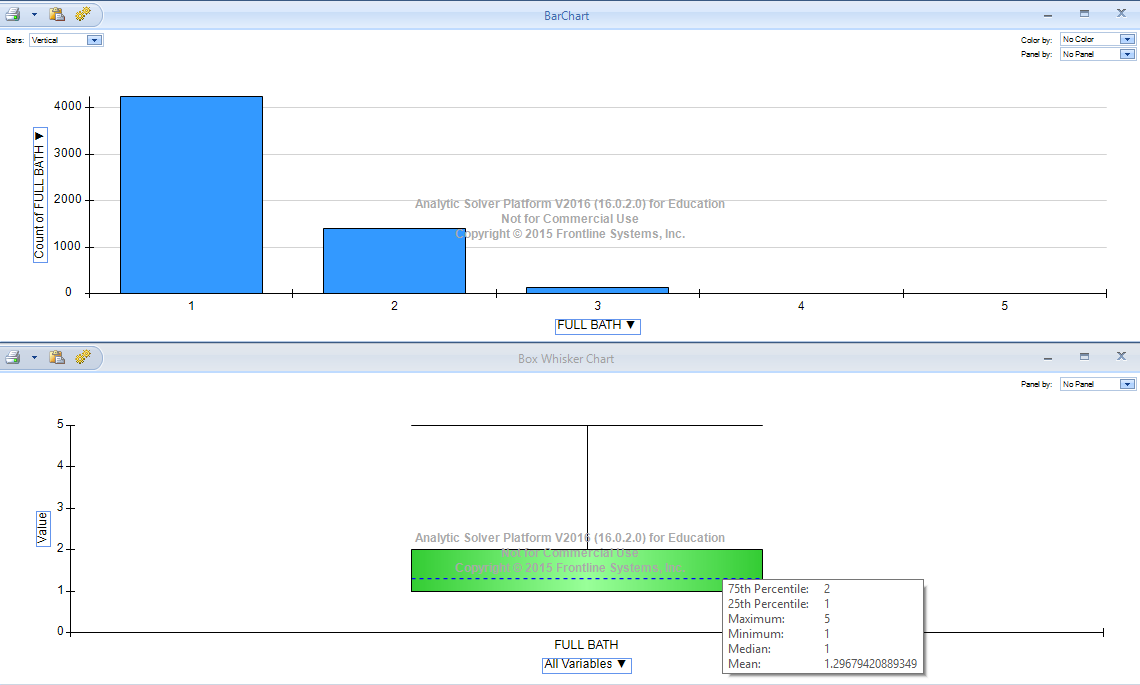
Bedrooms: 

Scatter Plot in Tableau for Bedrooms:

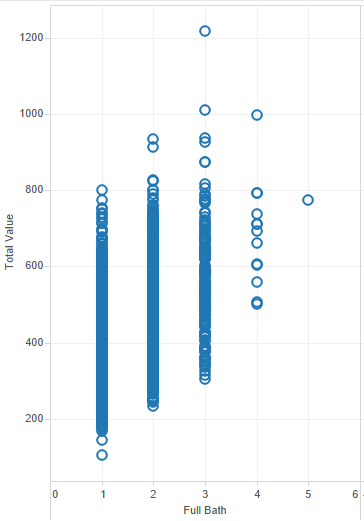


There is just one house with 9 bedrooms in this dataset. Also houses with 8 bedrooms are very less relatively.

Full Bath

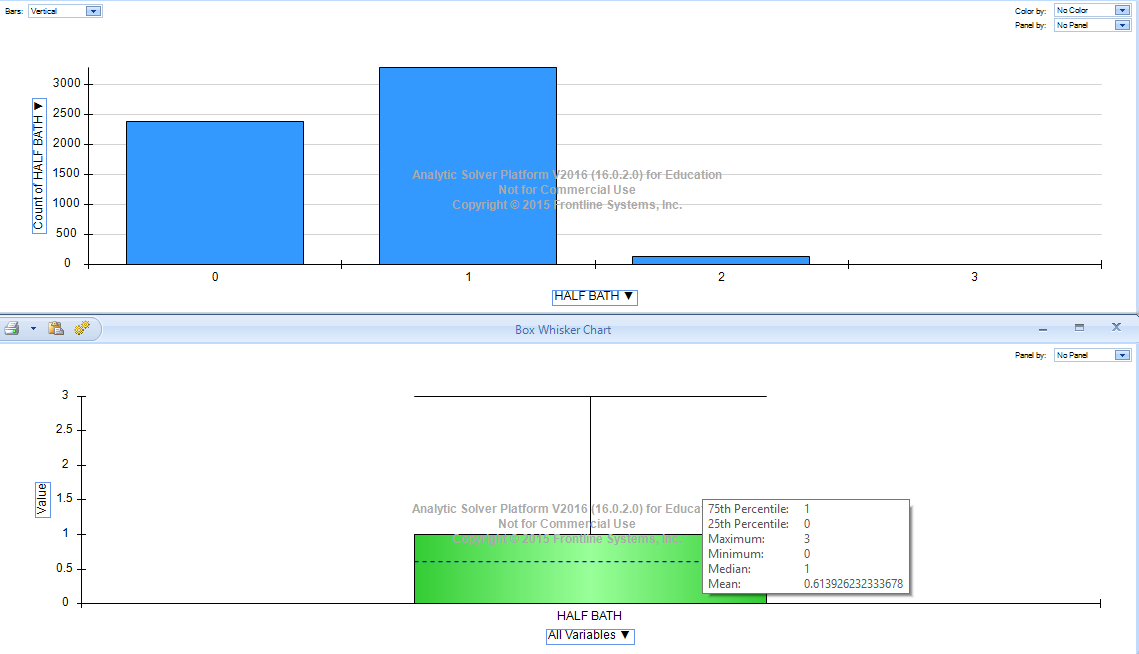


Scatter Plot in Tableau for Full Bath:

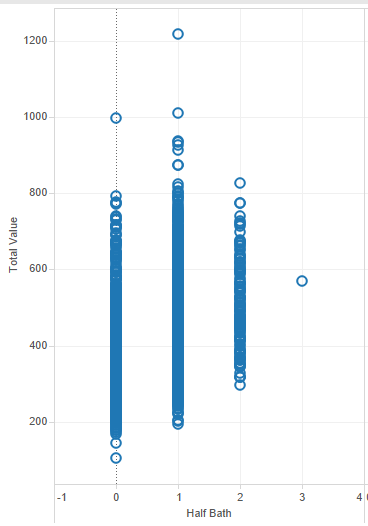


Here we can see very few houses have 4 Full Bath rooms and only 1 of them has 5 full baths. Also there are many houses with 1, 2 and 3 full baths. Houses with 3 full baths are priced a little higher.

Half bath

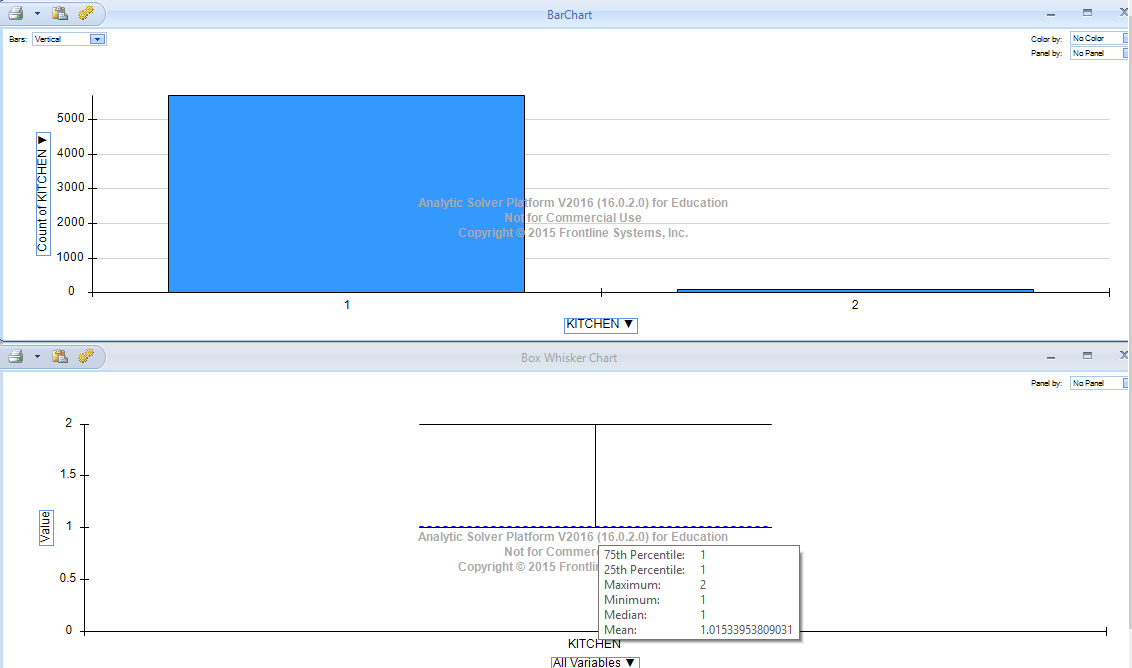


Scatter Plot in Tableau for Half bath

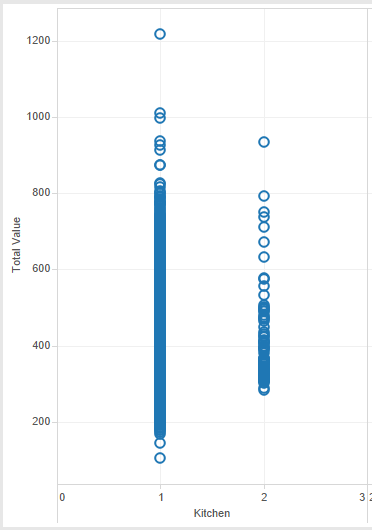


From looking at this we can find that there is only 1 data point which has 3 half baths, this may be an outlier.

Kitchen:

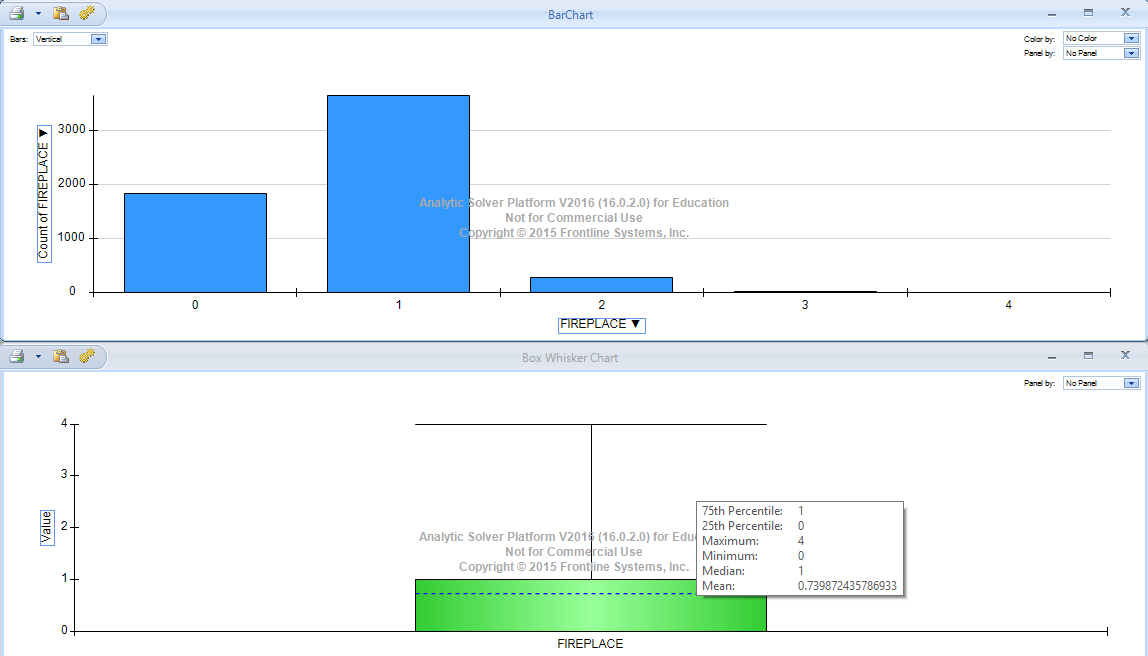


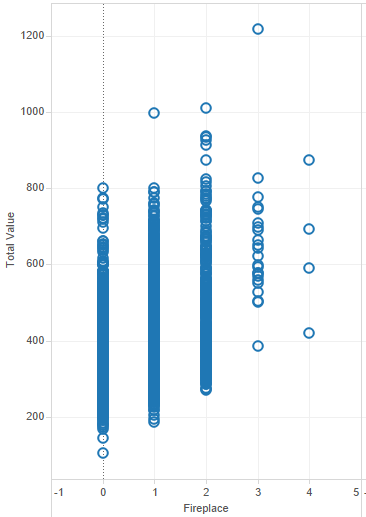
Scatter Plot in Tableau for Kitchen



Here there are very few houses with 2 kitchens which can be seen using the Barchart. Only 90 observations have 2 kitchen out of 5890 total observation.

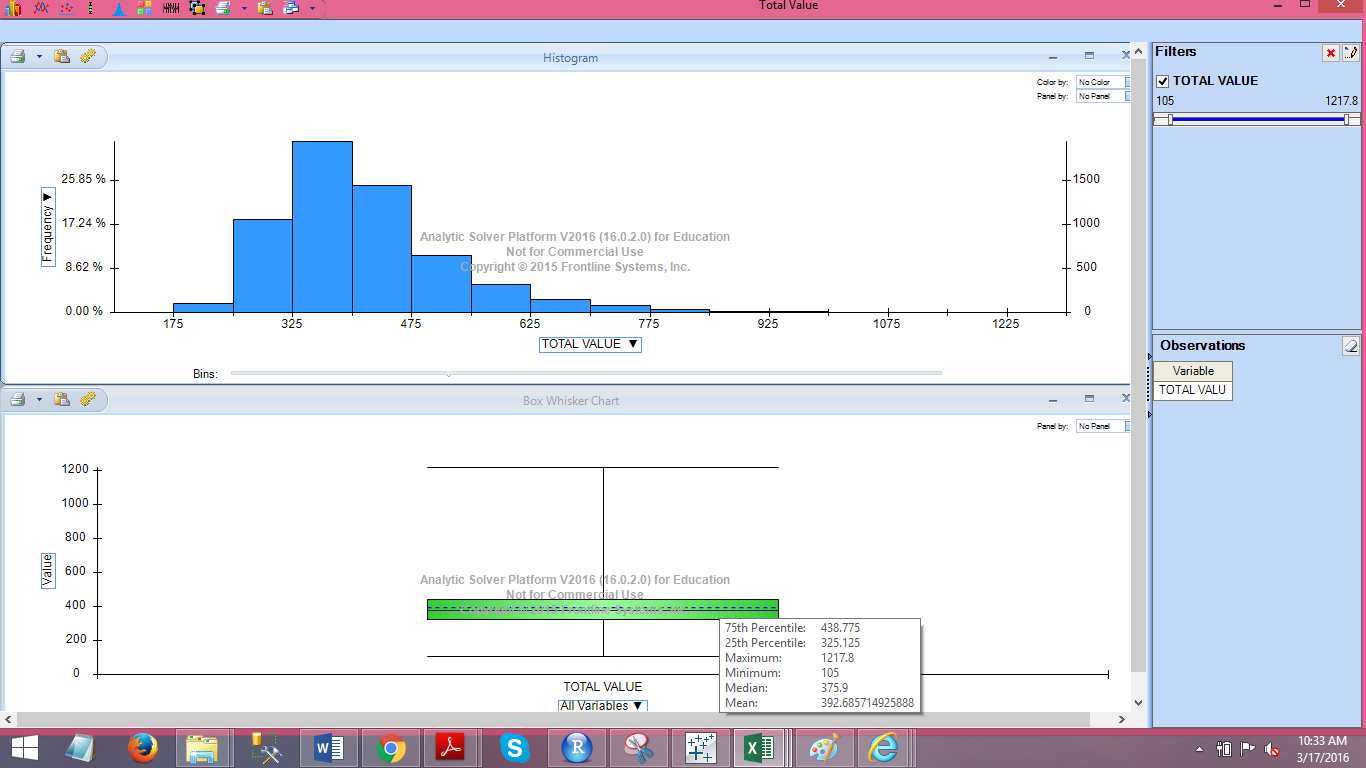
Fireplace:

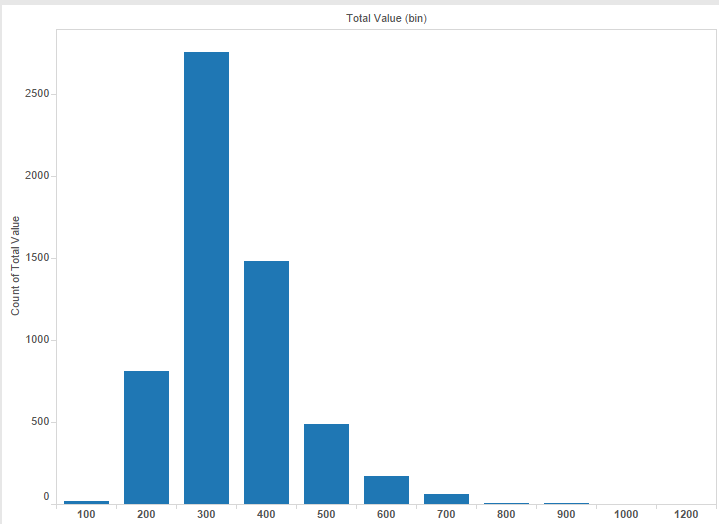




Here we can see there are very few (4) data points at fireplace=4 which means these are outliers.

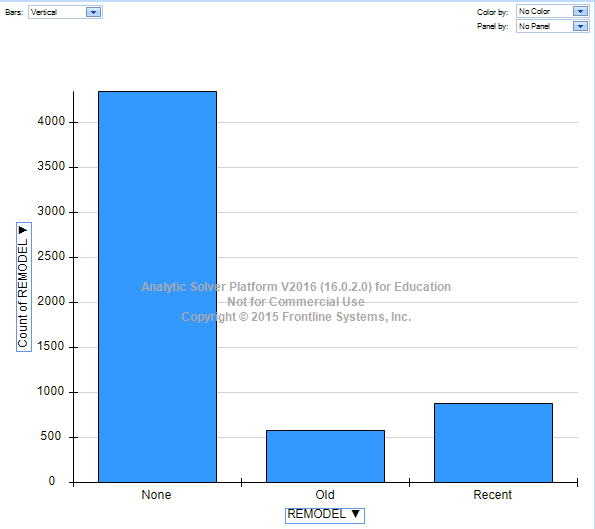
Summary of Total Value



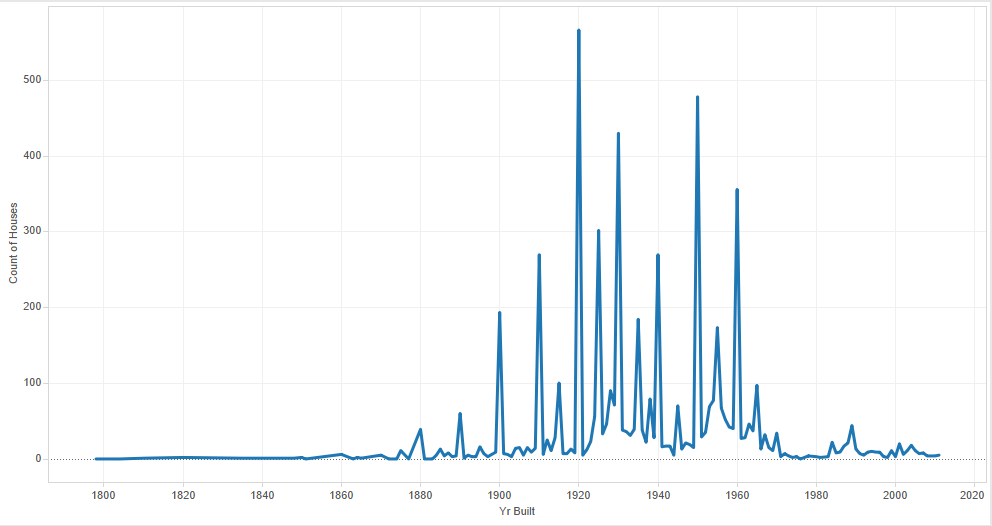


Most of the houses range from 200 to 600$ there are very few houses priced between 700-1000$ and only 1 house priced around 1200$.

Number of remodeled houses

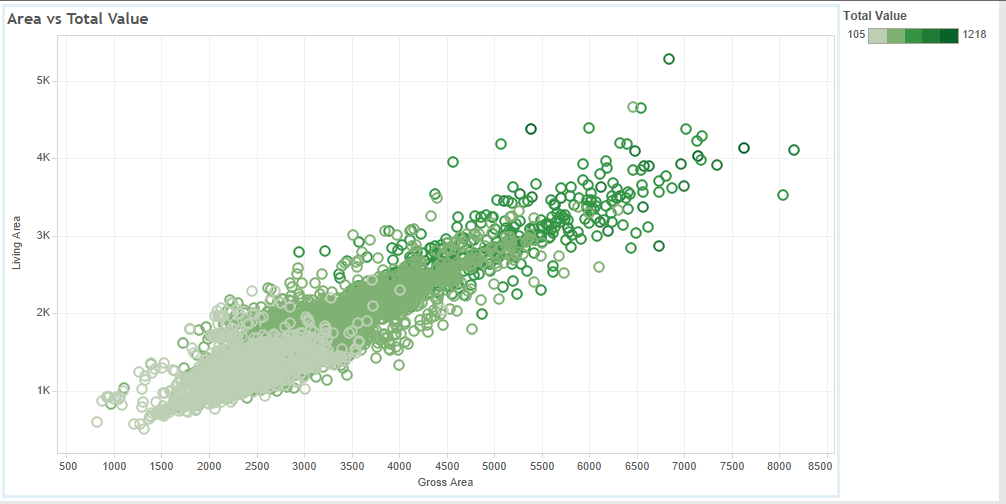


Trend of Year in which houses were built



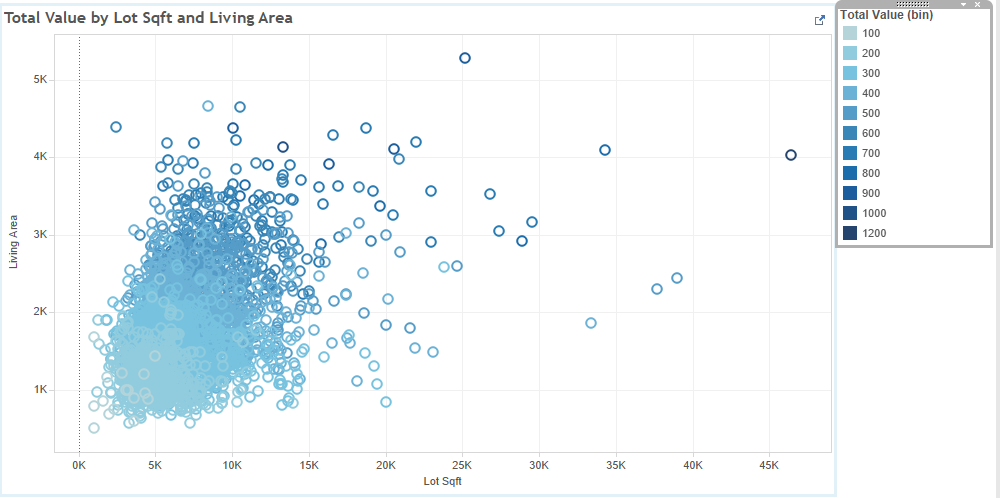
Below chart shows trend when most no. of houses were build. We can see here that there was increase in the number of houses build starting from 1880 and it increased a lot in and around the mid 90’s and it went down again the early 2000’s

Effect of Area on Total Value:



The chart shows when the Living Area and Gross area increases the total value increases too. The gradation in color shows the range of Total Value of the houses.

Effect of Lot Sqft and Living Area on Total Price



More Charts in the Problem1.twb show detailed Exploration of Data.

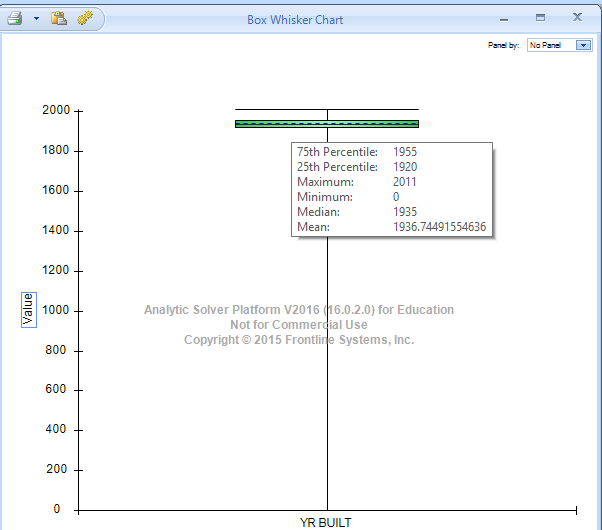
* 1. **Building Prediction model using Multiple Linear Regression, CART and Random Forest**

Tool: XLMiner

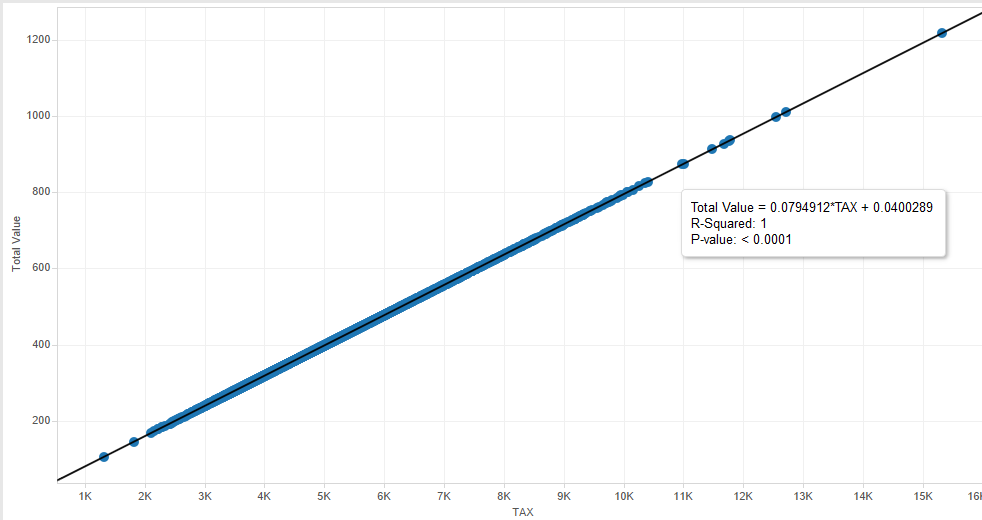
Steps:

1. **Data Pre-Processing:**

After carefully examining the dataset we find a row where the Year built is 0. Since we cannot randomly select a year and replace 0 with it, since many other predictors may be related to year. Also there is only one such row with year 0 so we can simply ignore this row and delete it.



Also the column Tax is calculated from the Total Value of the house. Tax is 12.5 times the Total Value of the house. Hence it is not an independent variable, it is directly dependent on Total Value. So we delete this column as we do not need it for the prediction.



The above chart shows the relation between Tax and Total amount.

1. **Creating Dummies for Categorical Variables:**

The Predictor Remodel (categorical feature) has 3 category which are NONE, OLD and RECENT. We create Dummies for this predictor since otherwise we cannot use them directly for Regression.

Similarly there are other variables like

ROOM (14 Categories)

BEDROOM (9 Categories)

KITCHEN (2 Categories)

FLOORS (5 Categories)

FULL BATH (5 Categories)

HALF BATH (4 Categories)

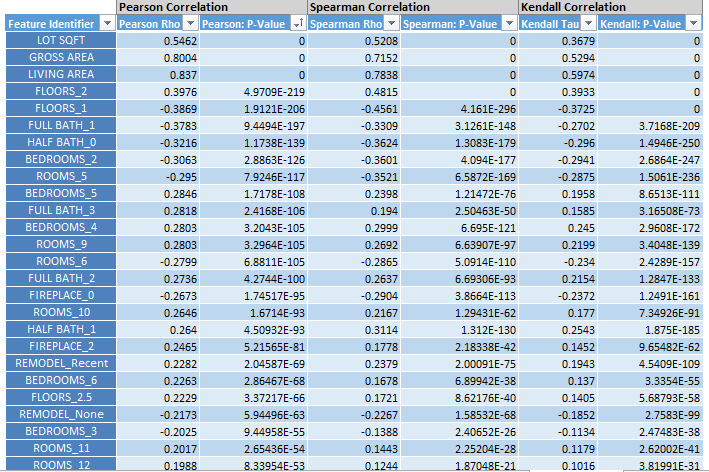
FIREPLACE (5 Categories)

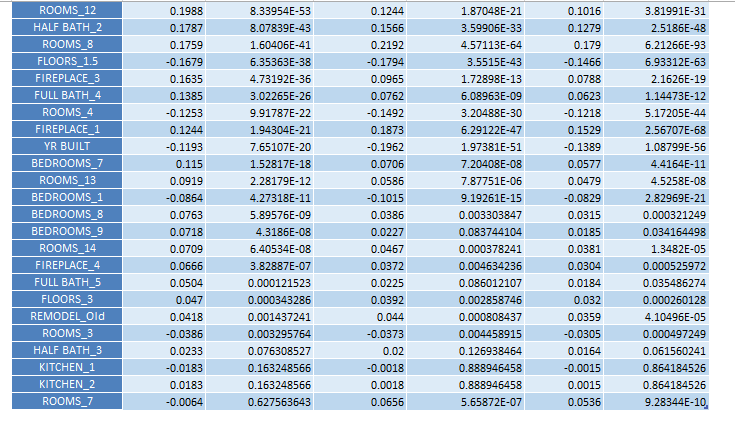
Now we will have 49 variable in total now of which 8 of them would not be useful since if there are n dummy variables n-1 are only useful. So from these 41 variables we need to determine which of them matter how much.

We can determine this by Feature Selection feature of XLMiner. By using Feature selection we can see the Co-relation and the P value of the independent variables on Total Value. This Feature Selection uses Pearson, Spearman and Kendall Correlation for determining the Correlation between the variables. We will consider the values of Pearson Correlation for this problem. Higher Correlation and lower P values are considered to build a good model.

* 1. **Building Model and its evaluation**

1. **Feature Selection**

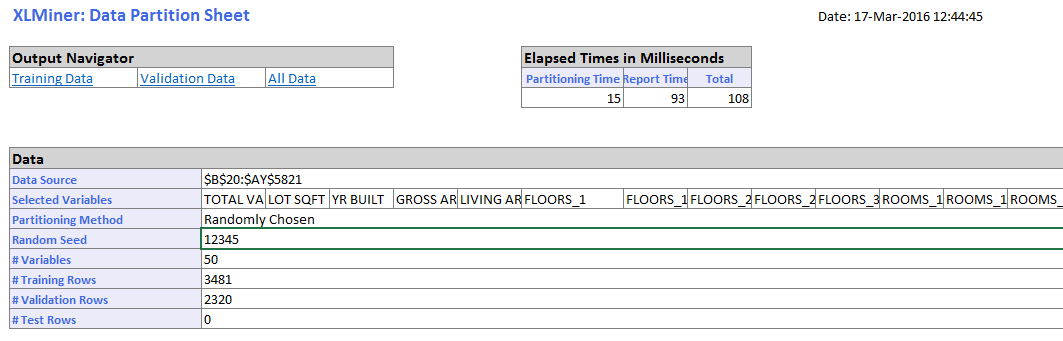




1. **Partition the Data**

Next Step would be to partition the data

We create a random standard partition of 60% Training data and 40% for Validation



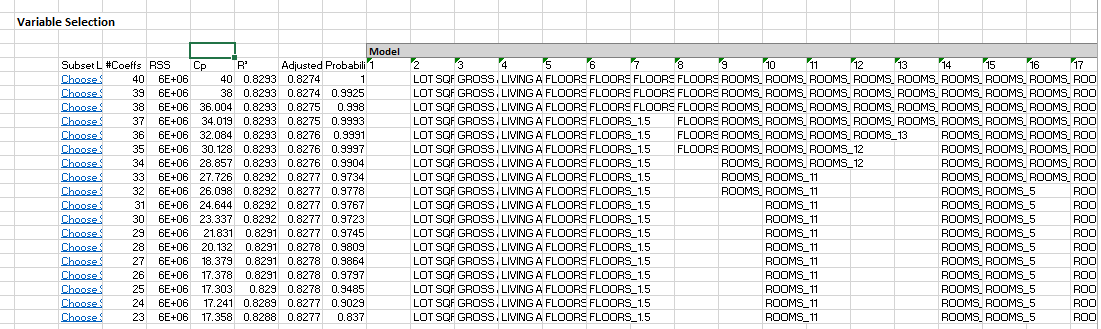
1. **Build Models**
2. We build a model using Multiple Linear Regression, CART and Random Forest. For the Multiple Linear Regression we use the process of Backward Variable Selection to find out the best subsets of variables which would give us a good model.

The goal of the model is to have the RMSE value as low as possible. The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a frequently used measure of the differences between values (sample and population values) predicted by a model or an estimator and the values actually observed.

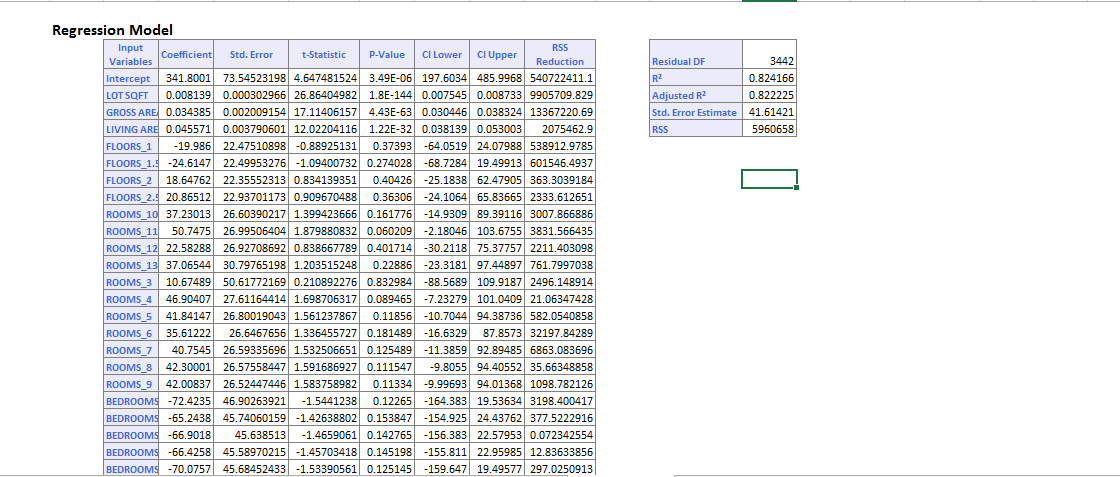
So a good model would be the one with low RMSE value and high R-squared and adjusted R squared value. Also for a good model the RMSE of Training and Validation should be close to each other.

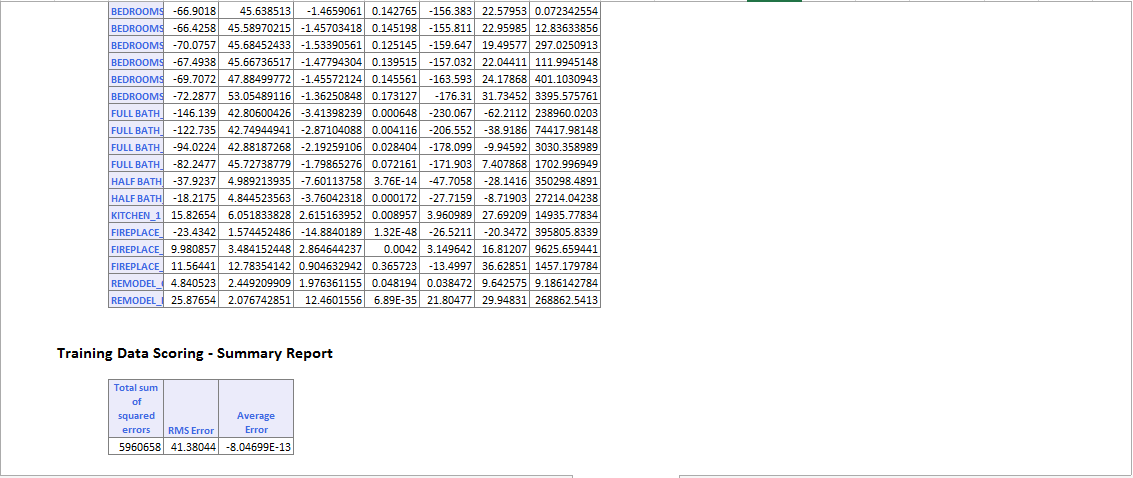
1. The most important factor in predicting the total value in any prediction problem is to select the variables correctly and find the best regression equation for prediction using these features.
2. **Regression Model:**

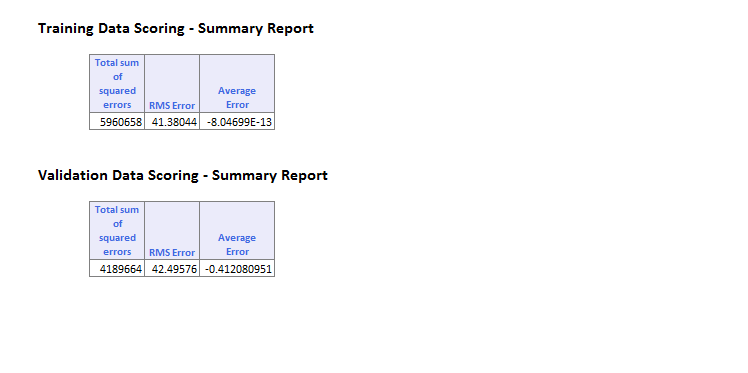
Using feature selection we can select the top variables and perform the regression. But how would we know how many variables will create best suitable model. For this we can use Variable Selection. We have used Backward Variable Selection as there are many variables to select from.



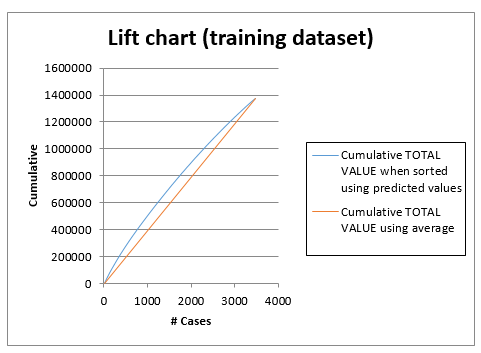
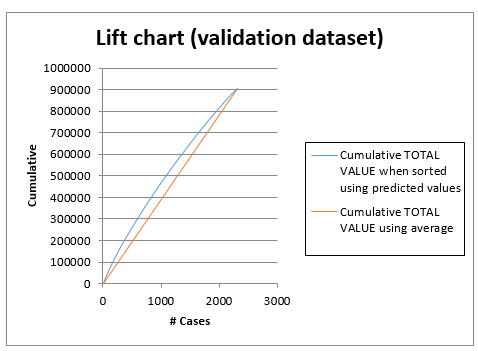
We select the Subset with 38 variables because the Cp value is close to no. of variables and also the R square and adjusted R squared values are pretty good and close to each other.



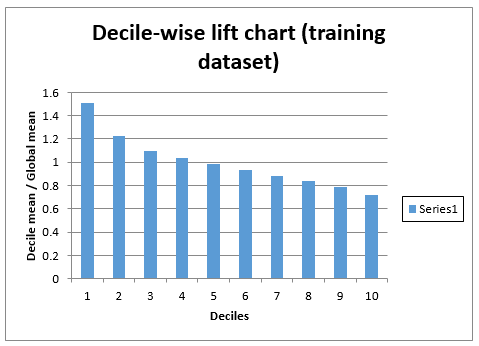
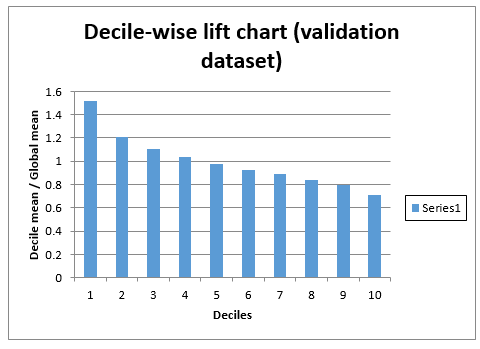


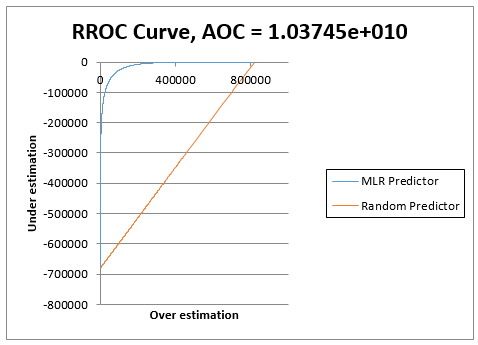
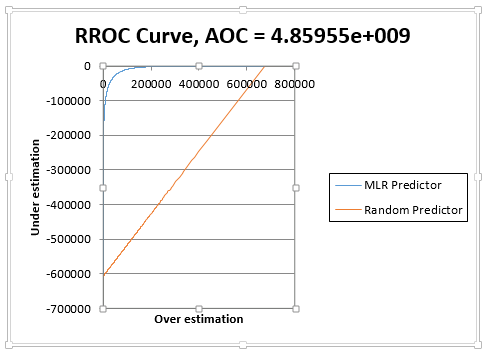


The RMSE we obtain for the Training is 41.38 and Validation dataset is 42.49 which are very much similar and also the average error is low. Also if we check the R-squared and adjusted R squared they are as high as 0.822 and are close to each other indicating this is a good model.

**Lift Charts for Regression:**

**Decile wise lift chart for Regression**



**RROC Curve**

The above charts show the Lift Chart, Decile Wise Lift Chart and the RROC charts for training and validation dataset of Multiple Linear Regression.

**Lift Chart**

Lift is a measure of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model.

The red line in the lift chart shows the baseline, which indicates the measure of effectiveness of any random model. The blue line shows the measure of effectiveness of our model which is above the baseline indicating our model is better than the baseline.

**Decile Chart**

After building a statistical model, a decile analysis is created to test the model’s ability to predict the intended outcome. Each column in the decile analysis chart represents a collection of records that have been scored using the model. The height of each column represents the average of those records’ actual behavior.

**Ideal Situation: The Staircase Effect**  
When you’re looking at a decile analysis, you want to see a staircase effect that is, you’ll want the bars to descend in order from left to right.

**Not-So-Ideal Situations**

In contrast, if the bars seem to be out of order, the decile analysis is telling you that the model is not doing a very good job of predicting actual responses.

If the bars seem to be the same height, or the decile analysis looks “flat”, the decile analysis is telling you that the model isn’t performing any better than randomly binning people into deciles would. In both cases, your model should be improved before moving forward with it.

Our linear regression model follows the staircase effect to some extent which means it is a good model.

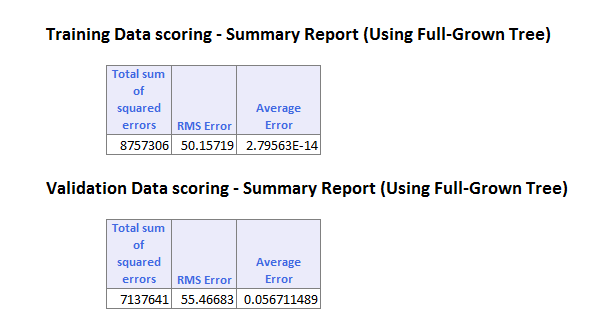
**RROC Curve**

RROC curves plot the performance of regressors by graphing over estimations (or predicted values that are too high) versus under estimations (or predicted values that are too low.) The closer the curve is to the top left corner of the graph (the smaller the area above the curve), the better the performance of the model. Area Over the Curve (AOC) is the space in the graph that appears above the ROC curve and is calculated using the formula: sigma2 \* n2/2 where n is the number of records. The smaller the AOC, the better the performance of the model.

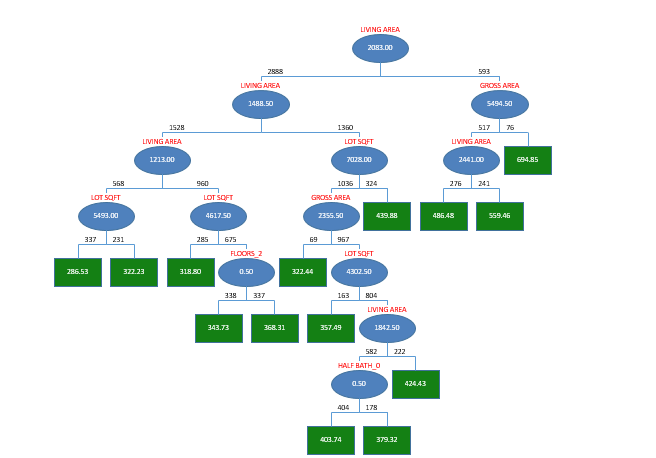
From above charts we can see that the AOC is very less which means that the model is a good model.

1. **CART Model**

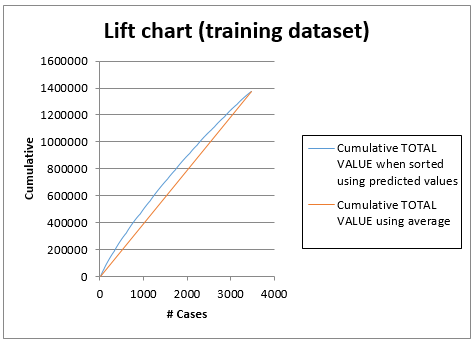
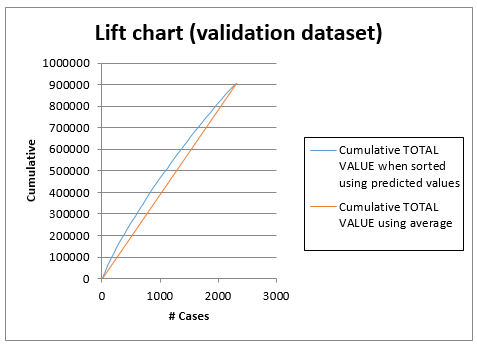
Results for CART are as follows:



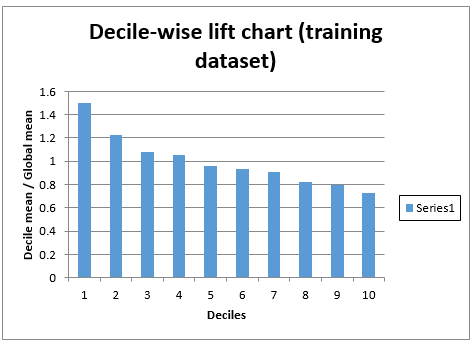
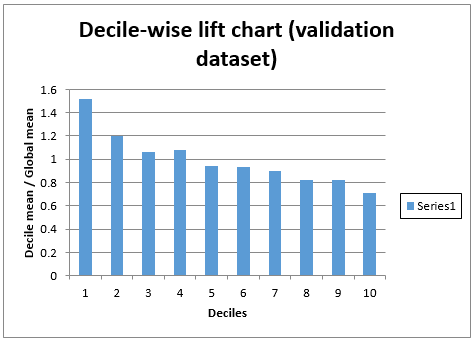
Here the RMSE is 50.15 for training dataset and 55.46 for Validation dataset. Also, only 5 variables are used by the Regression tree which can be seen below. The difference between RMSE for training and validation is greater in this model than in Multiple Linear Regression.

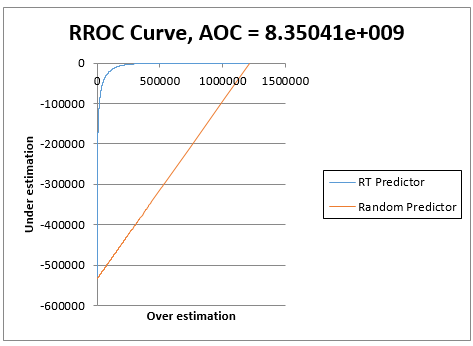
Full Grown Regression Tree

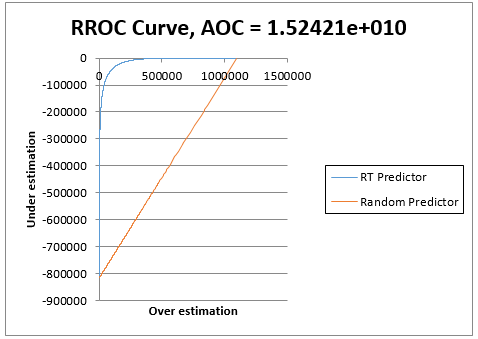
**Lift Charts for CART**

****

**Decile wise Lift Chart for CART**

****

**RROC Curve**



**Lift Chart**

The lift chart for Regression Tree is similar to the one of Multi Linear Regression so it is better than the baseline model. We cannot compare the Linear Regression with CART based on this lift charts as it is not giving us any significant difference.

**Decile Chart**

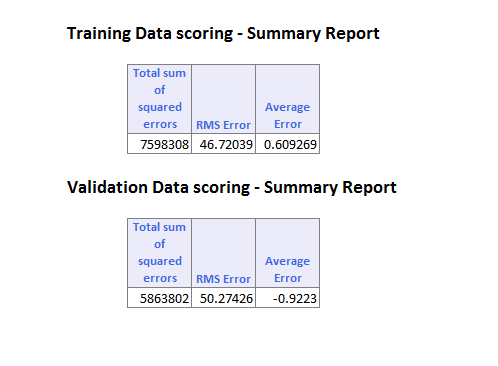
The decile chart for CART is little unstable which shows that the model is not very good.

**RROC Curve**

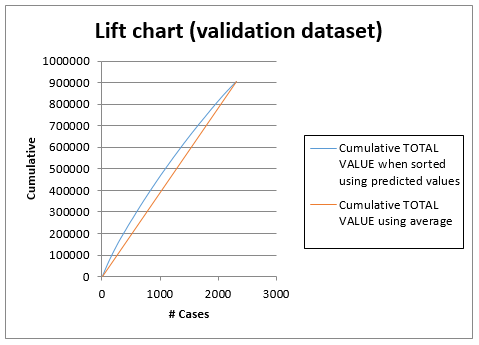
The value of the AOC is less in this model, but the AOC for multiple linear regression is the least amongst the 3 models.

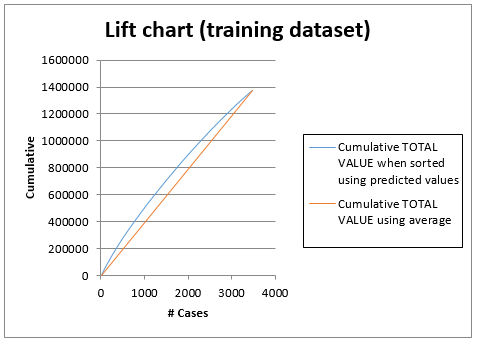
1. **Random Forest**

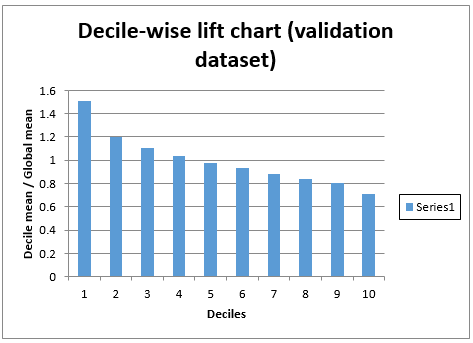
Result for Random Forest are as follow:

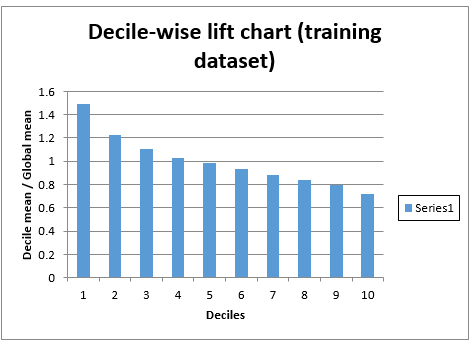


Here the RMSE for Training and Validation is better than CART since in CART only single tree is used and Random Forest creates many such trees and gives best output. The RMSE for Training and validation still differ more than that in Linear Regression. Also the RMSE for Multiple Linear Regression was lower than that in Random Forest.

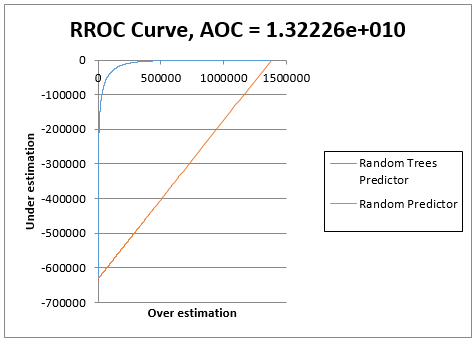
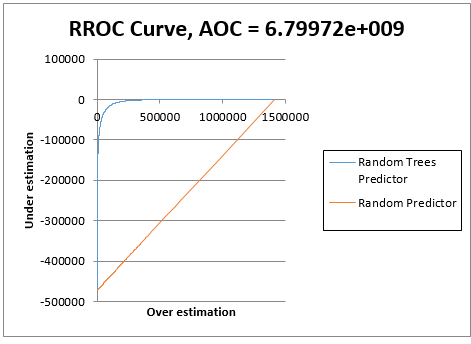
Lift Charts for Random Forest:



Decile Wise Lift Charts:



RROC Curve



**Lift Chart**

The lift chart for Random Forest is similar to the one of Multi Linear Regression so it is better than the baseline model. We cannot compare the Linear Regression with CART based on this lift charts as it is not giving us any significant difference.

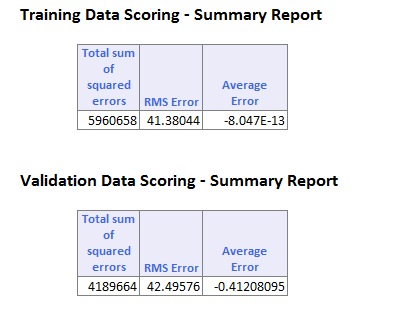
**Decile Chart**

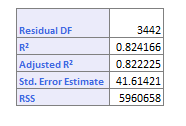
The decile chart for Random Forest is better in this case as compared to CART.

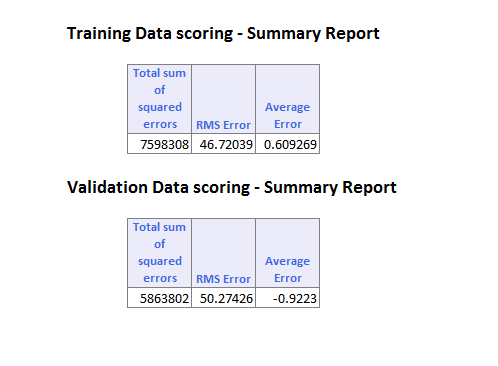
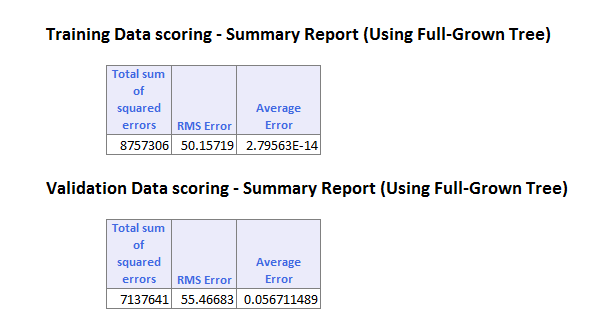
**RROC Curve**

The value of the AOC is less in this model. But the AOC for multiple linear regression is the least amongst the 3 models.

* 1. **Model Recommendation**

**Multiple Linear Regression:**



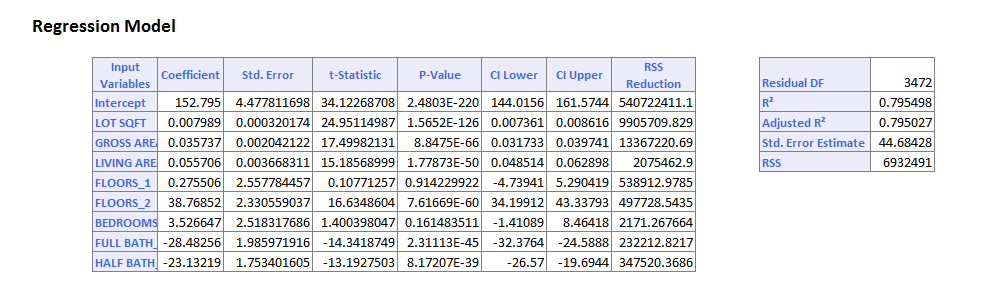
**Random Forest: CART Results:**

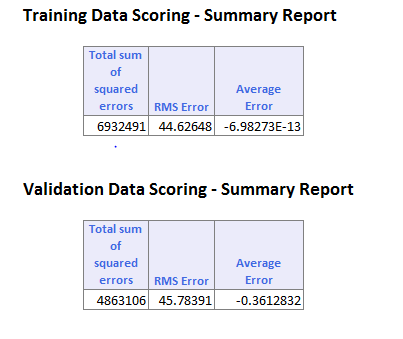
Comparing these three we can see the best (lowest) RMSE value of 41.38 for training and 42 for validation is derived by multiple linear Regression and the R-square as well as Adjusted R-squared is also around 0.82.

Also the Lift charts the Decile wise lift chart for the Multiple Linear regression are better in the case of Multiple Linear Regression.

It can be argued that since we have included a large no. of features for Multiple Linear Regression we are getting RMSE smaller than the other two model. But this is not the case. We have selected the variables by performing variable selection and feature selection. Also if we reduce the number of features that we input into the Multiple Linear Regression model by selecting only top 5-6 features even then the RMSE would be lesser than what we get by Random Forest and CART.

Proof that Multi Linear Model works better for this problem in comparison to CART and Random Forest even with lower number of features:





So the model derived by using Multi Linear regression would be the best model for this case of determining the prices of the houses in West Roxbury. The second best based on performance is Random Forest and CART is the last based on performance that we would recommend.

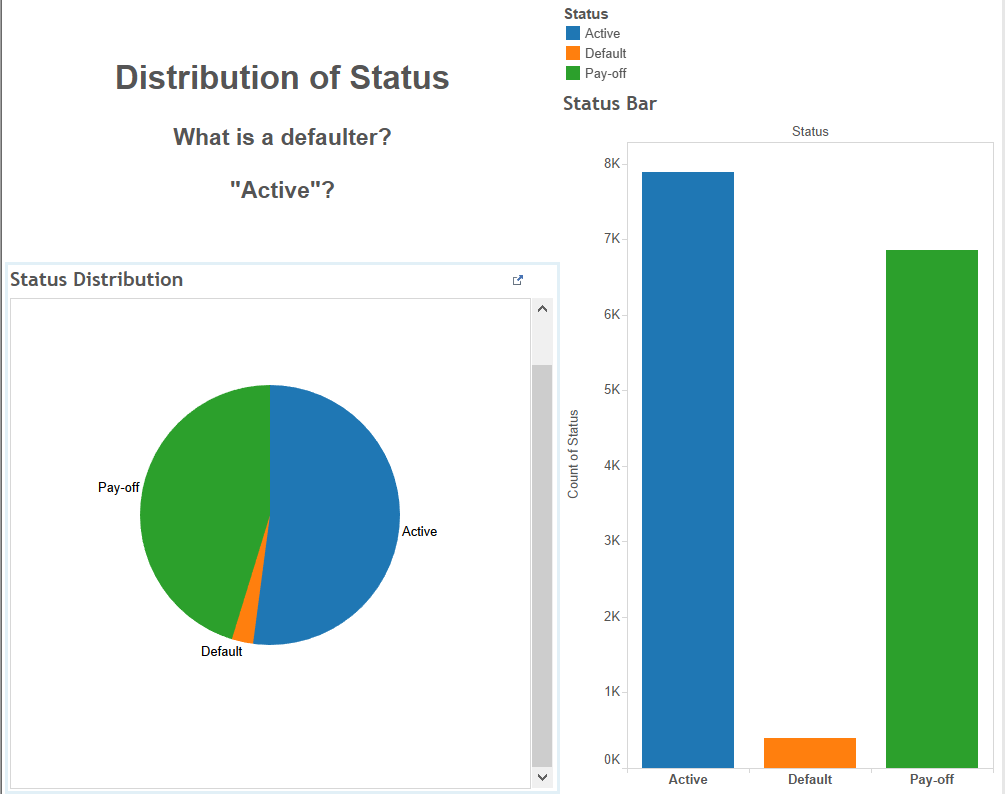
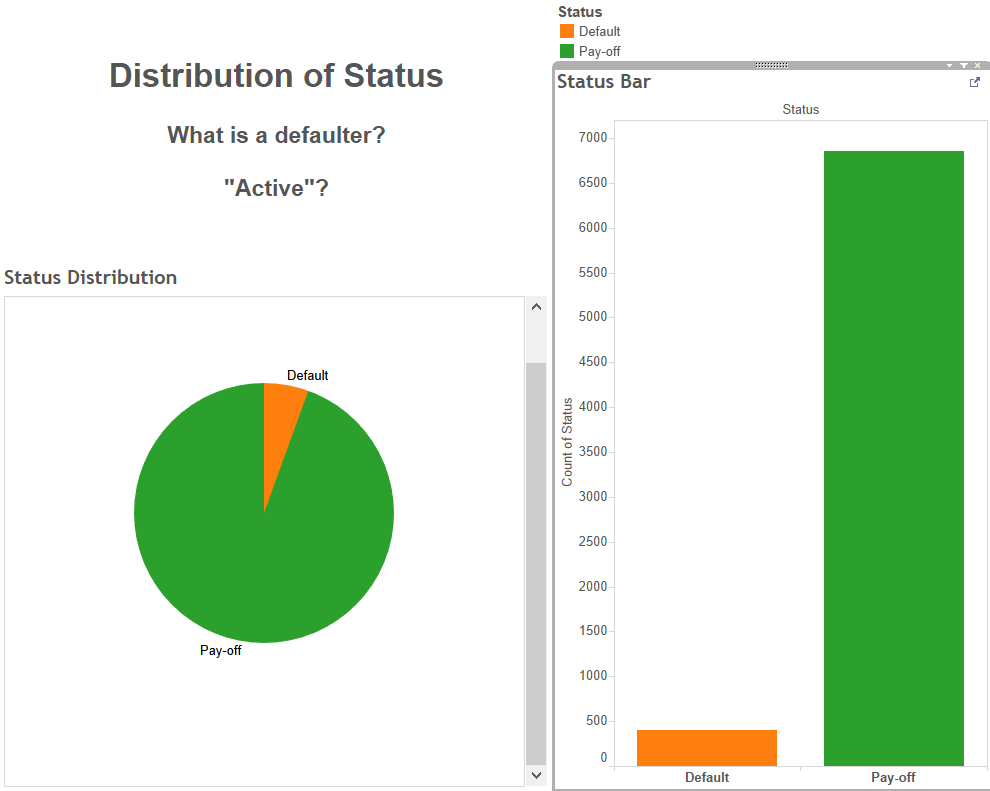
**2 Mortgage Defaults**

**2.1** **Exploratory data analysis**

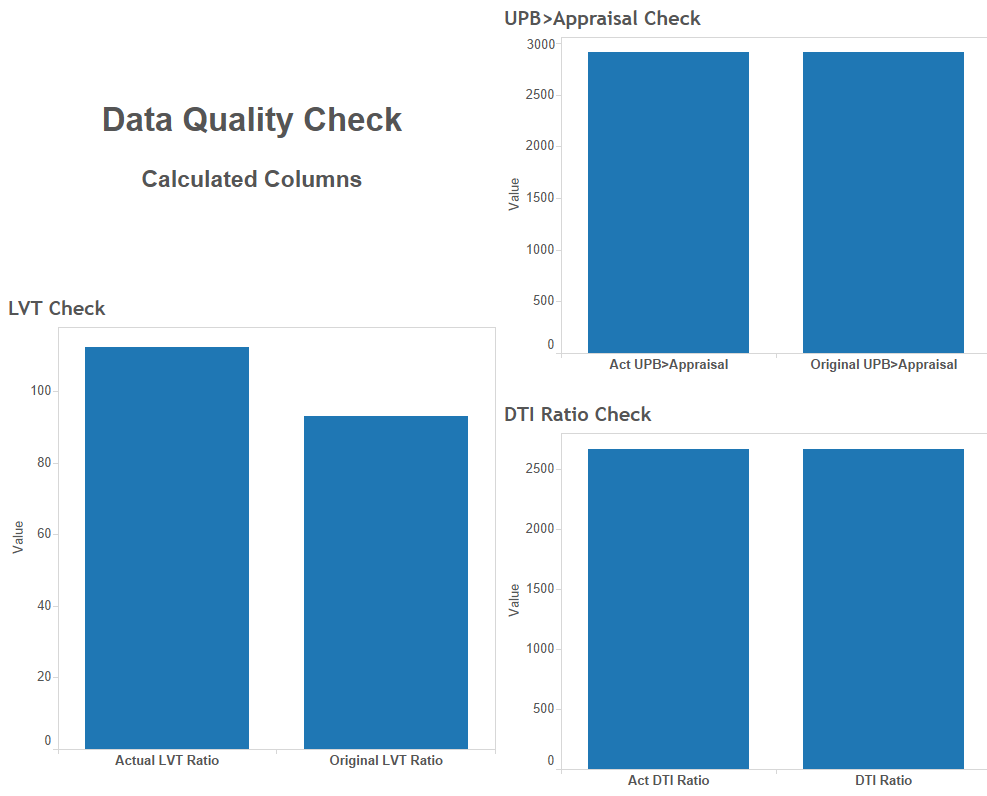
Tool: Tableau.

Steps:

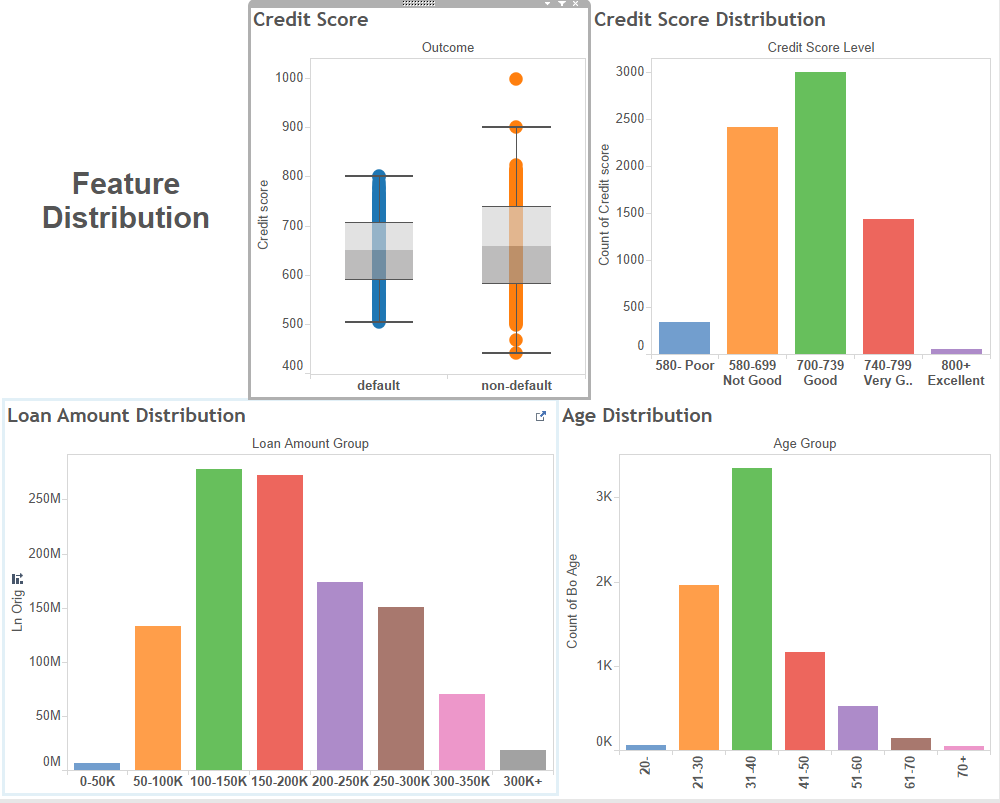
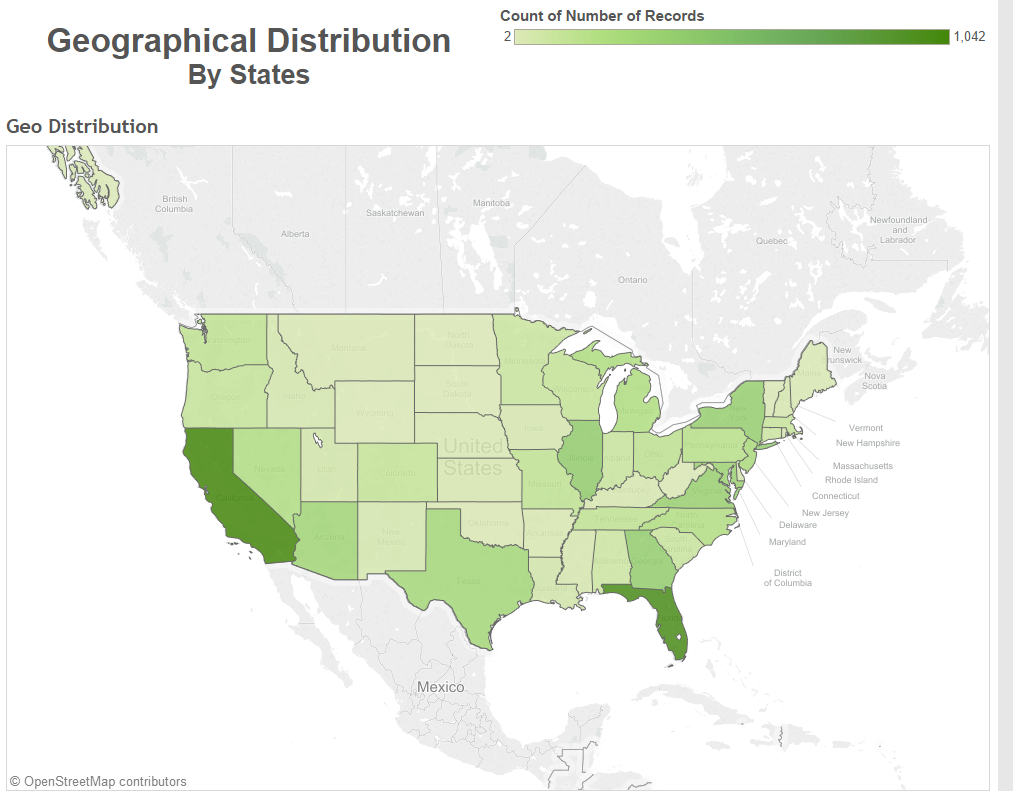
1. According to the chart, noticed that the group of mortgage status “Active” has been classified as non-default. However, in the real life, these active mortgage may turn into a default as soon as the borrowers are no longer able to pay for that. Therefore, all the “active” records should be dropped from the non-default group, which decrease the record number by 7893.

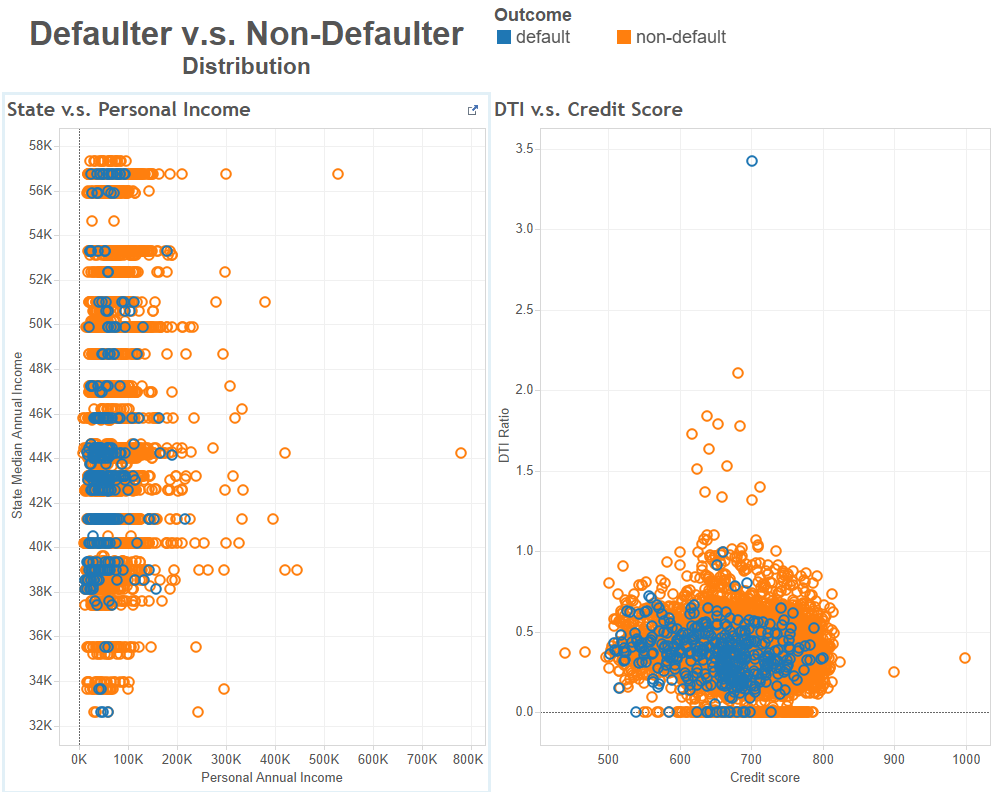
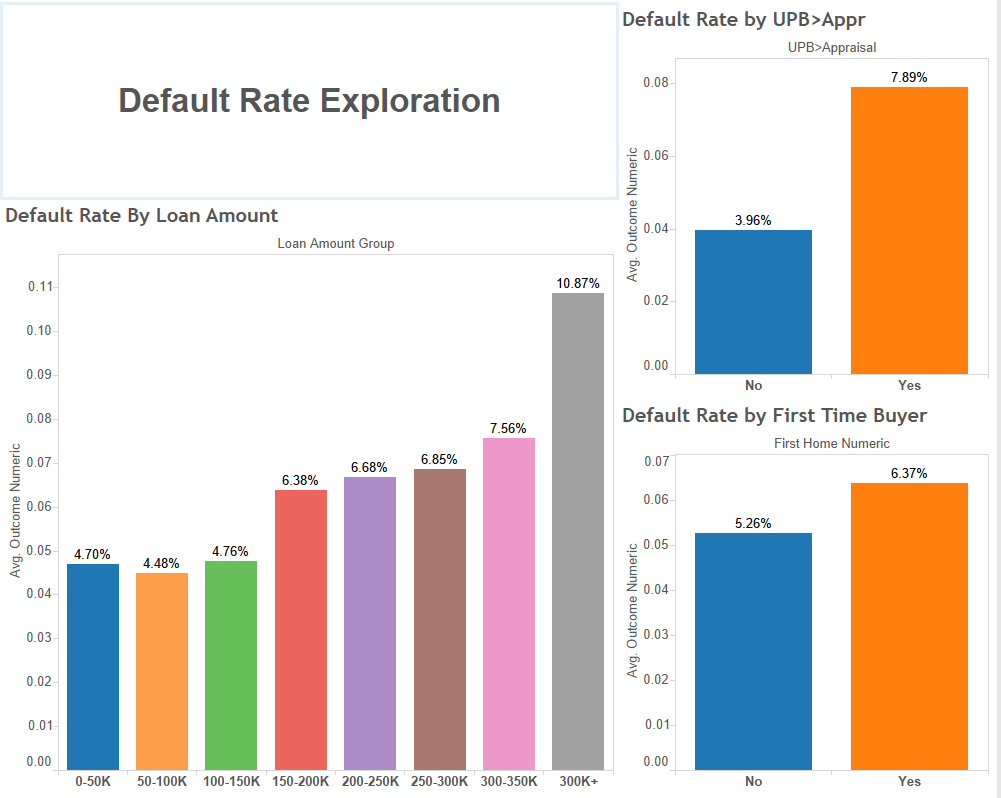
 

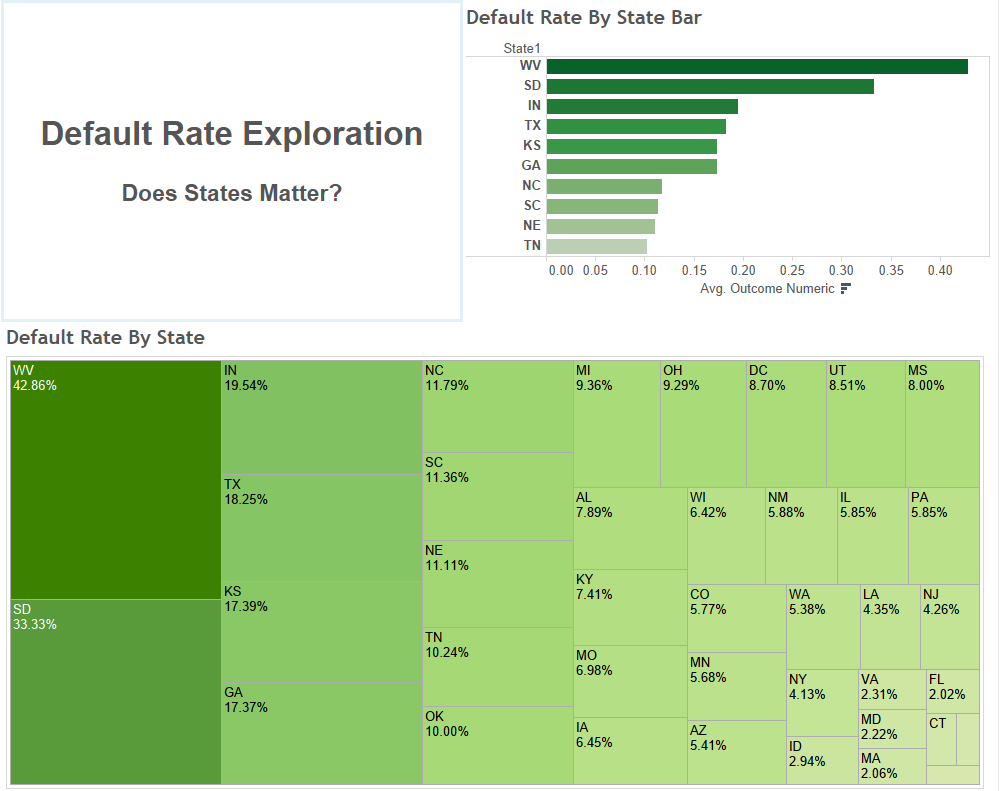
1. There are calculated columns in the dataset need to be checked by creating new calculated columns in Tableau. After the process, the data inaccuracy on LVT ratio have been discovered. The new calculated column will be used as the correct data for further analysis.



1. Further data exploration has been performed in multiple ways. First, mortgage distribution using different features has been visualized using bar chart, geo-map and box plot. Second, in order to explore the relationships between default/non-default and different features, scatter plot charts have been used. Third, because of the significant sample size difference between default and non-default, the KPI default rate has been created to measure the impact of different features. Bar charts and heat maps are used in default rate analysis.



**2.2** **Build classification models**

Tool: XLMiner

Steps:

1. Data screening and pre-processing:
   1. Filter out the rows having “active” in Status column.
   2. Create calculated column “Act\_LTV\_Ratio” to replace the “Orig\_LTV\_Ratio\_Pct”. The formula used is Ln\_Orig divided by the smaller between orig\_apprd\_val\_amt and pur\_prc\_amt. If orig\_apprd\_val\_amt = 0, then use pur\_prc\_amt[[1]](#footnote-1).
   3. Drop unused column: Status, state.
   4. Convert binary categorical columns’ values into 1 and 0: First\_home, OUTCOME.
2. Outlier identification and data cleansing: according to the result of visualization. The following outliers have been identified and need to be deleted from the dataset.
   1. One column has a credit score of 999, and with monthly income of $1,187, the borrower managed to pay-off a $228,000 loan. The house price is $50,000, which is far below the loan amount. This row shows obvious abnormal features of fraud data, and need to be deleted from the original dataset.
3. Partition data
   1. Data partition has been performed through standard partition option with a 60% training and 40% testing. Random seeds has been set as 12345. Same training and testing partition will be used in all the classification models.
4. Build models
   1. Set cut-off values. Cut-off value has been set to 0.05538 (5.538%), the generic percentage of default for the dataset.

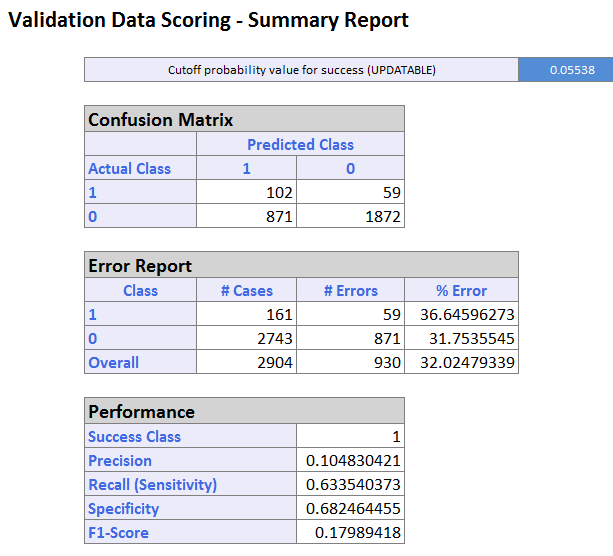
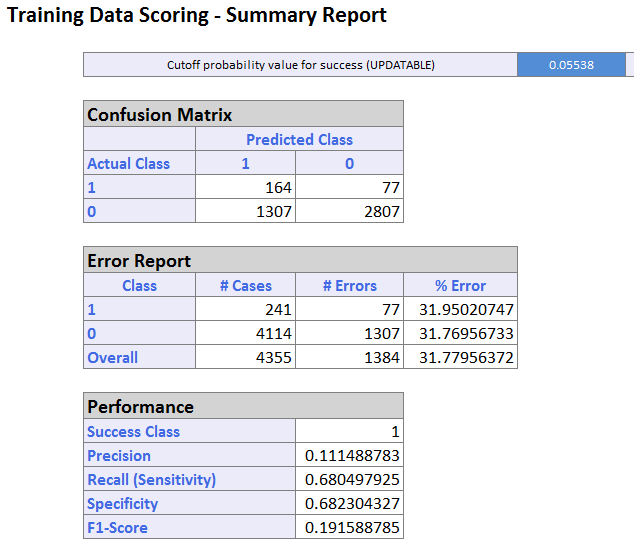
Reason: According to the scenario, the bank (lender)’s goal is to identify the potential defaulters so that they can purchase secondary insurance to prevent potential loss. Also, the cost of misidentifying a non-defaulter as a defaulter is much less than missing a real defaulter. Therefore, the model is expecting an unfair measurement of accuracy for a two-side classification. A cut-off value of 50% is not applicable in this case.

* 1. Based on the cut-off value. Classification has been performed using the following machine learning algorithms: Logistic regression, CART (Classification and Regression Tree) and Random forest. Default settings are applied for CART & Random Forest.

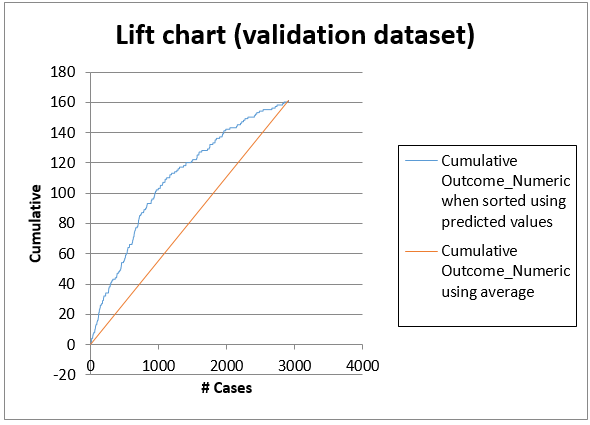
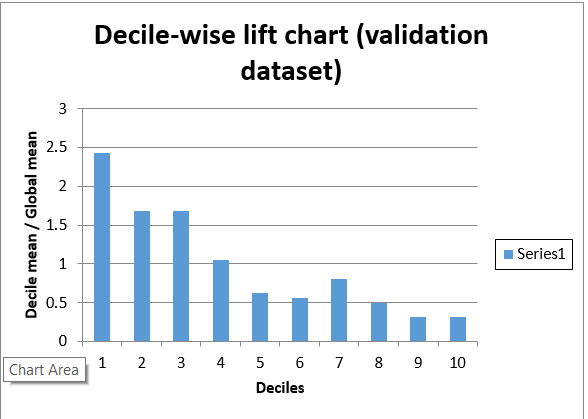
**2.3 Performance evaluation**

Measurement: confusion matrix. Lifting chart. Decile-wise lift chart.

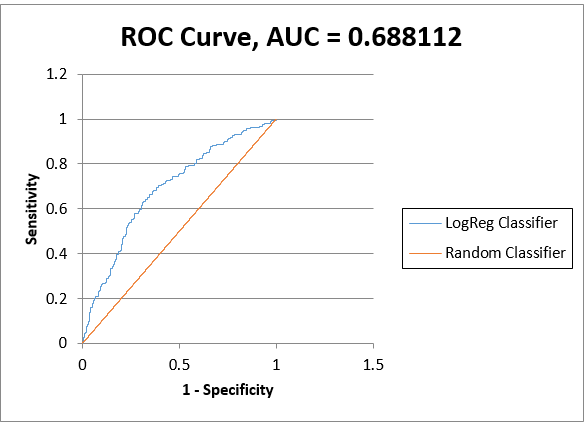
1. **Logistic Regression:** 
   1. In the confusion matrix of training dataset, the error rate for defaulter and non-defaulter are both close to 32%, while in the testing dataset, the error rate of defaulter increased to 36.6% and for non-defaulter it still remain lower than 32%. The overall error rate for both training and testing are similar (32%)



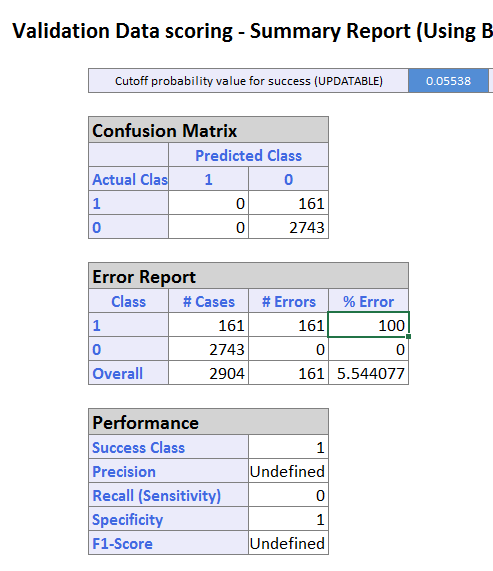
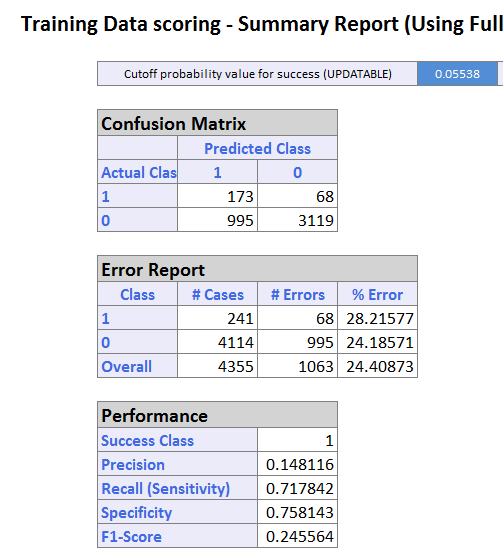
* 1. In the lift chart for testing dataset, the curve is above the straight line which indicates random classification rules. The Decil-wise lift chart also has higher head and lower tail. Both lift chart indicate the classification model performs better than a random model.

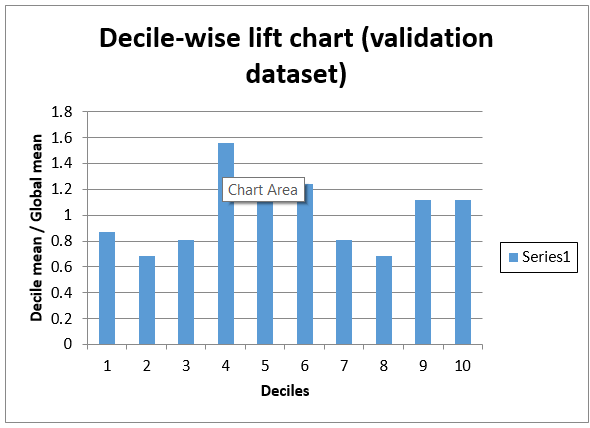
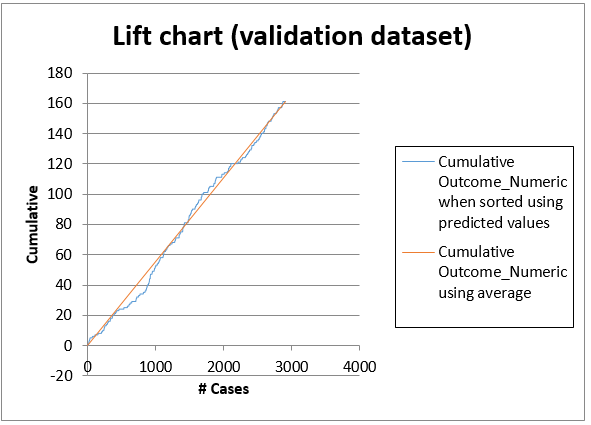
* 1. Roc Curve for testing dataset is above the straight line which indicates random classification rules. AUC = 0.688112, which also indicate the logReg Classifier performs better than a random classifier.



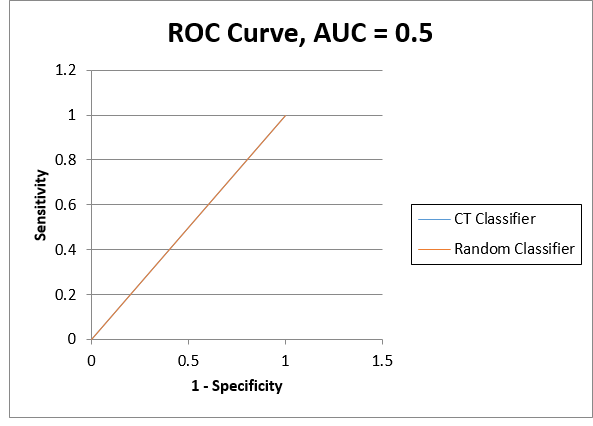
1. **CART**
   1. In the confusion matrix of training dataset, the error rate for defaulter is 28% and for non-defaulter is 24%, while in the testing dataset, the error rate of defaulter increased to 100% and for non-defaulter dropped down to 0%. The overall error rate for training is 24% and for testing is 5%



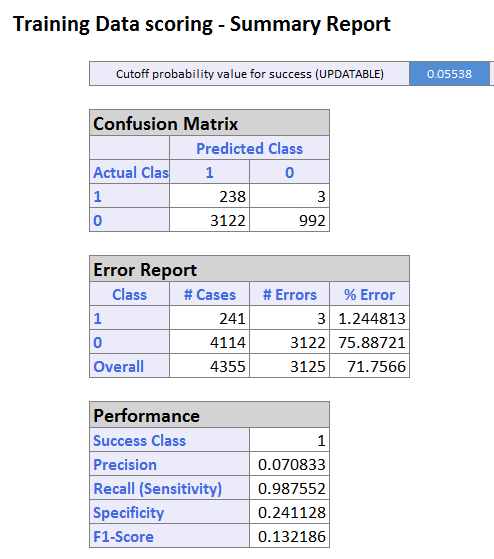
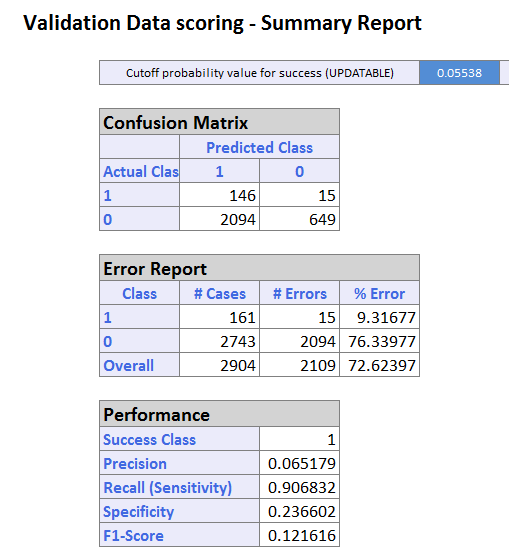
* 1. In the lift chart for testing dataset, the curve is along with straight line which indicates randomly classify the outcome by using average. The Decil-wise lift chart also do not show higher head or lower tail. Both lift chart indicate the classification model performs no better than a random classification model.



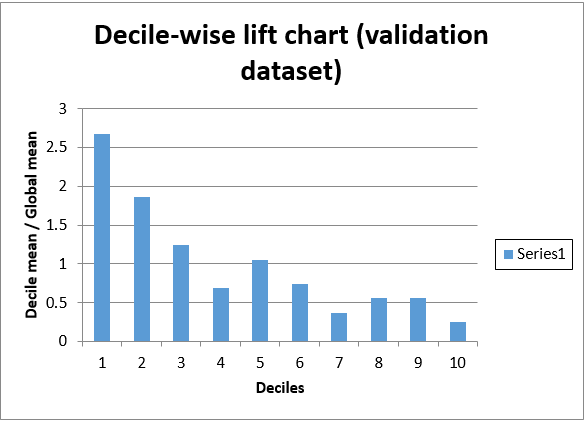
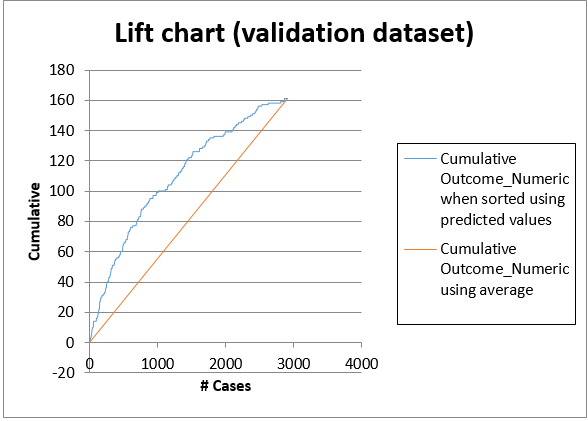
* 1. Due to the 100% error rate on one side of classified outcome, ROC Curve cannot be generated. AUC = 0.5, which indicates the model is no better than a random classifier.



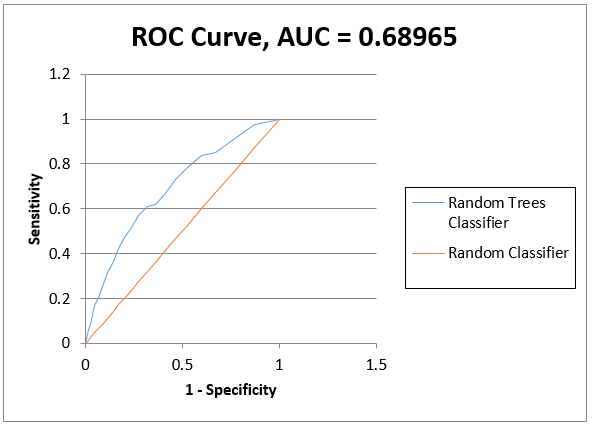
1. **Random Forest**
   1. In the confusion matrix of training dataset, the error rate for defaulter is 1.24% and for non-defaulter is 76%, while in the testing dataset, the error rate of defaulter increased to 9.31% and for non-defaulter remains 76%. The overall error rate for training and testing are both around 72%

* 1. In the lift chart for testing dataset, the curve is above the straight line which indicates a random classifier. The Decil-wise lift chart also has higher head and lower tail. Both lift chart indicate the classification model performs better than a random classifier.



* 1. Roc Curve for testing dataset is above the straight line which indicates random classification rules. AUC = 0.68965, which also indicate the logReg Classifier performs better than a random classifier.



1. Model Comparison:
   1. CART has the lowest overall error rate in the testing dataset. However, it failed to identify any defaulter, the performance is no better than a random classifier. Therefore, it is eliminated from the candidates of the classification models.
   2. Logistic regression has the lower overall error rate than random forest. However, the error rate of classifying a defaulter is much lower using random forest.
   3. AUC value and lift charts indicate random forest has a better performance than logistic regression.

**2.4 Model selection and recommendation**

1. Model selection:

In order to perform model selection, the following assumption need to be made based on the operation of bank and financial institution (lender) in real life.

Assumption 1: lender will purchase secondary insurance for any borrowers who has been classified as a potential defaulter in order to avoid their potential loss due to a mortgage default.

Assumption 2: the premium payment of purchasing a secondary insurance is much lower than the cost of missing a defaulter in classification. Assume the annual premium rate for a Lenders mortgage insurance is in the range of 0.66% to 0.75%[[2]](#footnote-2).

Assumption 3: the average mortgage payment duration is around 10 years. Defaults normally happen during after the 5th year of mortgage. The assumption is made conservatively since economic and unemployment rate changes significantly in a 10-year period, according to Federal Reserve Bank, during economic crisis, the default rate may increase to over 10%[[3]](#footnote-3).

1. Model Selection

With the three assumptions above. Random Forest has been chosen as the recommended classification model for this scenario. The following reasons are presented for reference:

* 1. Comparison has been made between Logistic Regression and Random Forest by calculating the expected costs based on the error rates of both models (testing + training).

The expected cost for Logistic Regression is

0.75%\*(102/2+871+1307+164/2)\*10+59+77-164-102 = 43.325

The expected cost for Random Forest is

0.75%\*(146/2+2094+3115+238/2)\*10+15+3-146-238 = 39.075

As a result, the expected cost of using Random Forest is lower than Logistic regression with fairly conservative assumptions[[4]](#footnote-4)

* 1. Default rate may increases due to external factors like economy downhill and unemployment rate increases.

1. Recommendation
   1. Considering the current resources and available options, we recommend Random Forest as the best Classification models to identify mortgage defaults.
   2. In order to make comprehensive decisions and finding the best model. Other models such as boosting trees should be considered, tested and compared to the current Random Forest Model.
   3. Once new data and factors are collected, both model and expected cost calculation should be refined and updated for future use.

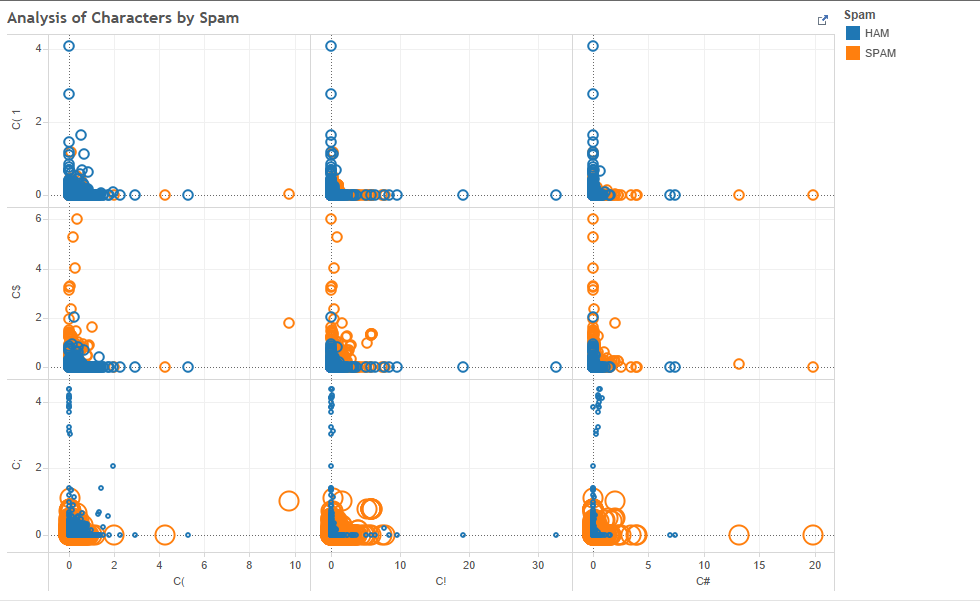
**3. Spambase**

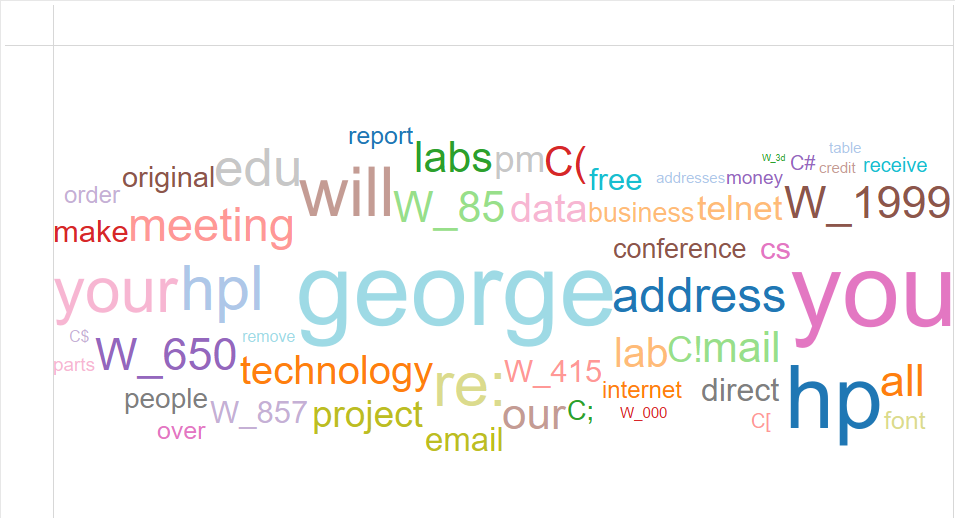
Analyzing the dataset

* There are total 48 attributes of type words
* Total 6 attributes of type characters like C
* 1 real attribute of type average of CAPITAL LETTERS
* 2 integer attributes of type total and longest of the length of CAPITAL LETTERS

**3.1 Exploratory Analysis:**

Tool: Tableau

* Scatter Plot of Characters in the list by Spam
* Word Clouds showing Ham words by average of the percentage of the words



* Word Clouds showing Spam words by average of the percentage of the words



**3.2 Building Models**

Tool: XLMiner

Steps:

1. Data Screening and Pre-processing
2. Performed Feature Selection: Feature selection is performed to select a subset of relevant features for use in model construction. Performing feature selection gives you the chi-squared P-Value which helps in determining whether a predictor accepts or reject ‘Null Hypothesis’. There is no significant difference between observed and expected frequencies

2. Data Partition

1. Data partition has been performed through standard partition option with a 60% training and 40% testing. Random seeds has been set as 12345. Same training and testing partition will be used in all the classification models.

3. Build Models

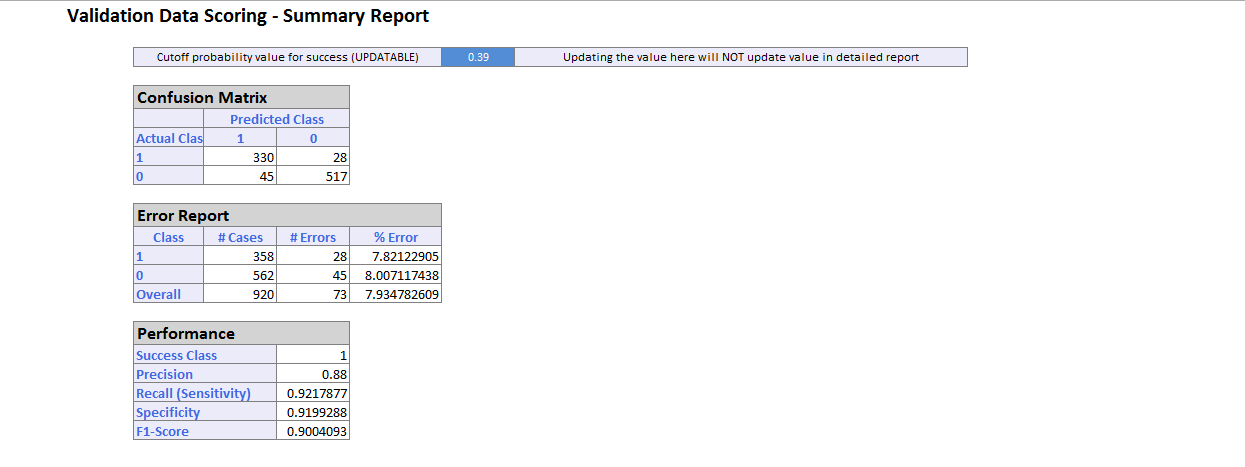
1. Considering 1813 emails tagged as spam from 4601 emails. The initial cutoff probability of success is taken as 0.39. If the probability of success for an email is less than this value then the email would be a non-spam email and if it greater than this value then the email would be a spam email.
2. Based on the cut-off value: Classification has been performed using the following machine learning algorithms: Logistic regression, CART (Classification and Regression Tree) and Random forest. Default settings are applied for CART & Random Forest.

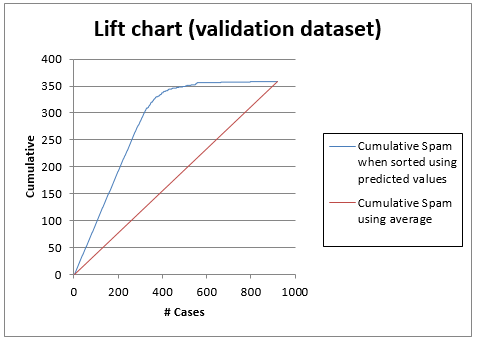
**3.3 Model Evaluation**

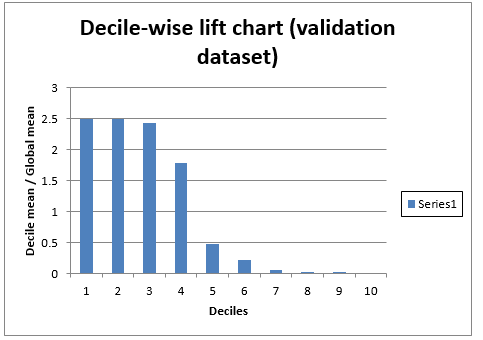
Measurement: Confusion Matrix, Lift Chart, Decile-wise Lift Chart

1. Logistic Regression:

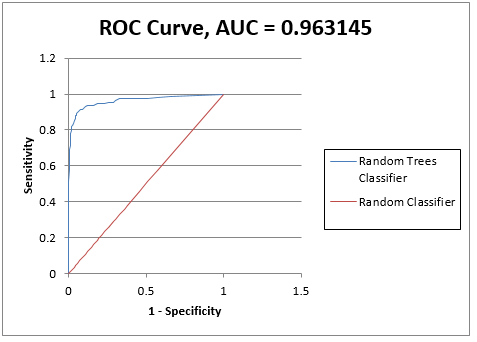
For detecting spam messages, the error rate of spam messages from the confusion matrix of validation dataset 7.82 %



The curve of the lift chart for validation dataset is above the straight line which is good. The decile-wise lift chart of the validation dataset has higher heads and lower tails. The top 2 decile contains 20% of the emails most likely to be spam emails. Whereas the bottom 2 decile contains 20% of the emails least likely to be the spam emails.

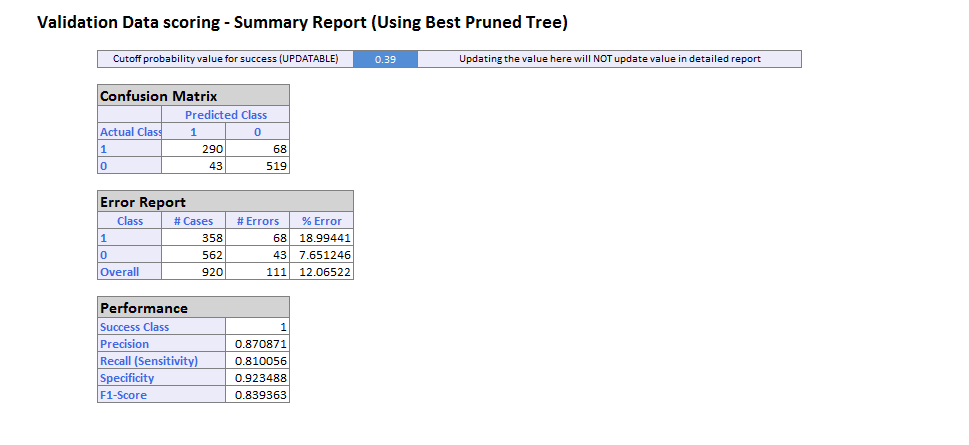


ROC Curve of the validation dataset is in the top left corner which indicates that the model performs better model than a random model

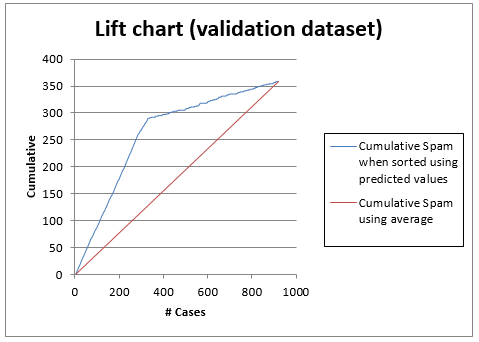
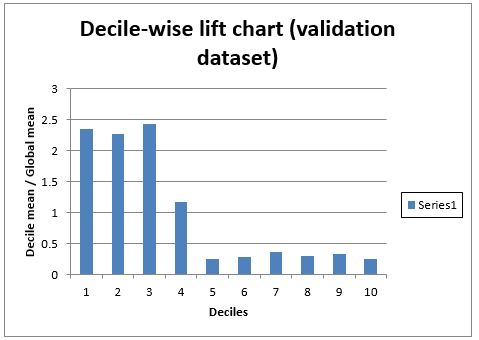


2. CART

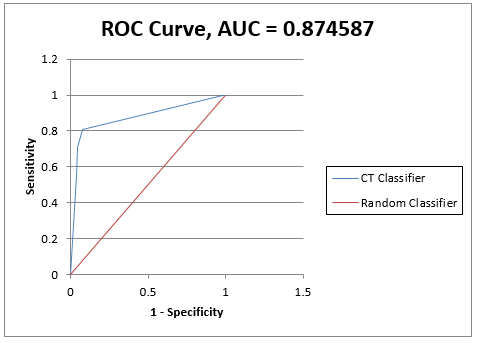
1. The error rate of spam messages from the confusion matrix is about 19% which is high compared to the error rate of Logistic regression



b. The curve of the lift chart for Validation dataset of CART is above the straight line. Whereas in the decile chart the higher and the lower heads are out of order which indicates that the model is not good compared to a random classifier model which should be in good staircase order from left to right.

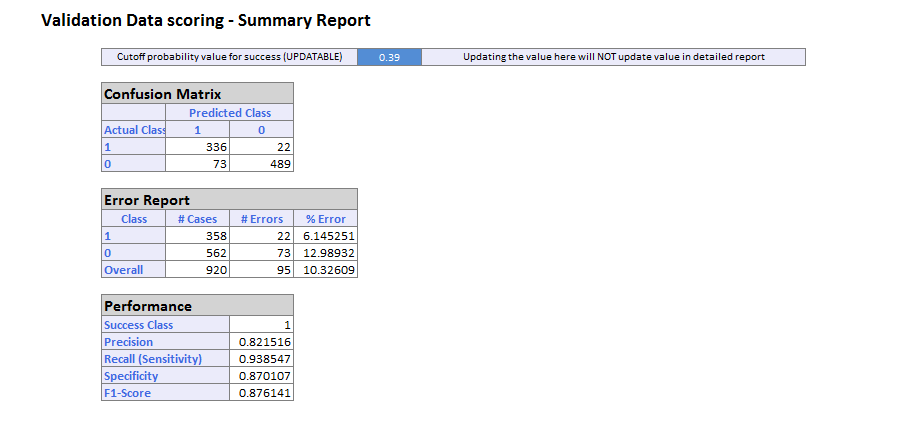


c.The ROC curve is away from top left corner which again indicates that the model is not good

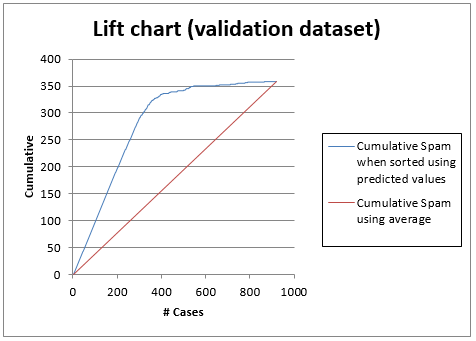
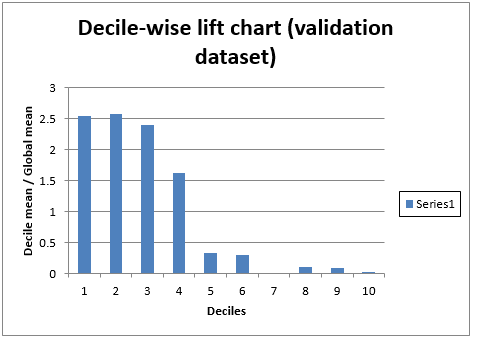


3. Random Forest

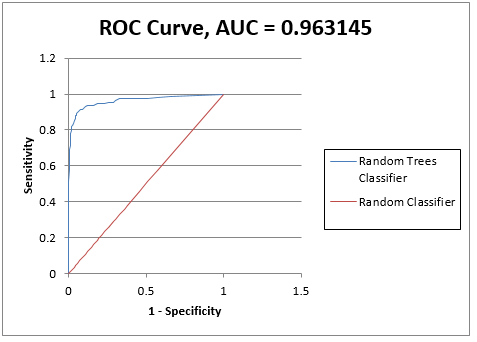
1. The error rate of the spam messages from the confusion matrix of Random Forest using Random Forest is 6.14% which is low compared to Logistic Regression and CART.



1. The Lift chart of Random Forest is above the straight line which is good. The decile-wise lift chart shows that the higher heads and the lower tails with decreasing order from which it can be determined that this is a good model compared to a random classifier.



1. The ROC curve of the Random Forest is above the straight line and near the top left corner of the chart



**3.3. Model Comparison**:

1. The overall error rate of Logistic Regression is 7.9 % which is lower compared to Random Forest which is 10.32%. But for detecting the Spam emails from the validation dataset the error rate of Random Forest is lower 6.14% compared to Logistic Regression which is 7.82%.
2. CART has the highest overall error rate in the validation dataset. And, again from the decile-wise lift chart and ROC curve it can be seen that the model is not good for detecting the spam emails.
3. For detecting spam emails Random Forest provides better performance compared to Logistic Regression

3.4 **Model evaluation and Recommendation**

1. Assumption: According to the dataset, if we assume the initial cutoff probability of success to be 0.39 or 39% then Random Forest model is good taking into account the error rates which are:

Logistic Regression: 7.9%

CART: 18.99%

Random Forest: 6.14%

If we consider the initial cutoff probability of success to be 0.5 (default) or 50% for detecting spam emails then the error rates are:

Logistic Regression: 12.29%

CART: 18.99%

Random Forest: 10.89%

Hence, considering both the conditions Random Forest is a good model

**4 Blog Feedback**

**4.1** **Building Prediction Models**

Tools: RStudio, Python

Steps:

1. Data Preprocessing:

* Concatenated all the csv files having test data with the help of Python

Python Code for concatenation:

import glob

import pandas as pd

path ='C:/Users/vanwu/Desktop/INFO 7390 ADS/Midterm/BlogFeedback/test'

allFiles = glob.glob(path + "/\*.csv")

frame = pd.DataFrame()

list\_ = []

for file\_ in allFiles:

   df = pd.read\_csv(file\_, header = None)

   list\_.append(df)

frame = pd.concat(list\_)

frame.to\_csv('blog\_data\_test.csv', sep=',')

* Looking into the data and referring the paper, we inferred that the significance of 200 bag-of-words is very weak in predicting the number of comments in next 24 hours. Also, the trend of the words may change which is not predictable. Hence, we deleted the 200 columns containing bag-of-words.

1. Build Models:

* Prediction has been performed using the following machine learning algorithms: Multiple Linear Regression, CART(Classification and Regression Tree) and Random Forest.

**4.2** **Performance Evaluation**

Tool: RStudio

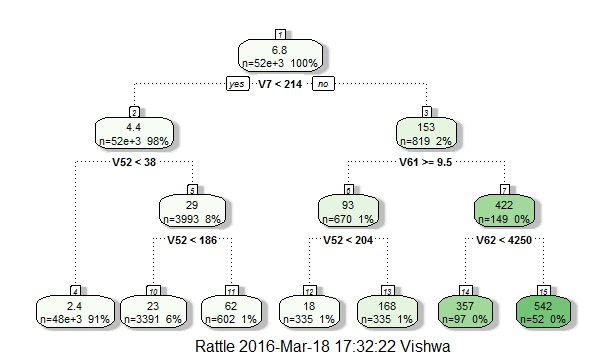
Measurement: RMSE value

1. Linear Regression:

* Applying the linear regression formula in training data and predicting the rmse value by applying the model to the testing data gives rmse value to be 25.24 and multiple R-squared value to be 0.36

1. CART:

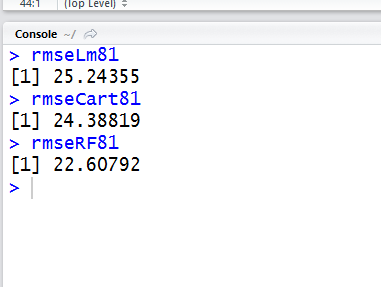
* The packages used for CART algorithm for calculating RMSE value and visualization of the tree are: rpart, rattle, dynamicGraph, rattle, rpart.plot, cartFit81
* The RMSE value calculated with CART is 24.39 which is less compared to Regression model.
* The regression tree that we got from CART algorithm is:



3. Random Forest:

* The package used for Random Forest in R is ‘randomForest’
* The RMSE value calculated with Random Forest is 2.20 which is the lowest of all the models.

The list of RMSE values calculated using all the algorithms in R are:



**4.3** **Model Selection and Recommendation**

1. Model Selection:

In order to perform model selection, the following assumptions need to be made:

**Positive**

Assumption 1:

More Total/Average/Max/Min (1,3-5, 6, 8-10, 11, 13-15) comments before basetime (51)/ within the 24 hours before basetime (52)/ within 48-24 hours before basetime, there may be more comments in next 24 hours (53).

Assumption 2:

More Total/Average/Max/Min (26, 28-30, 31, 33-35, 36, 38-40) trackbacks before basetime(56)/ within the 24 hours before basetime (57)/ within 48-24 hours before basetime(58), there may be more comments in next 24 hours.

Assumption 3:

More Average/Max/Min (278 - 280) of comments the parent posts (277) received, there may be more comments in next 24 hours.

**Negative**

Assumption 1:

Longer the hours between publication & basetime (61), lesser comment will be in the next 24 hours.

**Uncertain**

Assumption 1:

Weekday of post publication (270-276) may affect the comments in the next 24 hours.

Assumption 2:

Weekday of basetime (263-269) Sat, Sun More

Assumption 3:

Length of post (62), too short or too long, less comments

Assumption 4:

Bag of words (63-262) not known, trends change.

Assumption 5:

Standard deviation (2,7,12,17,23,27,32,37,42,47) may affect consistency

Assumption 6:

24 hours after publication & before basetime:

1. Published at least 24 hours before basetime. More comments/links (54, 59), more comments in the next 24 hours. Positive
2. Published less than 24 hours before basetime. -**--- to be determined.**

Assumption 7:

Difference of comments/Link (55,60) number between last 24 hours and last last 24 hours.

1. If >=0, attention grows/maintains, positive
2. If <0, attention drops, negative

At first we calculated the RMSE value of all the algorithms using all the parameters of the training and testing dataset. The results we gained using 281 parameters are:

Linear Regression: 21.96

CART: 27.06

Random Forest: 28.39

According to this, regression model would be a good model.

After close analysis, we found out that the significance of the bag-of-words on the model very less. Hence, we removed columns with bag-of-words from the csv files and use those files for all the algorithms. The result we received with 81 parameters are:

Linear Regression: 25.24

CART: 24.39

Random Forest: 22.61

2. Model Recommendation:

* If we take into account all the parameters, then Linear Regression is good, but if we remove bag-of-words columns and evaluate the model Random Forest is good.

http://www.cs. bme.hu/~buza/pdfs/gfkl\_buza\_social\_media.pdf

1. https://en.wikipedia.org/wiki/Loan-to-value\_ratio [↑](#footnote-ref-1)
2. https://en.wikipedia.org/wiki/Lenders\_mortgage\_insurance [↑](#footnote-ref-2)
3. https://research.stlouisfed.org/fred2/series/DRSFRMACBS [↑](#footnote-ref-3)
4. The highest premium rate has been applied. [↑](#footnote-ref-4)