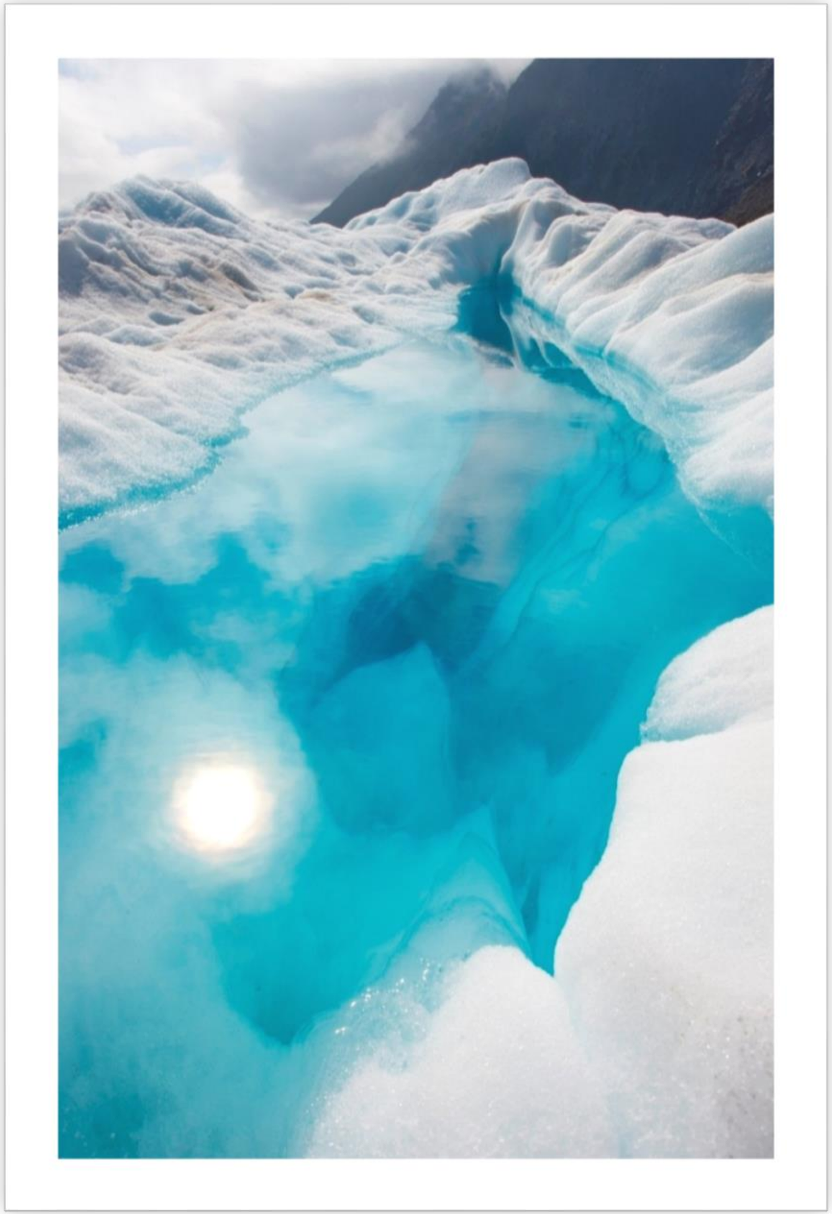


Advanced Data-Science Midterm Project Report

TEAM:7

Submitted by:
Jyotirmayee Mahanandia
Aditya Shinde
Rachitha Dhanraj

Question 1: West Roxbury dataset

The West Roxbury dataset consists data that describes the total value of the houses in West Roxbury based on 14 features as given below. The objective is to predict the total value of the houses by building prediction models namely Multiple Regression Model, CART and Random Forest Models.

Attributes:

1. TOTAL_VALUE– total value of the house
2. TAX – tax incurred on the value of the house
3. LOT_SQFT – the area of the house
4. YR_BUILT – the year the house was built
5. GROSS_AREA – the gross area occupied by the house
6. LIVING_AREA – the living area occupied by the house
7. FLOORS – number of floors
8. ROOMS – number of rooms
9. BEDROOMS – number of rooms
10. FULL_BATH – number of full bathrooms
11. HALF_BATH – number of half bathrooms
12. KITCHEN – number of kitchens
13. FIREPLACE – number of fireplaces
14. REMODEL – if the house has been remodeled or not.

Here the TOTAL_VALUE is the outcome variable and the REMODEL column is excluded since it's not numeric.

Preprocessing of data: The data had a couple of null values that were deleted using the Trifacta Wrangler tool. No other issues were encountered.

Visualization in Tableau:

Figure1: Total Value vs Tax

From the correlation matrix we find that Tax has a linear relationship with Total Value. Tax is used as a significant variable to predict Total Value of the house.

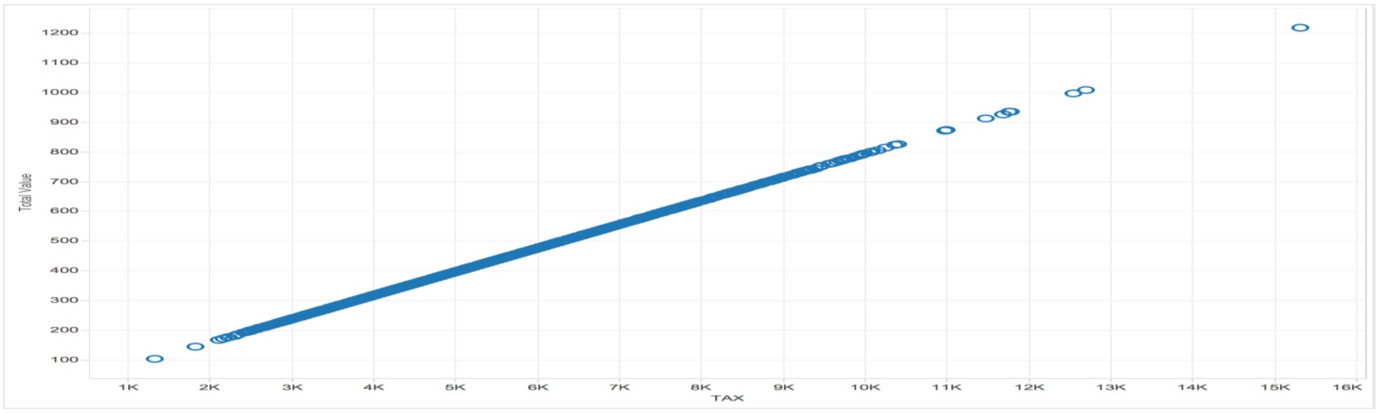


Figure2: Total Value vs Tax considering the year the houses were built

From the figure below we find that year has no significant relationship in predicting the total value of the house.

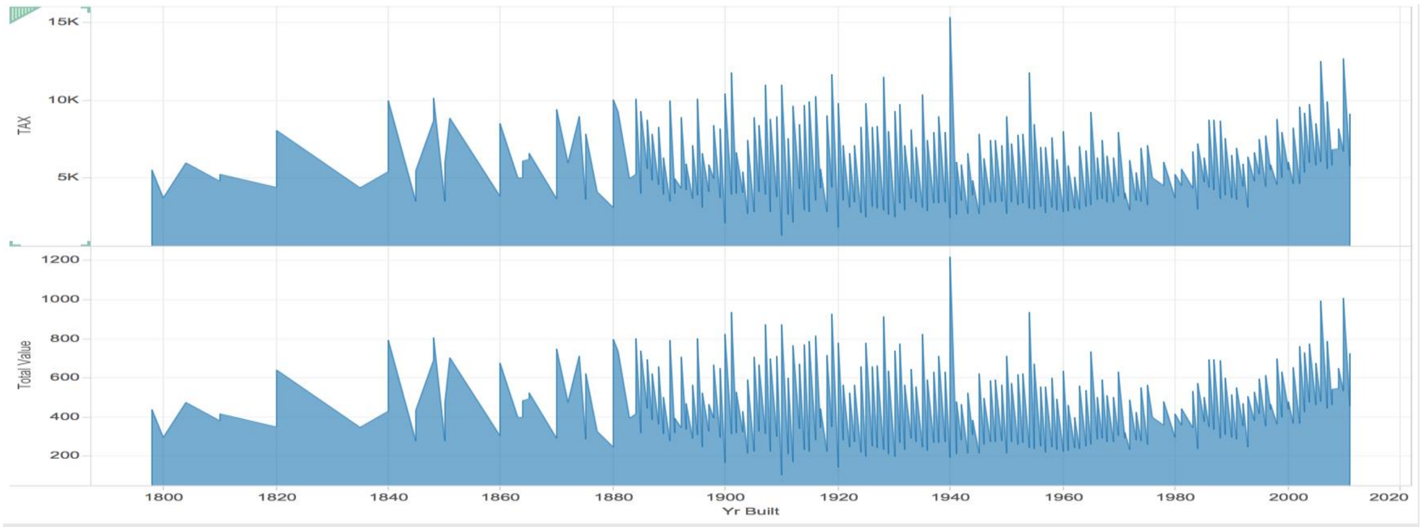
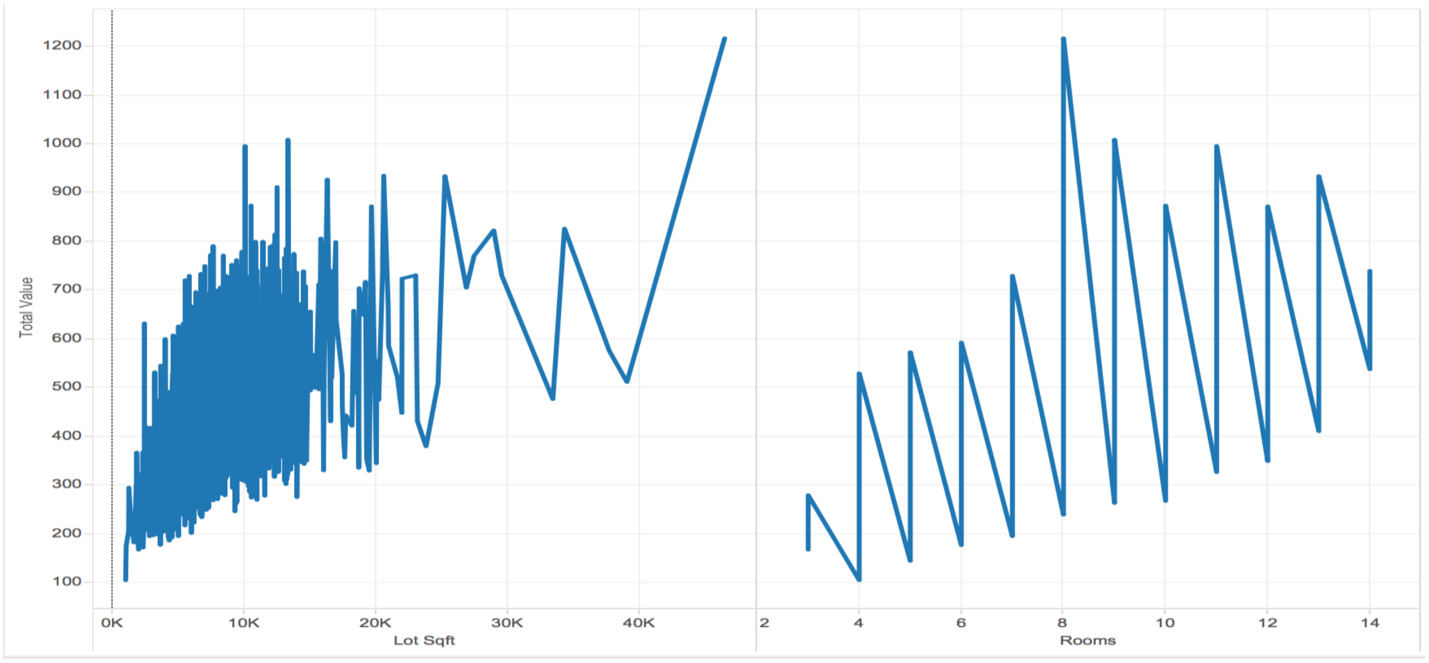


Figure 3: Total Value vs the area in square feet and the number of rooms

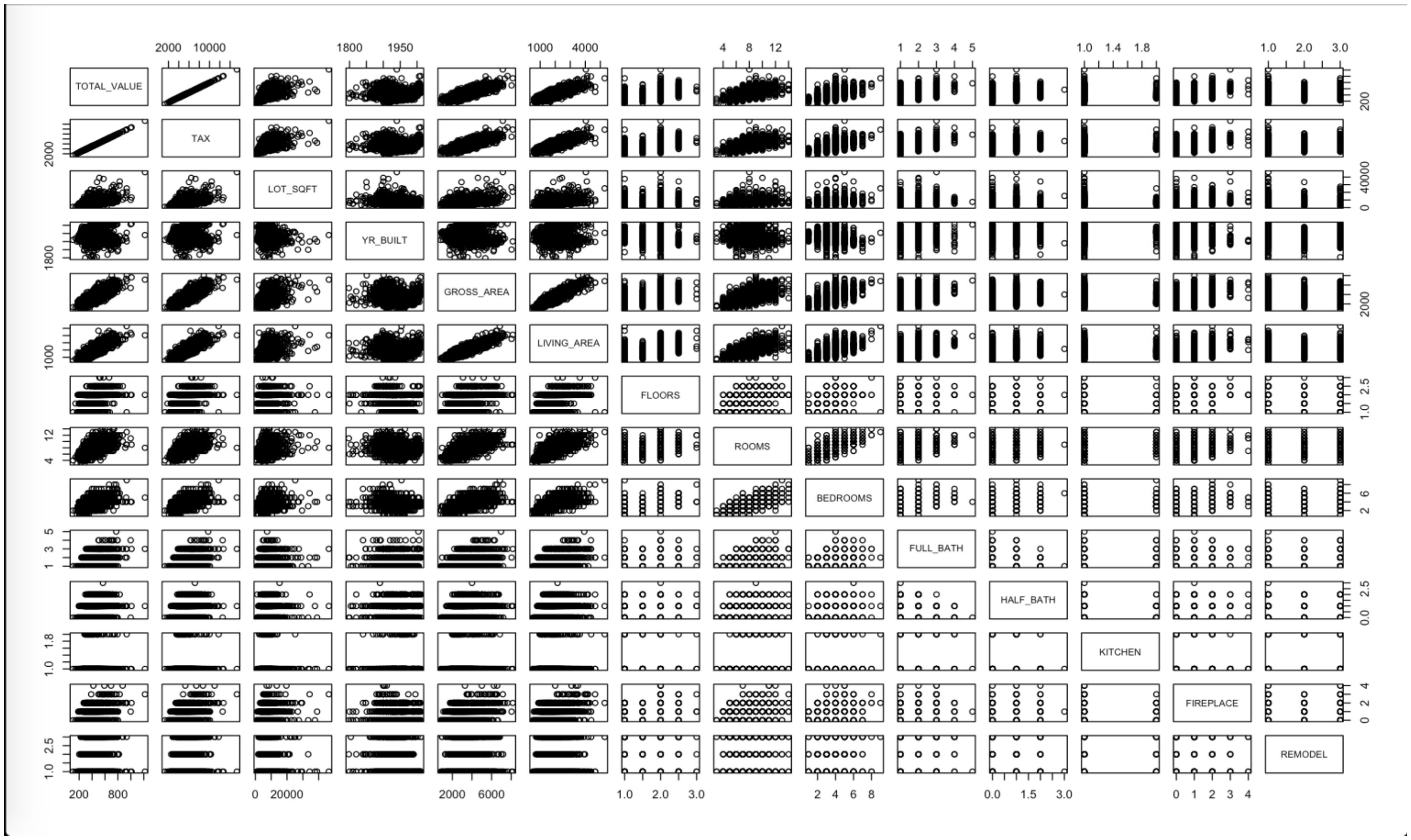
The graphs below shows the relationship between the area in square feet and the total value. It is not necessary for the total value of the house to high if the area is more. Similarly is not required for the total value of the house to be more if the number of rooms are more.

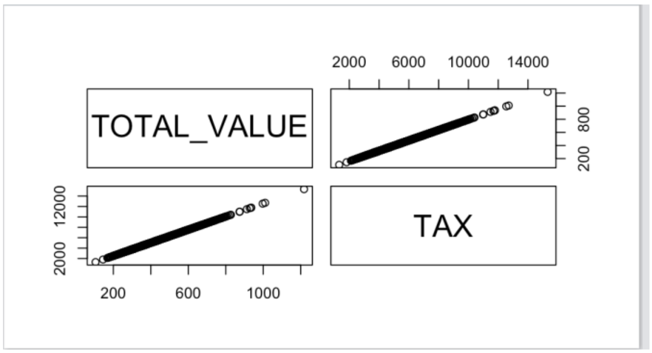


Multiple Regression Model:

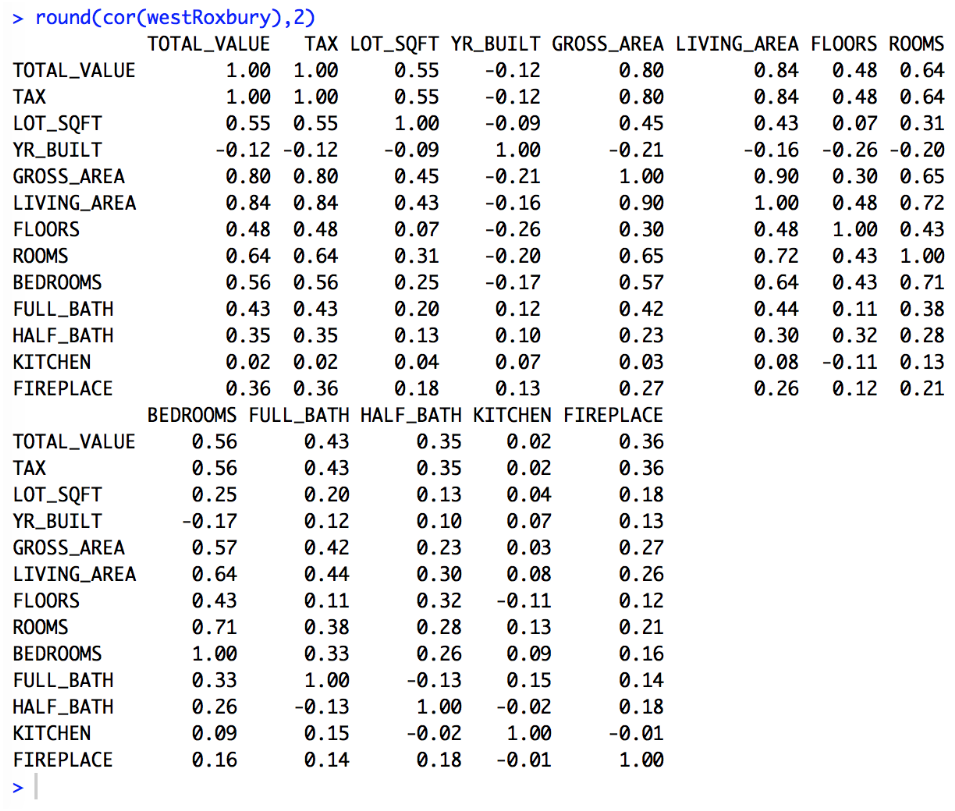
The scatter plot of all the features of the West Roxbury dataset is as given below. We clearly observe that the

TAX variable has a linear relationship with the TOTAL_VALUE.





Also, the correlation of all the variables in R is displayed below where TAX, GROSS_AREA, LIVING_AREA, ROOMS are showing the highest correlation of 1.00, 0.80, 0.84 and 0.64 respectively. We can eliminate one of the variables, for instance GROSS_AREA to avoid multi-collinearity.

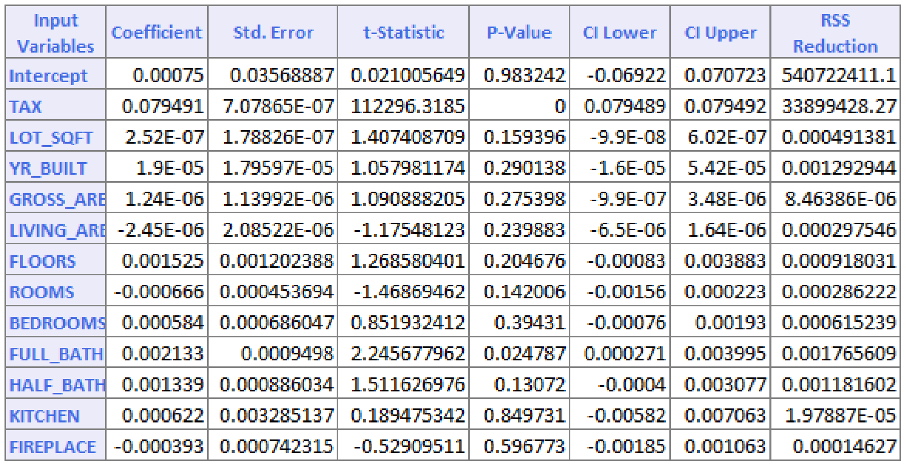


```
> round(cor(westRoxbury),2)
            TOTAL_VALUE   TAX LOT_SQFT YR_BUILT GROSS_AREA LIVING_AREA FLOORS ROOMS
TOTAL_VALUE        1.00  1.00     0.55    -0.12       0.80        0.84   0.48  0.64
TAX                1.00  1.00     0.55    -0.12       0.80        0.84   0.48  0.64
LOT_SQFT           0.55  0.55     1.00    -0.09       0.45        0.43   0.07  0.31
YR_BUILT          -0.12 -0.12    -0.09     1.00      -0.21       -0.16  -0.26 -0.20
GROSS_AREA         0.80  0.80     0.45    -0.21       1.00        0.90   0.30  0.65
LIVING_AREA        0.84  0.84     0.43    -0.16       0.90        1.00   0.48  0.72
FLOORS             0.48  0.48     0.07    -0.26       0.30        0.48   1.00  0.43
ROOMS              0.64  0.64     0.31    -0.20       0.65        0.72   0.43  1.00
BEDROOMS           0.56  0.56     0.25    -0.17       0.57        0.64   0.43  0.71
FULL_BATH          0.43  0.43     0.20     0.12       0.42        0.44   0.11  0.38
HALF_BATH          0.35  0.35     0.13     0.10       0.23        0.30   0.32  0.28
KITCHEN            0.02  0.02     0.04     0.07       0.03        0.08  -0.11  0.13
FIREPLACE          0.36  0.36     0.18     0.13       0.27        0.26   0.12  0.21
            BEDROOMS FULL_BATH HALF_BATH KITCHEN FIREPLACE
TOTAL_VALUE     0.56      0.43      0.35    0.02      0.36
TAX             0.56      0.43      0.35    0.02      0.36
LOT_SQFT        0.25      0.20      0.13    0.04      0.18
YR_BUILT       -0.17      0.12      0.10    0.07      0.13
GROSS_AREA      0.57      0.42      0.23    0.03      0.27
LIVING_AREA     0.64      0.44      0.30    0.08      0.26
FLOORS          0.43      0.11      0.32   -0.11      0.12
ROOMS           0.71      0.38      0.28    0.13      0.21
BEDROOMS        1.00      0.33      0.26    0.09      0.16
FULL_BATH       0.33      1.00     -0.13    0.15      0.14
HALF_BATH       0.26     -0.13      1.00   -0.02      0.18
KITCHEN         0.09      0.15     -0.02    1.00     -0.01
FIREPLACE       0.16      0.14      0.18   -0.01      1.00
>
```

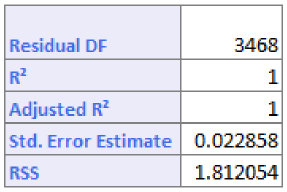
The West Roxbury dataset was split into training (60%) and validation datasets (40%).

The following were the results and their interpretation:

Regression Model

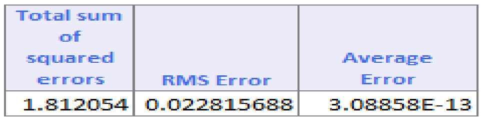


Input Variables	Coefficient	Std. Error	t-Statistic	P-Value	CI Lower	CI Upper	RSS Reduction
Intercept	0.00075	0.03568887	0.021005649	0.983242	-0.06922	0.070723	540722411.1
TAX	0.079491	7.07865E-07	112296.3185	0	0.079489	0.079492	33899428.27
LOT_SQFT	2.52E-07	1.78826E-07	1.407408709	0.159396	-9.9E-08	6.02E-07	0.000491381
YR_BUILT	1.9E-05	1.79597E-05	1.057981174	0.290138	-1.6E-05	5.42E-05	0.001292944
GROSS_ARE	1.24E-06	1.13992E-06	1.090888205	0.275398	-9.9E-07	3.48E-06	8.46386E-06
LIVING_ARE	-2.45E-06	2.08522E-06	-1.17548123	0.239883	-6.5E-06	1.64E-06	0.000297546
FLOORS	0.001525	0.001202388	1.268580401	0.204676	-0.00083	0.003883	0.000918031
ROOMS	-0.000666	0.000453694	-1.46869462	0.142006	-0.00156	0.000223	0.000286222
BEDROOMS	0.000584	0.000686047	0.851932412	0.39431	-0.00076	0.00193	0.000615239
FULL_BATH	0.002133	0.0009498	2.245677962	0.024787	0.000271	0.003995	0.001765609
HALF_BATH	0.001339	0.000886034	1.511626976	0.13072	-0.0004	0.003077	0.001181602
KITCHEN	0.000622	0.003285137	0.189475342	0.849731	-0.00582	0.007063	1.97887E-05
FIREPLACE	-0.000393	0.000742315	-0.52909511	0.596773	-0.00185	0.001063	0.00014627



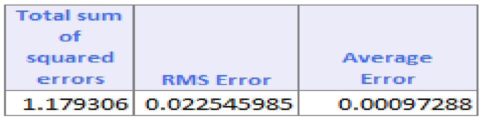
Residual DF	3468
R^2	1
Adjusted R^2	1
Std. Error Estimate	0.022858
RSS	1.812054

Training Data Scoring - Summary Report



Total sum of squared errors	RMS Error	Average Error
1.812054	0.022815688	3.08858E-13

Validation Data Scoring - Summary Report



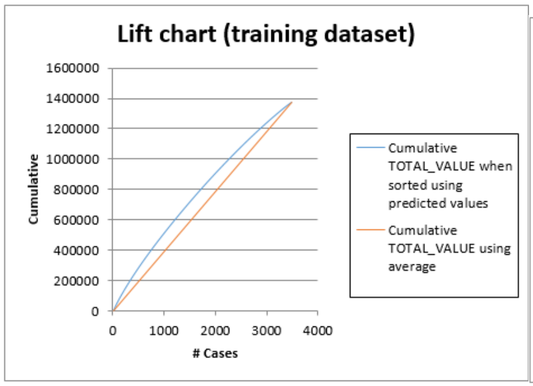
Total sum of squared errors	RMS Error	Average Error
1.179306	0.022545985	0.00097288

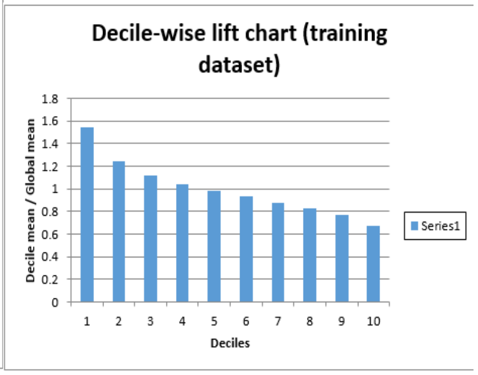
From the above Regression Model, the following can be inferred:

- **P-Value** shows that TAX co-efficient is statistically significant since its 0.

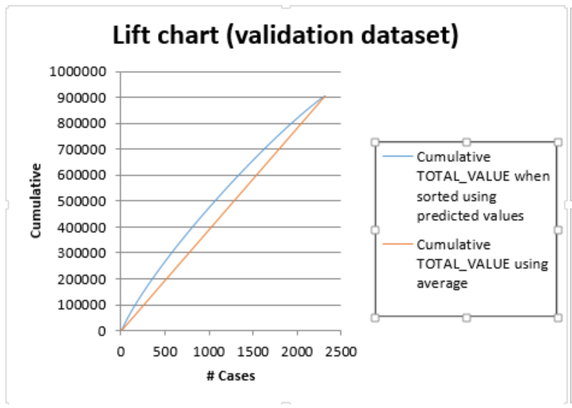
- The Determination of coefficient, **R-squared value** and the **adjusted R-squared values** are exactly 1 which means the actual and the predicted values are very close to each other. The TOTAL_VALUE can be predicted without error from the independent variables.
- **RMSE Error** for both training and validation data are significantly low which indicates a good fit of the model.

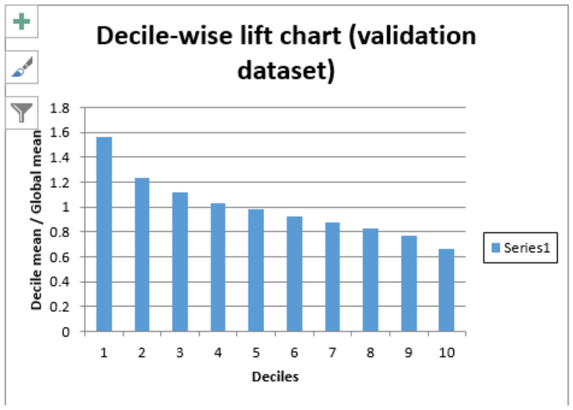
Training Dataset:





Validation Dataset:

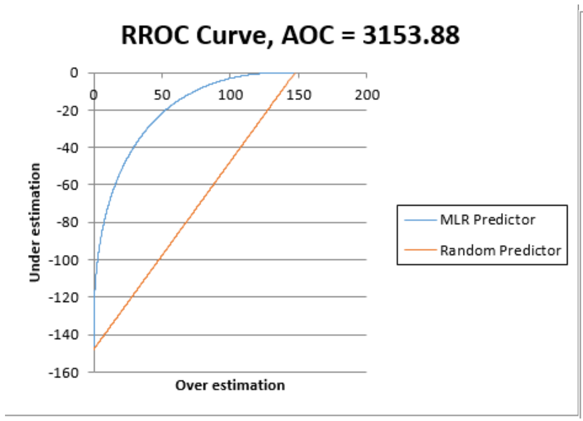


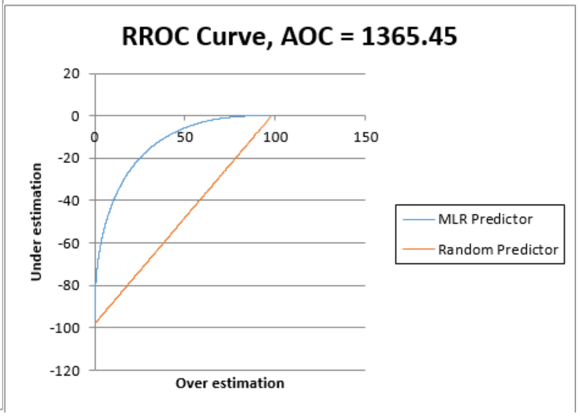


- The **Lift charts** are a measure of effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model. Bigger the area between the base line and lift curve, better is the model.
- The **decile-wise lift curve** is drawn as the decile number versus the cumulative actual output variable value divided by the decile's mean output variable value. This bars in this chart indicate the factor by which the MLR model outperforms a random assignment, one decile at a time

In the first decile, taking the most expensive predicted total value of the houses in the dataset, the predictive performance of the model is almost 1.6 times better as simply assigning a random predicted value.

Training Dataset and Validation Dataset:

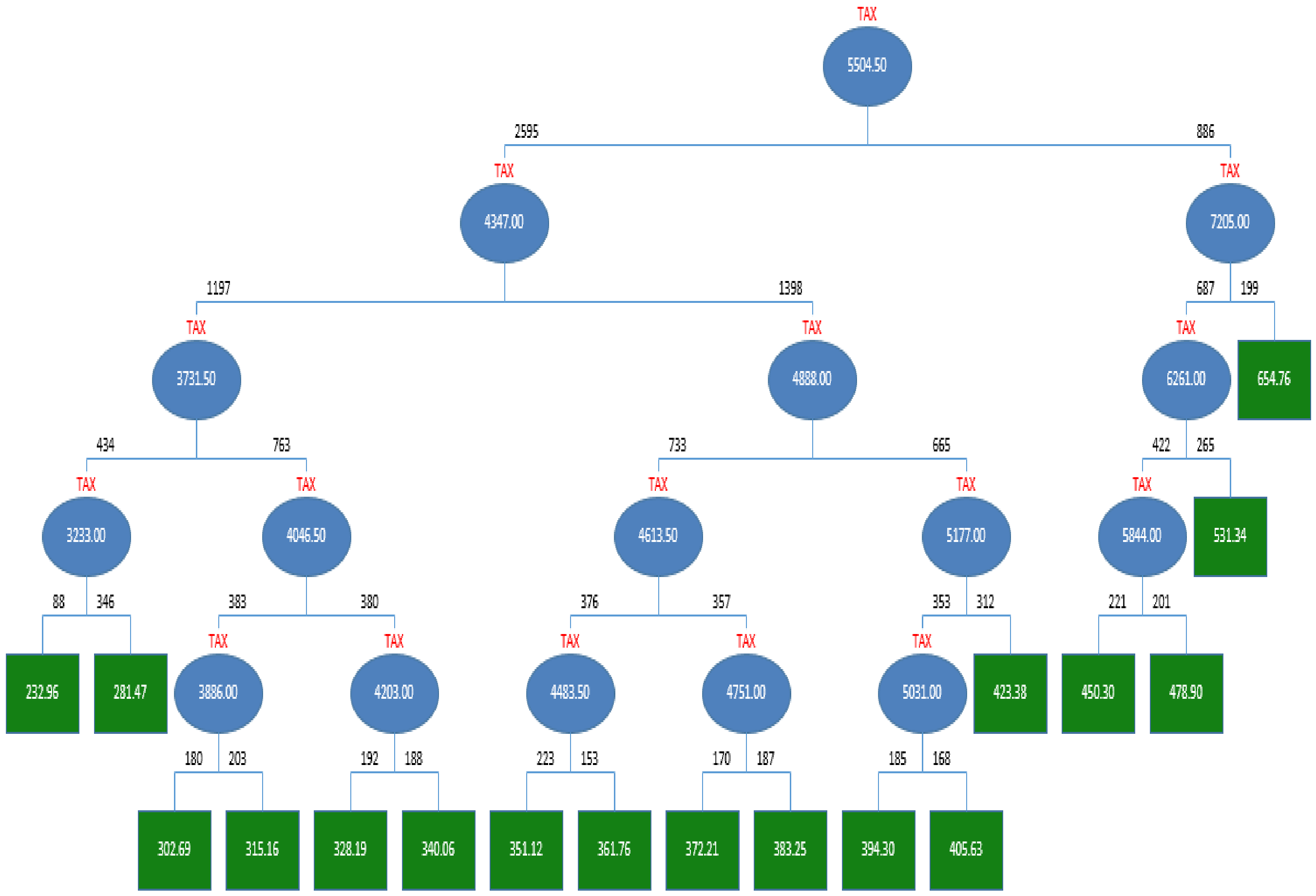




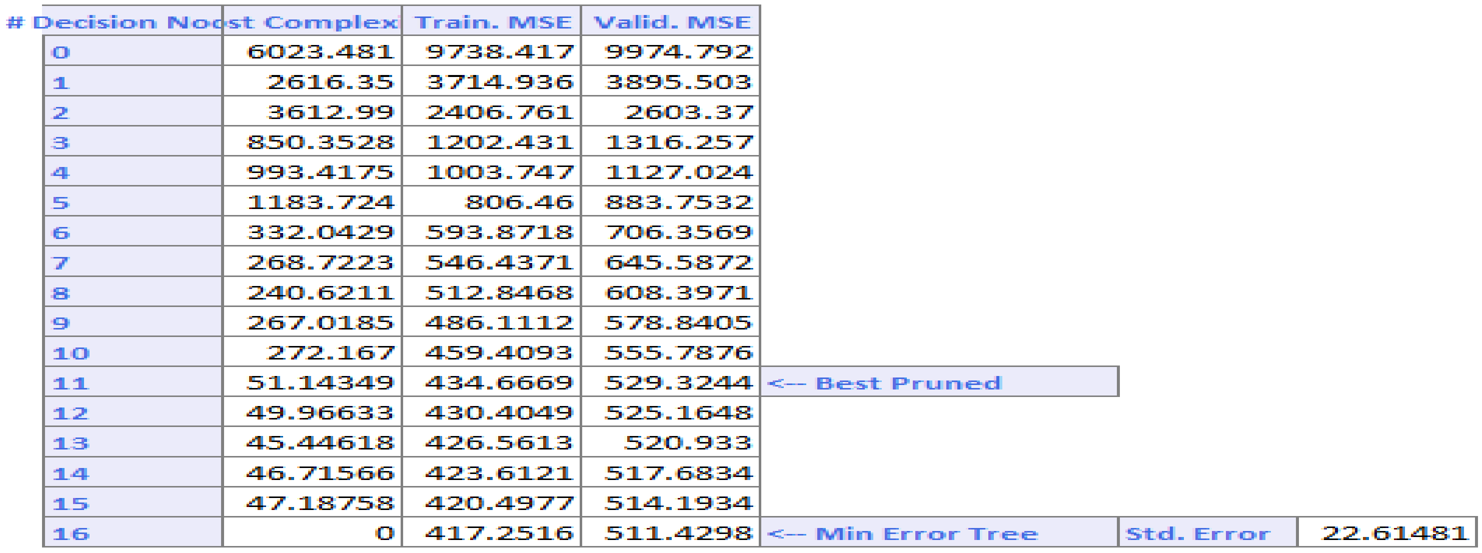
- **RROC** (regression receiver operating characteristic) curves plot the performance of repressors by graphing over-estimations (predicted values that are too high) versus underestimations (predicted values that are too low.) The closer the curve is to the top-left corner of the graph (the smaller the area above the curve), the better the performance of the model.
  In an RROC curve, we can compare the performance of a regressor with that of a random guess (red line) for which over-estimations are equal to under-estimations. Anything to the left of this line signifies a better prediction, and anything to the right signifies a worse prediction. The best possible prediction performance would be denoted by a point at the top-left of the graph at the intersection of the x and y axis. This point is sometimes referred to as the perfect classification. As for the Area over the Curve (AOC), the smaller the AOC, the better the performance of the model.

CART Model:

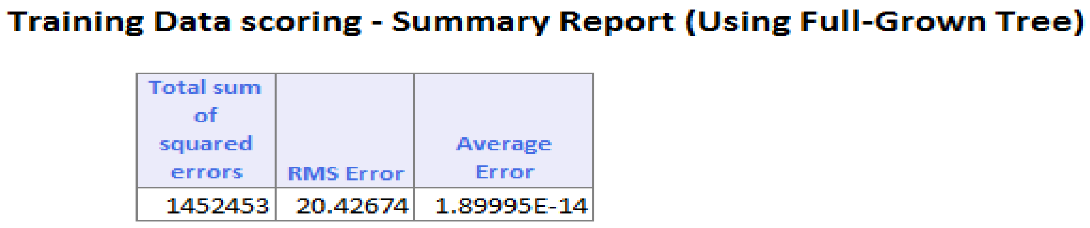
The CART Model was developed using the full grown tree option to display the full regression tree. The tree was drawn according to the maximum number of levels in the tree that are specified. Here the TAX variable is chosen as the first splitting variable. If the TAX is >= 5504.50 (886 cases) then TAX is split again. If the TAX variable is < 4347.00 (2595 cases), then TAX variable is split again and so on. We notice that TAX variable is predominantly used as the splitting variable for every step. This could be because TAX variable is statistically significant in predicting the total value of the house and is in a linear relationship with TOTAL_VALUE variable.



The Prune log shows the root-mean-square error (RMSE) at each stage of the tree for both the Training and Validation Sets. The prune log shows that the training RMSE continues reducing as the tree continues to split. XLMiner chooses the number of decision nodes for the pruned tree and the minimum error tree from the values of RMSE. In the Prune Log, **the smallest Validation RMSE error belongs to the tree with 16 decision** nodes. The **Minimum Error Tree is 511.4298** which is the tree with the smallest misclassification error in the Validation Set. The Best Pruned Tree is the tree with the largest error that is still less than the sum of the Standard Error and the Validation RMSE error. Here, **the tree with 11 nodes is the Best Pruned Tree**.

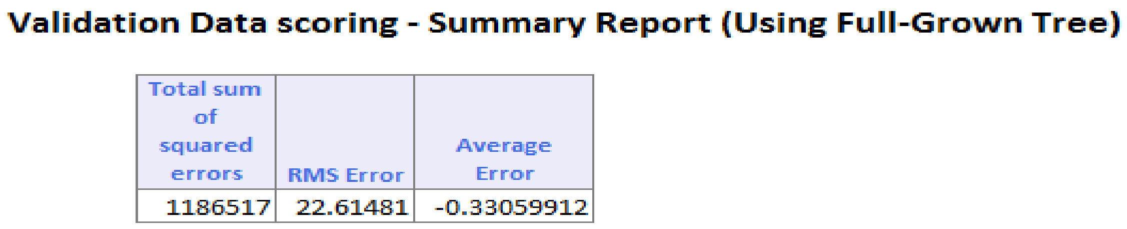


# Decision Noc	st Complex	Train. MSE	Valid. MSE			
0	6023.481	9738.417	9974.792			
1	2616.35	3714.936	3895.503			
2	3612.99	2406.761	2603.37			
3	850.3528	1202.431	1316.257			
4	993.4175	1003.747	1127.024			
5	1183.724	806.46	883.7532			
6	332.0429	593.8718	706.3569			
7	268.7223	546.4371	645.5872			
8	240.6211	512.8468	608.3971			
9	267.0185	486.1112	578.8405			
10	272.167	459.4093	555.7876			
11	51.14349	434.6669	529.3244	<-- Best Pruned		
12	49.96633	430.4049	525.1648			
13	45.44618	426.5613	520.933			
14	46.71566	423.6121	517.6834			
15	47.18758	420.4977	514.1934			
16	0	417.2516	511.4298	<-- Min Error Tree	Std. Error	22.61481



Training Data scoring - Summary Report (Using Full-Grown Tree)

Total sum of squared errors	RMS Error	Average Error
1452453	20.42674	1.89995E-14

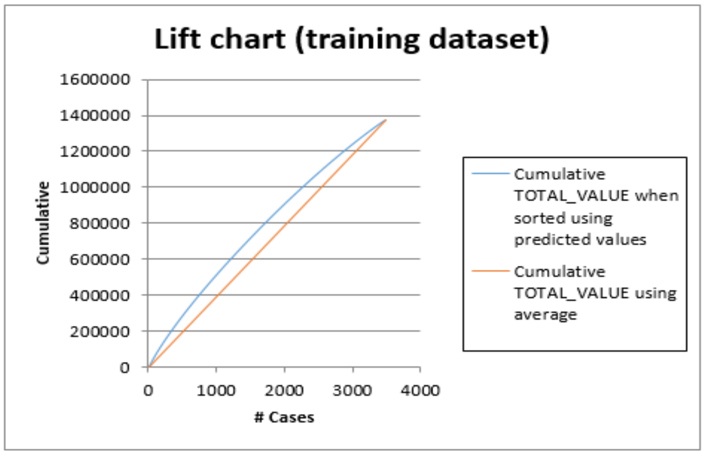


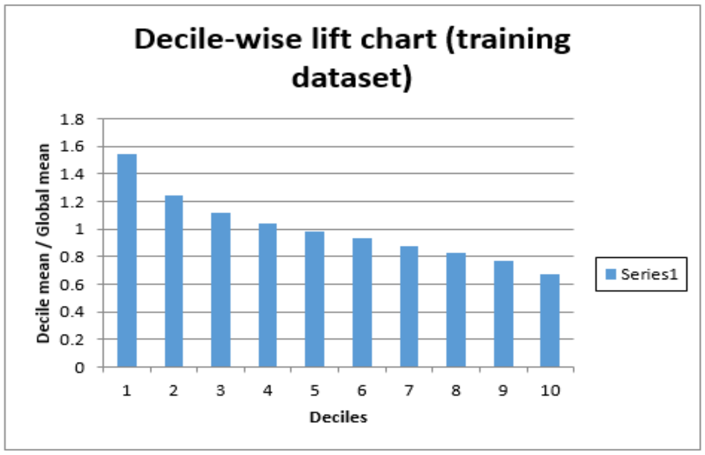
Validation Data scoring - Summary Report (Using Full-Grown Tree)

Total sum of squared errors	RMS Error	Average Error
1186517	22.61481	-0.33059912

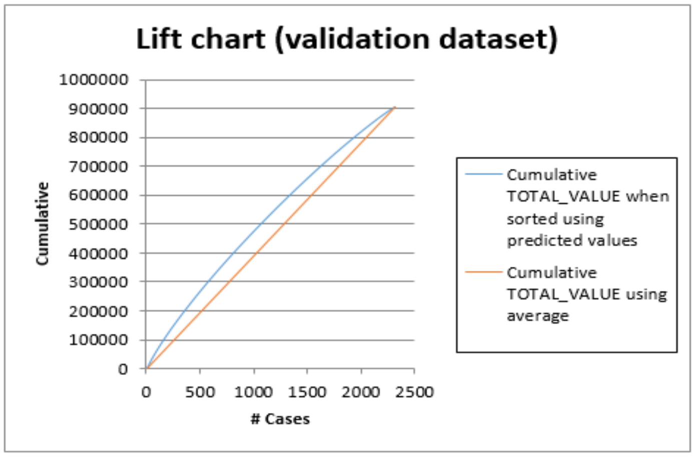
The above figure shows the RMS Error value of both the training and the validation dataset. From the above tabular columns we infer that the RMSE for training data is 20.42674 and 22.61481 for validation data. This suggests that the model was able to learn from the training dataset to predict the total value of the houses in the validation data with an average error of -0.33059912.

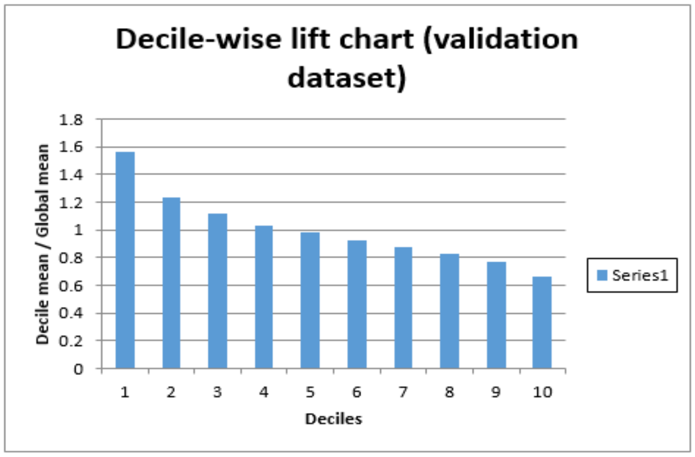
Training dataset:



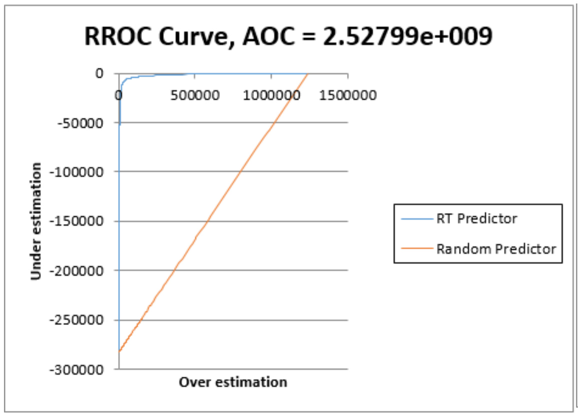


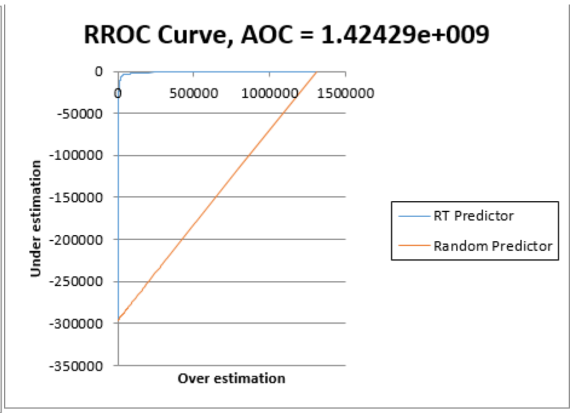
Validation dataset:





Training Dataset and Validation Dataset:



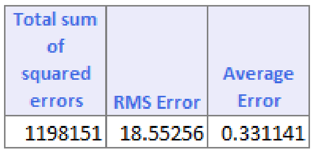


- The **Lift charts** are a measure of effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model. Bigger the area between the base line and lift curve, better is the model. Both the training and validation data have the same area between the base line and lift curve.
- The **decile-wise lift curve** shows the factor by which the CART model outperforms a random assignment, one decile at a time. In the first decile, taking the most expensive predicted total value of the houses in the dataset, the predictive performance of the model is almost 1.6 times better as simply assigning a random predicted value.
- **RROC** curves plot the performance of regressors by graphing over-estimations (predicted values that are too high) versus underestimations (predicted values that are too low). Here the RROC curve is closest to the top-left corner of the graph indicating that the smaller the area above the curve the better the performance of the model. Therefore the Area over the Curve (AOC) is very small, also suggesting that the performance of the model is good.

Random Forest:

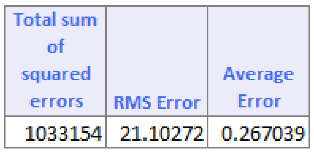
Random forests is a notion of the general technique of random decision forests that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Training Data scoring - Summary Report



Total sum of squared errors	RMS Error	Average Error
1198151	18.55256	0.331141

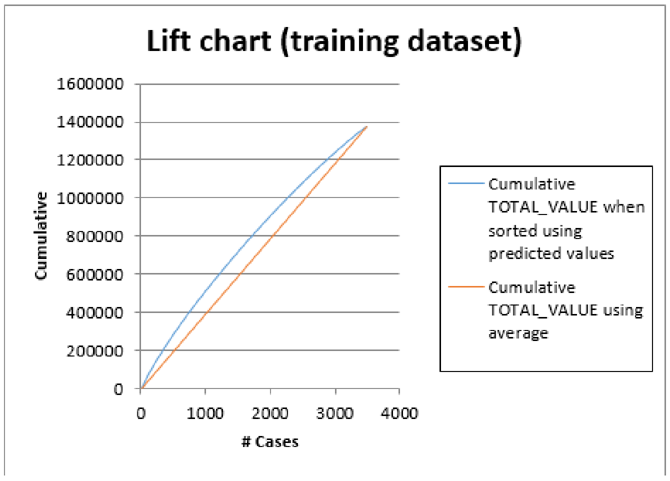
Validation Data scoring - Summary Report

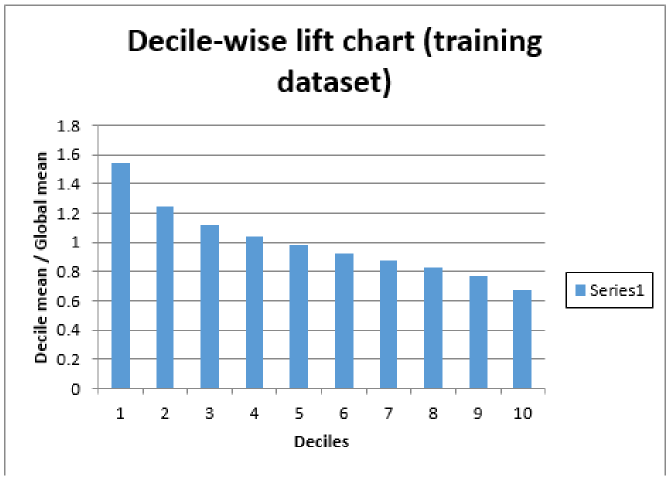


Total sum of squared errors	RMS Error	Average Error
1033154	21.10272	0.267039

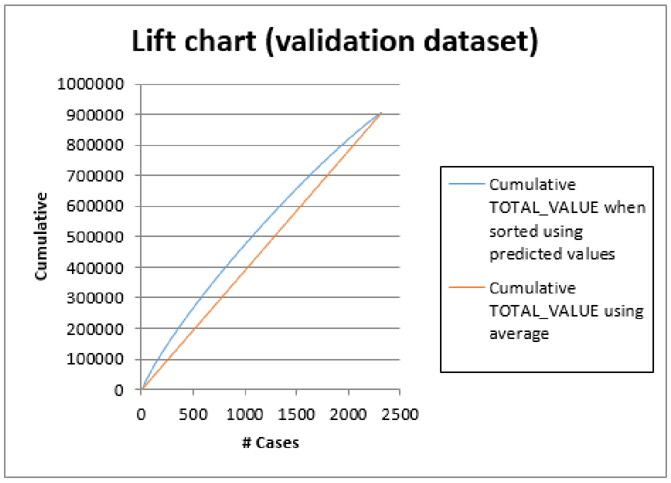
The above figure shows the RMS Error value of both the training and the validation dataset. From the above tabular columns we infer that the RMSE for training data is 18.55256 and 21.10272 for validation data. This suggests that the model was able to learn from the training dataset to predict the total value of the houses in the validation data with an average error of 0.267039.

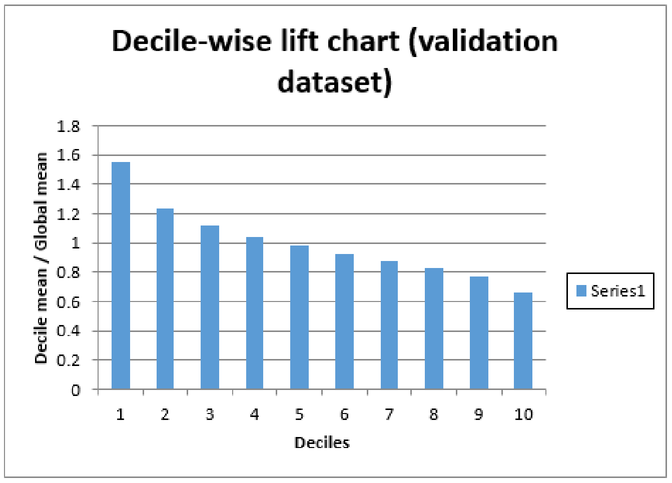
Training Dataset:



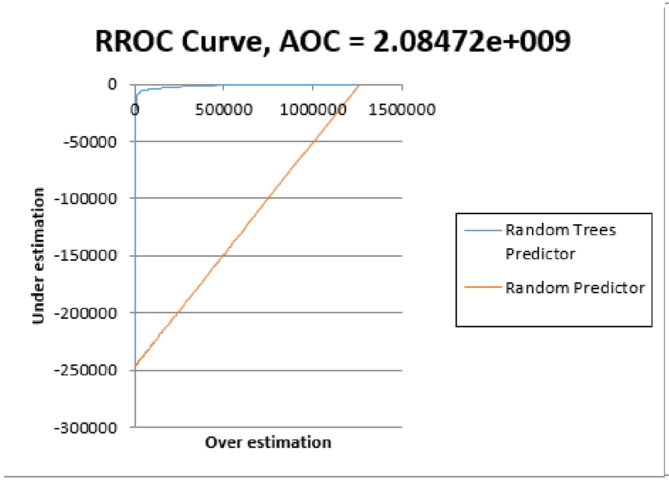


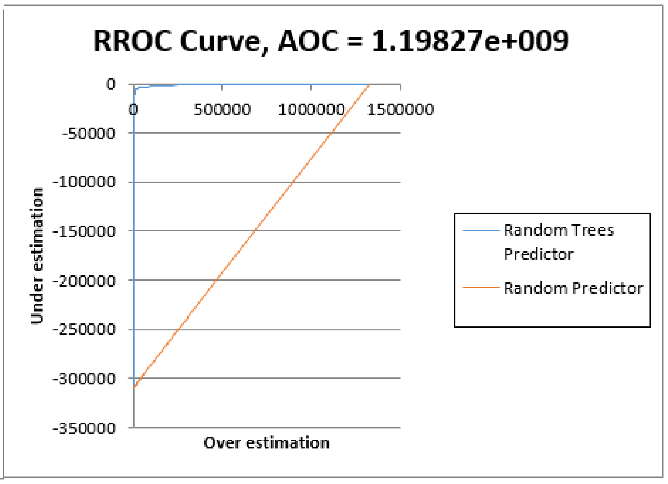
Validation Dataset:





Training and Validation dataset:





- The **Lift charts** are a measure of effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model. Bigger the area between the base line and lift curve, better is the model. Both the training and validation data have the same area between the base line and lift curve.
- The **decile-wise lift curve** shows the factor by which the RANDOM FOREST model outperforms a random assignment, one decile at a time. In the first decile, taking the most expensive predicted total value of the houses in the dataset, the predictive performance of the model is almost 1.6 times better as simply assigning a random predicted value.
- **RROC** curves plot the performance of regressors by graphing over-estimations (predicted values that are too high) versus underestimations (predicted values that are too low). Here the RROC curve is extremely close to the top-left corner of the graph indicating that the smaller the area above the curve the better the performance of the model. Therefore the Area over the Curve (AOC) is very negligible, also suggesting that the performance of the model is good.

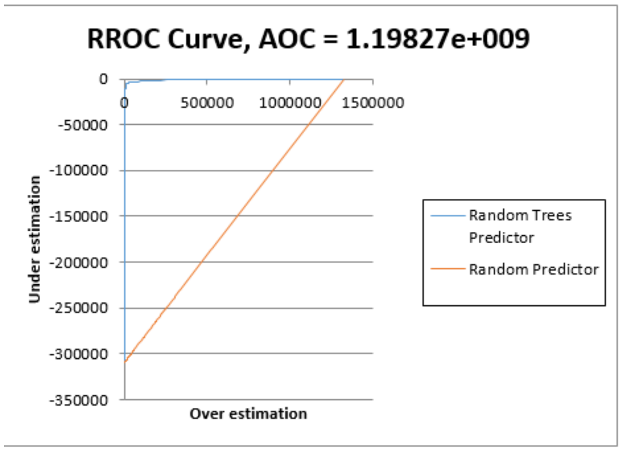
Final Comparison:

Random Forest:

AOC= 1.19827e+009

RMSE= 21.102

RROC curve

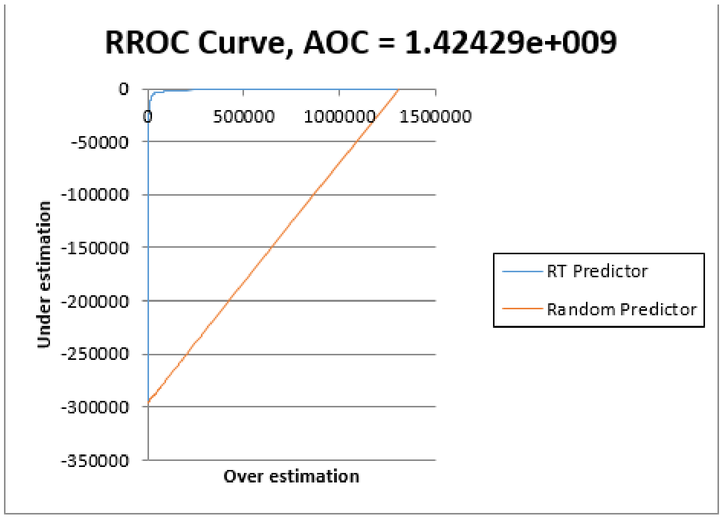


CART:

AOC = 1.42429e+009

RMSE= 22.614

RROC curve

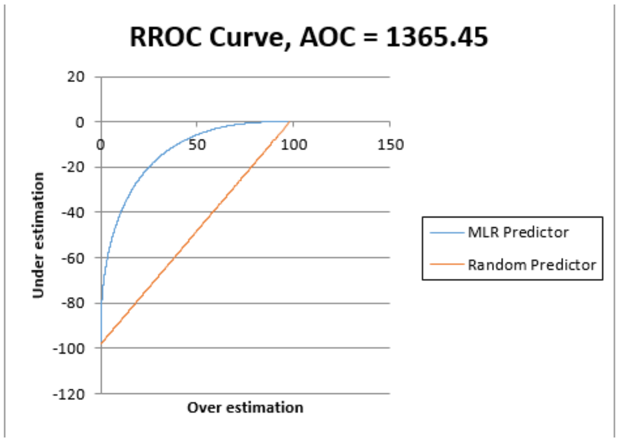


Multiple Linear Regression:

AOC = 1365.45

RMSE = 0.02254

RROC curve



Conclusion:

From the below observations we recommend that **Random Forest to be the best model for prediction** of the total value of the houses in the West Roxbury Dataset with lower AOC and RMSE values.

Multiple Regression Model is not chosen as the best because the problem of overfitting could have occurred and its Area over curve value is significantly higher compared to CART and Random Forest.

Problem 2

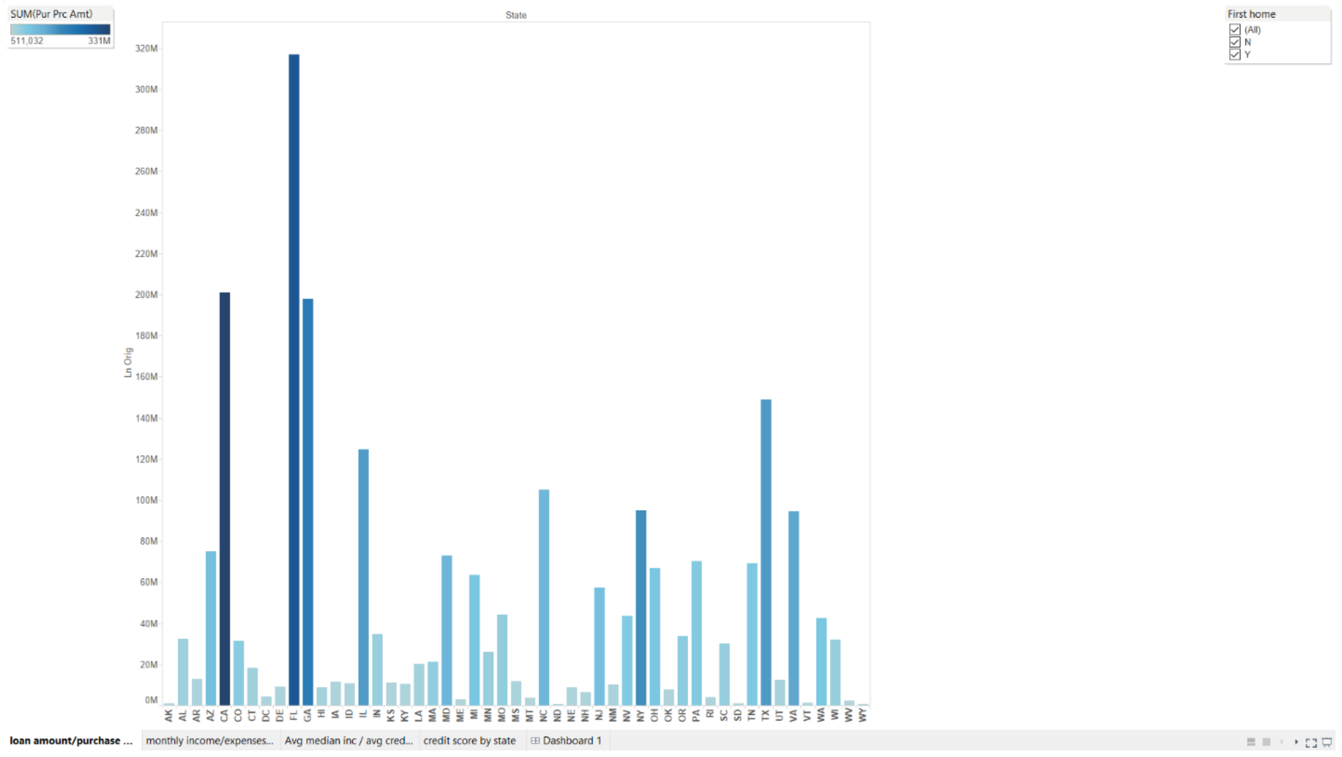
Mortgage Defaults:

Given dataset is contain mortgage characteristics like loan amount, loan status, monthly income etc. It also shows loan is defaulted or not. The dataset have 15154 records of different people who taken loan for buying the house. There are different viable which decides whether they fall into defaulter or non-defaulter category.

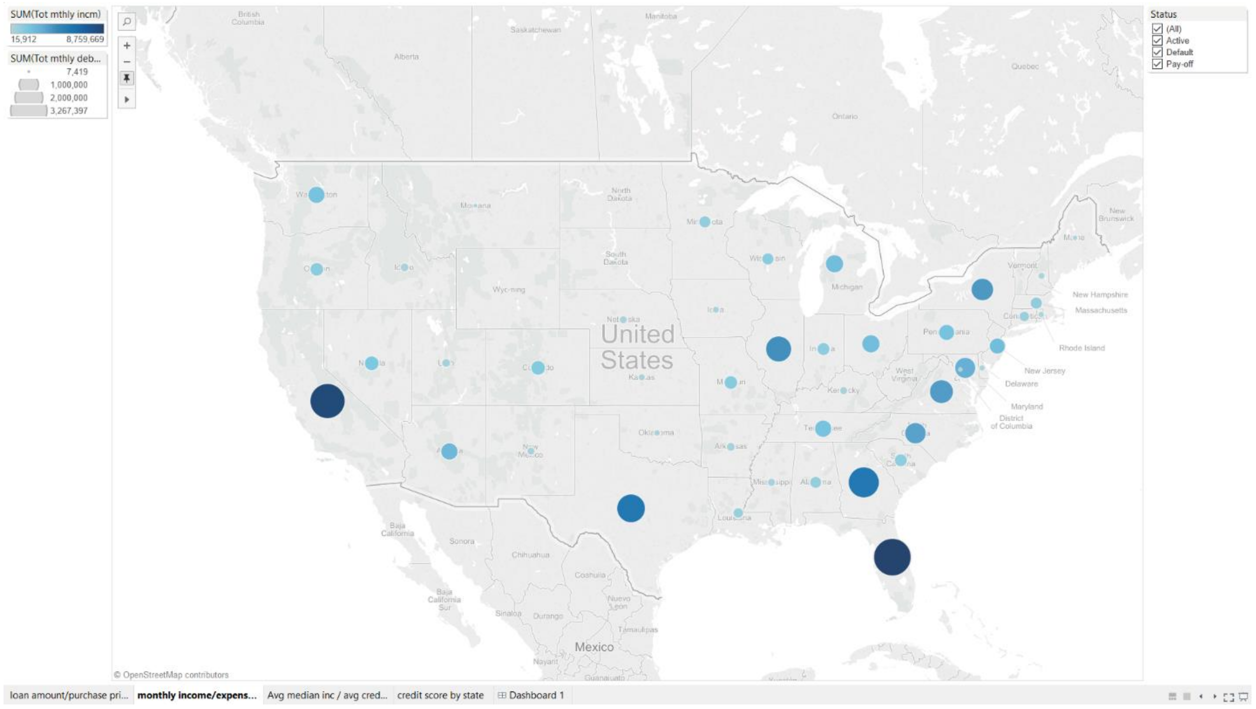
Exploratory data analysis:

1) **Loan amount & purchase amount by state**

This bar graph shows value of loan and sum purchase amount of home by different state. From this graph we can clearly determine which state have highest loan amount and purchase amount. In this graph Florida State approved largest loan amount and also have highest purchase amount of home. There is also filter of whether it is their first home or not. By this you can analyze amount of loan in particular state approved for first home buyers.

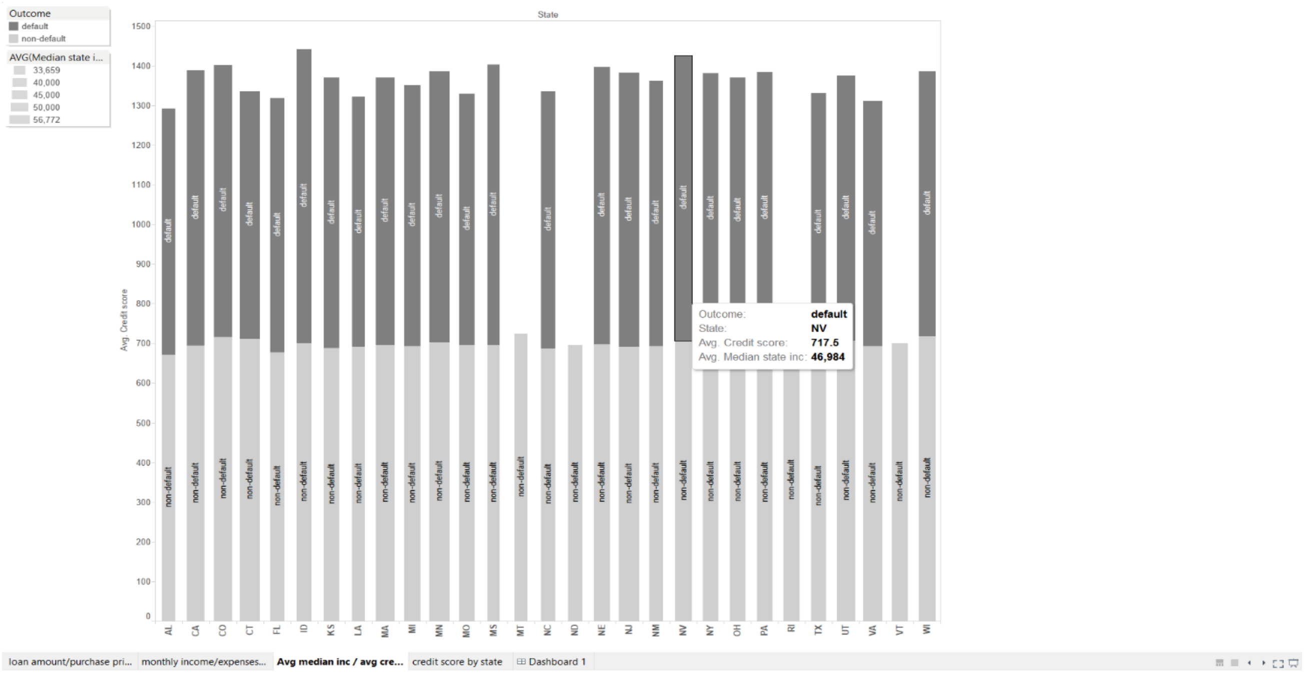

SUM(Pur Prc Amt)
511,032
331M
State
Ln Orig
First home
(All)
N
Y
loan amount/purchase ...
monthly income/expenses...
Avg median inc / avg cred...
credit score by state
Dashboard 1

2) Monthly income & Monthly expenses by states



From this Geo map we can determine total monthly income & expenses of people who have taken loan. From map we can clearly determine that people in Florida have maximum monthly income & also have maximum expenses. There is also filter of status by which can filter data. You can filter data by active, pay-off, default.

3) Avg. credit score and Avg. median state income by state



Outcome
▇ default
▢ non-default

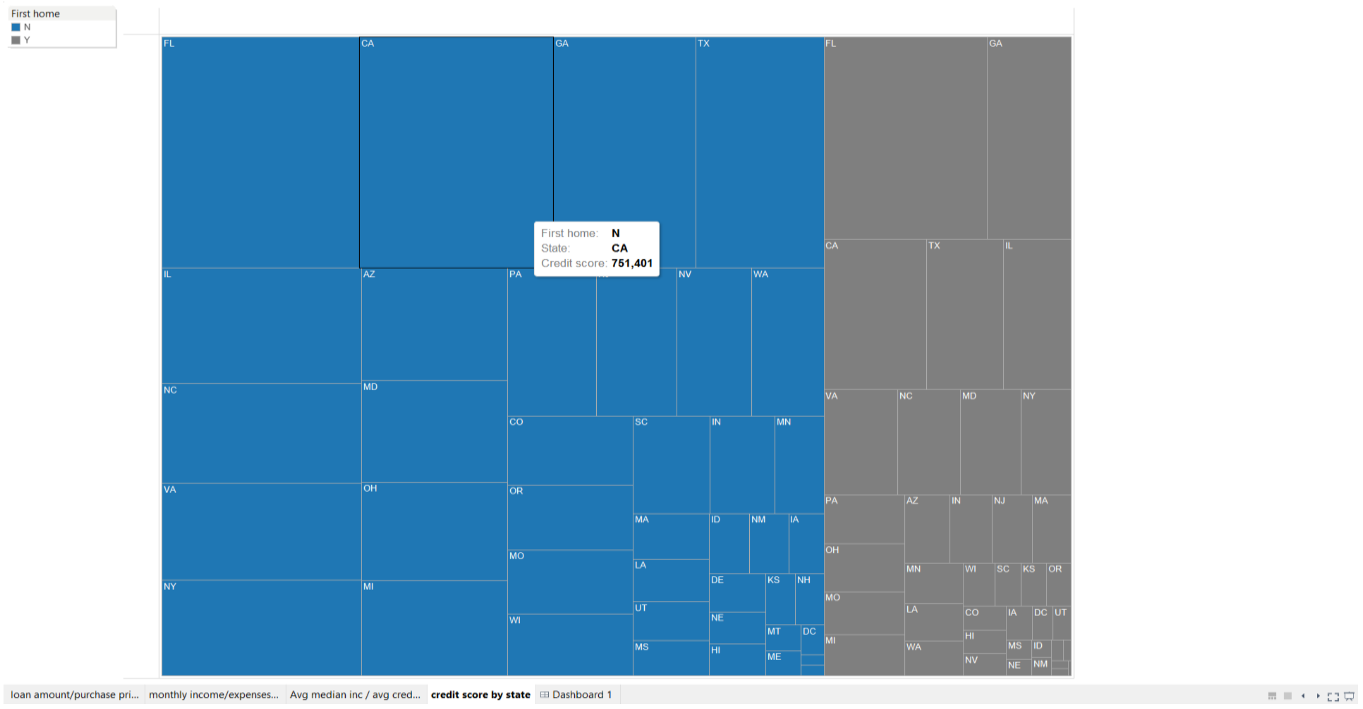
AVG(Median state i...
33,659
40,000
45,000
50,000
56,772

Outcome:	default
State:	NV
Avg. Credit score:	717.5
Avg. Median state inc:	46,984

In this graph we can analyze average credit score and median state income by state. We can also determine average income and credit score for defaulter and non-defaulter.

4) Credit scored by state

This heat map shows credit card scores of people in different states. Also classify heat map by whether people buying their first home or not. You can determine in which state people having highest credit score. So people in Florida having highest credit score in both (First home Y/N).



Logistic Regression Model:

- In this model we partition data as **Training data=60% and Test data= 40%**
- After partition we started to build classification model using logistic regression
- We take Ln_orig, credit_score, Tot_mthly_debt_exp, Tot_mthly_incm, orig_apprd_val_amt, pur_prc_amt as **input variable** and OUTCOME as **output variable**

Confusion Matrix:

A Confusion Matrix is used to evaluate the performance of a classification method. This matrix summarizes the records that were classified correctly and those that were not. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class.

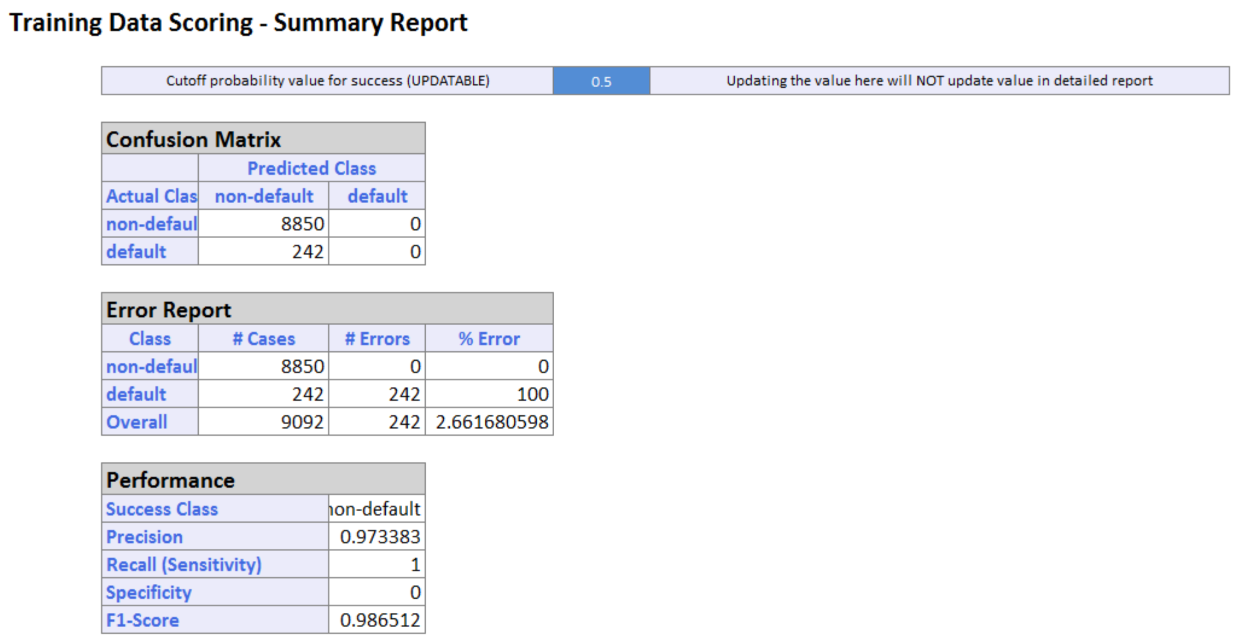
Error Report:

Error Report gives overall errors in our model.

Performance Table:

Performance matrix give us total overview of performance of algorithm depending upon several attributes. Success class specify which class is successful in algorithm. Precision gives us overall success rate of model. Sensitivity measures the proportion of actual positives that are correctly identified as positive. Specificity measures proportion of negatives that are correctly identified as negative

Training data scoring



Training Data Scoring - Summary Report

Cutoff probability value for success (UPDATABLE)	0.5	Updating the value here will NOT update value in detailed report

Confusion Matrix		
	Predicted Class	
Actual Clas	non-default	default
non-defaul	8850	0
default	242	0

Error Report			
Class	# Cases	# Errors	% Error
non-defaul	8850	0	0
default	242	242	100
Overall	9092	242	2.661680598

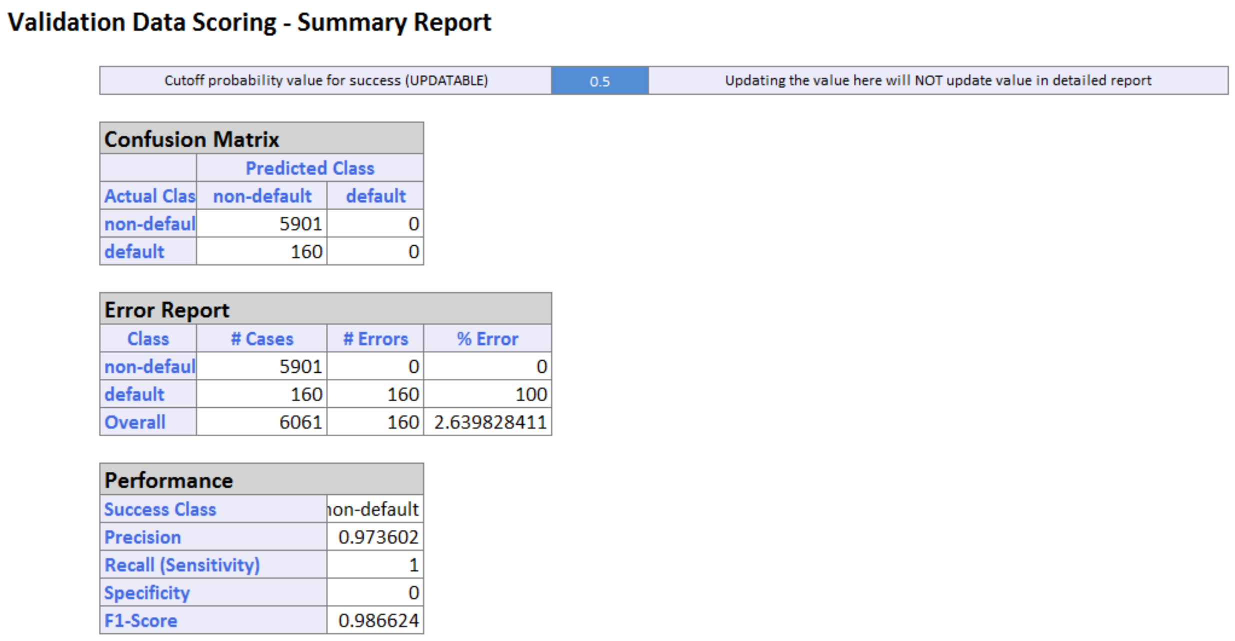
Performance	
Success Class	non-default
Precision	0.973383
Recall (Sensitivity)	1
Specificity	0
F1-Score	0.986512

From the report we can see that there are total 9092 records in training dataset. **In confusion matrix** we can see that there are 8850 records are assigned as non-default which are really belongs to class non-default, also 242 records assigned as non-default which are actually belongs to default class

Error report shows that there are 0% errors non default class and 100% error in default class. Overall error rate in this model for training dataset is 2.66%

Performance table shows that success class is non-default, Precision value is 0.9733, and sensitivity is also 1, Specificity is 0

Validation data scoring



Validation Data Scoring - Summary Report

Cutoff probability value for success (UPDATABLE)	0.5	Updating the value here will NOT update value in detailed report

Confusion Matrix		
	Predicted Class	
Actual Clas	non-default	default
non-defaul	5901	0
default	160	0

Error Report			
Class	# Cases	# Errors	% Error
non-defaul	5901	0	0
default	160	160	100
Overall	6061	160	2.639828411

Performance	
Success Class	on-default
Precision	0.973602
Recall (Sensitivity)	1
Specificity	0
F1-Score	0.986624

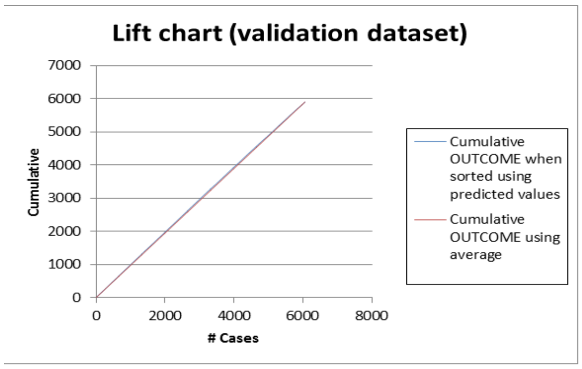
From the report we can see that there are total 6601 records in validation dataset. **In confusion matrix** we can see that there are 5901 records are assigned as non-default which are actually non-default. Also 160 records assigned as non-default, all default class value assigned as non-default class value.

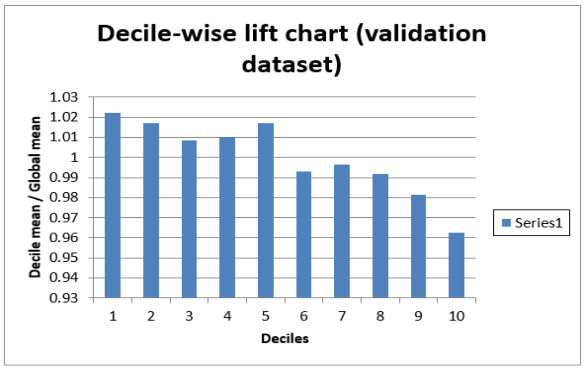
Error report shows that there are 0% errors non default class and 100% error in default class. Overall error rate in this model for validation dataset is 2.63%

Performance table shows that success class is non-default, Precision value is 0.9736, and sensitivity is also 1, Specificity is 0

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. A graphical way to assess predictive performance is through a lift chart which compares performance to a base line model that has no predictors. Lift chart gives the highest cumulative predicted values when we are searching for a set of records that are relevant. Lift chart consist of lift curve and baseline. It said that, greater the distance between baseline and lift curve better the model.

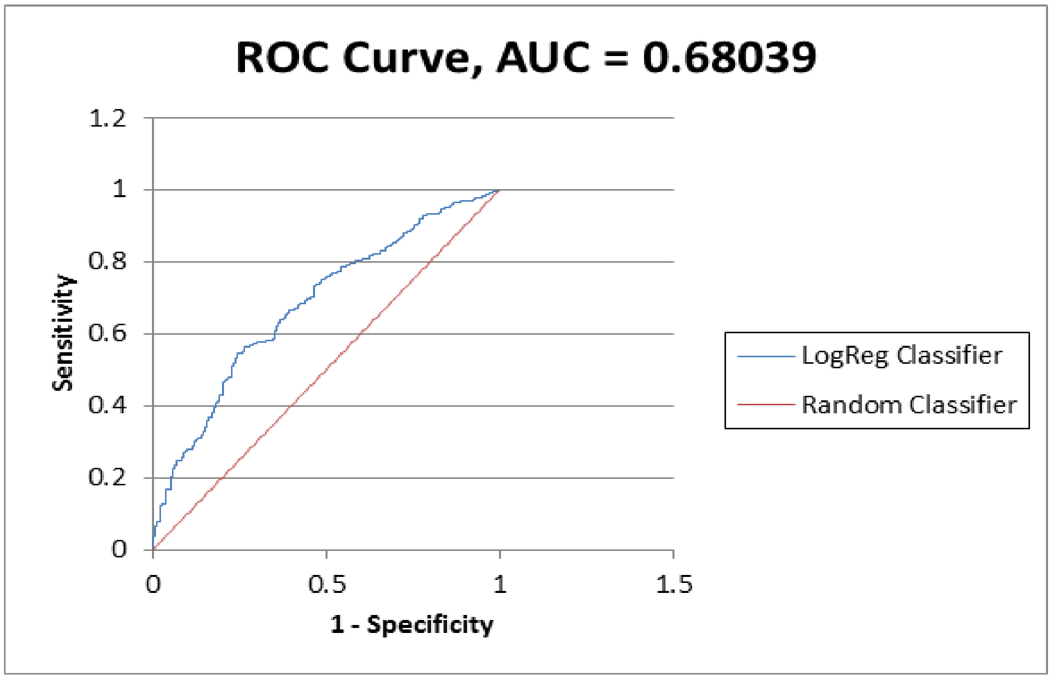




In above Lift chart baseline and lift curve almost overlapping each other.

ROC Curve:

The ROC curve plots the pair's sensitivity and 1- specificity as cut-off value increase from 0 and 1. Model reflects better performance if curve is closer to top left corner. Sensitivity measures the proportion of actual positives that are correctly identified as positive. Specificity measures proportion of negatives that are correctly identified as negative



Cart:

- In this model we partition data as **Training data=60% and Test data= 40%**
- After partition we started to build classification model using logistic regression
- We take Ln_orig, credit_score, Tot_mthly_debt_exp, Tot_mthly_incm, orig_apprd_val_amt, pur_prc_amt as **input variable** and OUTCOME as **output variable**

Confusion Matrix:

A Confusion Matrix is used to evaluate the performance of a classification method. This matrix summarizes the records that were classified correctly and those that were not. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class.

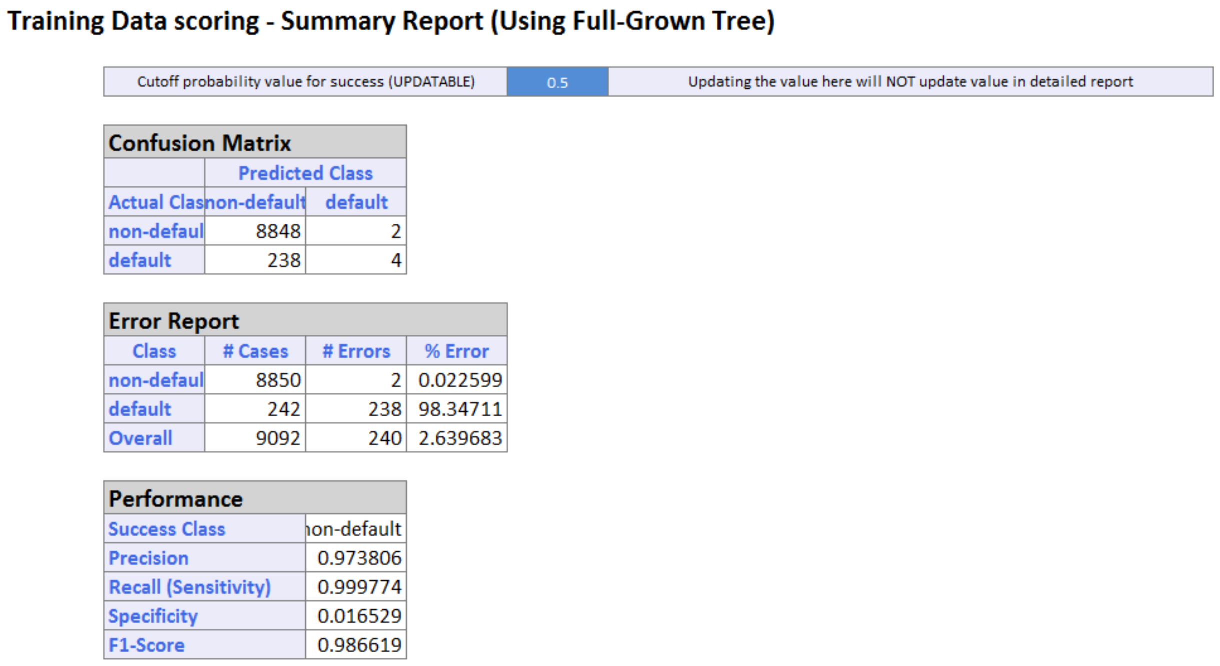
Error Report:

Error Report gives overall errors in our model.

Performance Table:

Performance matrix give us total overview of performance of algorithm depending upon several attributes. Success class specify which class is successful in algorithm. Precision gives us overall success rate of model. Sensitivity measures the proportion of actual positives that are correctly identified as positive. Specificity measures proportion of negatives that are correctly identified as negative

Training data scoring



Training Data scoring - Summary Report (Using Full-Grown Tree)

Cutoff probability value for success (UPDATABLE)	0.5	Updating the value here will NOT update value in detailed report

Confusion Matrix		
	Predicted Class	
Actual Clas	non-default	default
non-defaul	8848	2
default	238	4

Error Report			
Class	# Cases	# Errors	% Error
non-defaul	8850	2	0.022599
default	242	238	98.34711
Overall	9092	240	2.639683

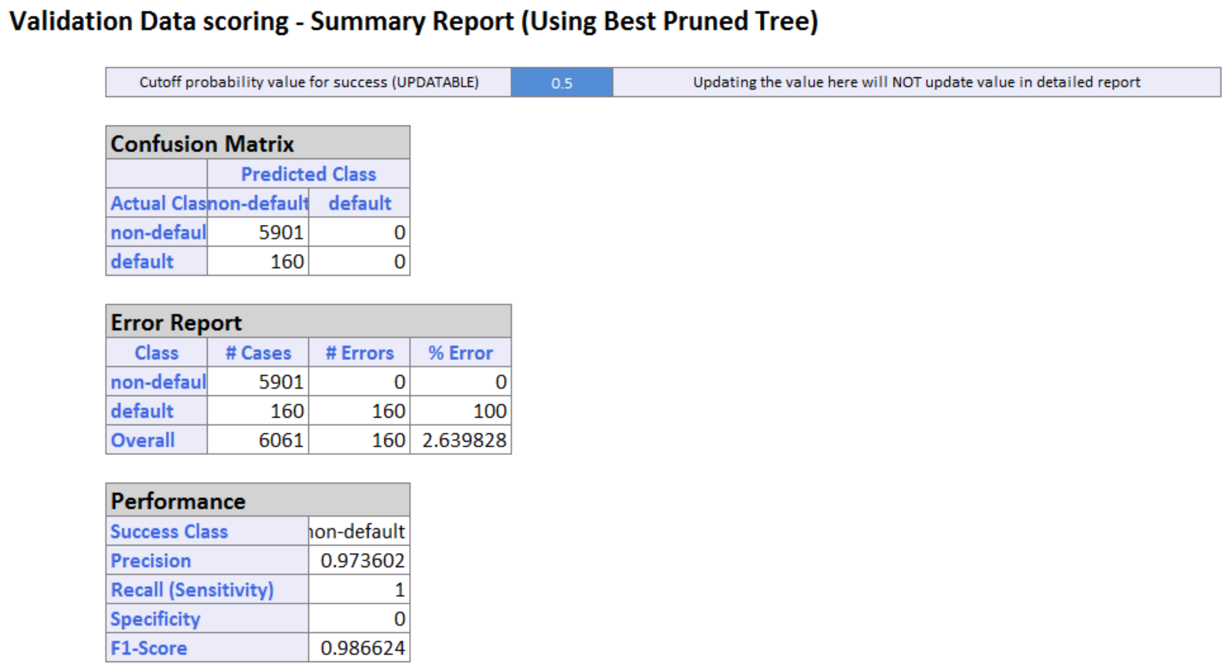
Performance	
Success Class	non-default
Precision	0.973806
Recall (Sensitivity)	0.999774
Specificity	0.016529
F1-Score	0.986619

From the report we can see that there are total 9092 records in training dataset. **In confusion matrix** we can see that there are 8848 records are assigned as non-default which are really belongs to class non-default and 2 non default class value assigned as default. Also 238 records assigned as non-default which are actually belongs to default class, but 4 default class value assigned correctly to default class.

Error report shows that there are 0.225% errors non default class and 98.347% error in default class. Overall error rate in this model for training dataset is 2.639%

Performance table shows that success class is non-default, Precision value is 0.9738, and sensitivity is also 0.999, Specificity is 0.01

Validation data scoring



Validation Data scoring - Summary Report (Using Best Pruned Tree)

Cutoff probability value for success (UPDATABLE)	0.5	Updating the value here will NOT update value in detailed report

Confusion Matrix

	Predicted Class	
Actual Clas	non-default	default
non-defaul	5901	0
default	160	0

Error Report

Class	# Cases	# Errors	% Error
non-defaul	5901	0	0
default	160	160	100
Overall	6061	160	2.639828

Performance

Success Class	ıon-default
Precision	0.973602
Recall (Sensitivity)	1
Specificity	0
F1-Score	0.986624

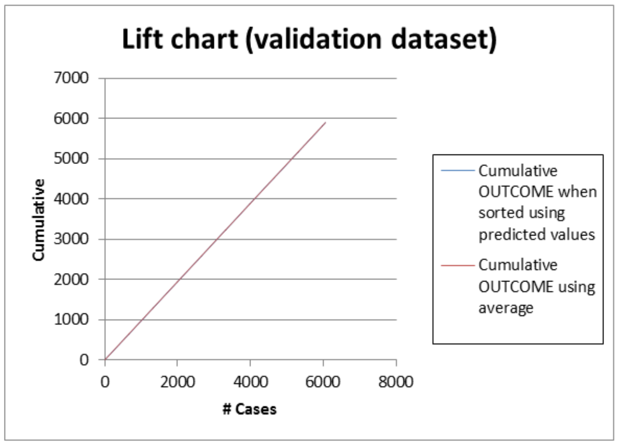
From the report we can see that there are total 6601 records in validation dataset. **In confusion matrix** we can see that there are 5901 records are assigned as non-default which are actually non-default. Also 160 records assigned as non-default, all default class value assigned as non-default class value.

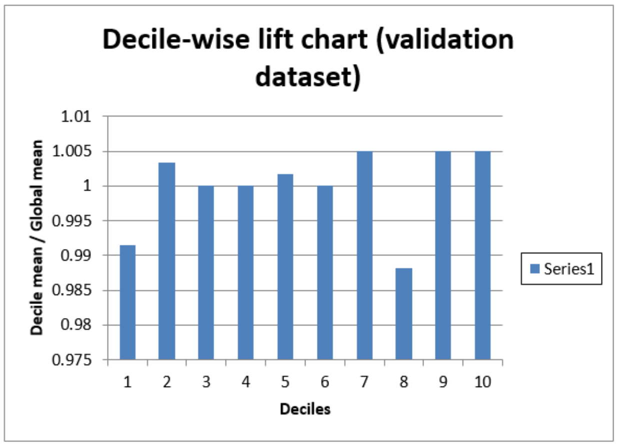
Error report shows that there are 0% errors non default class and 100% error in default class. Overall error rate in this model for validation dataset is 2.63%

Performance table shows that success class is non-default, Precision value is 0.9736, and sensitivity is also 1, Specificity is 0

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. A graphical way to assess predictive performance is through a lift chart which compares performance to a base line model that has no predictors. Lift chart gives the highest cumulative predicted values when we are searching for a set of records that are relevant. Lift chart consist of lift curve and baseline. It said that, greater the distance between baseline and lift curve better the model

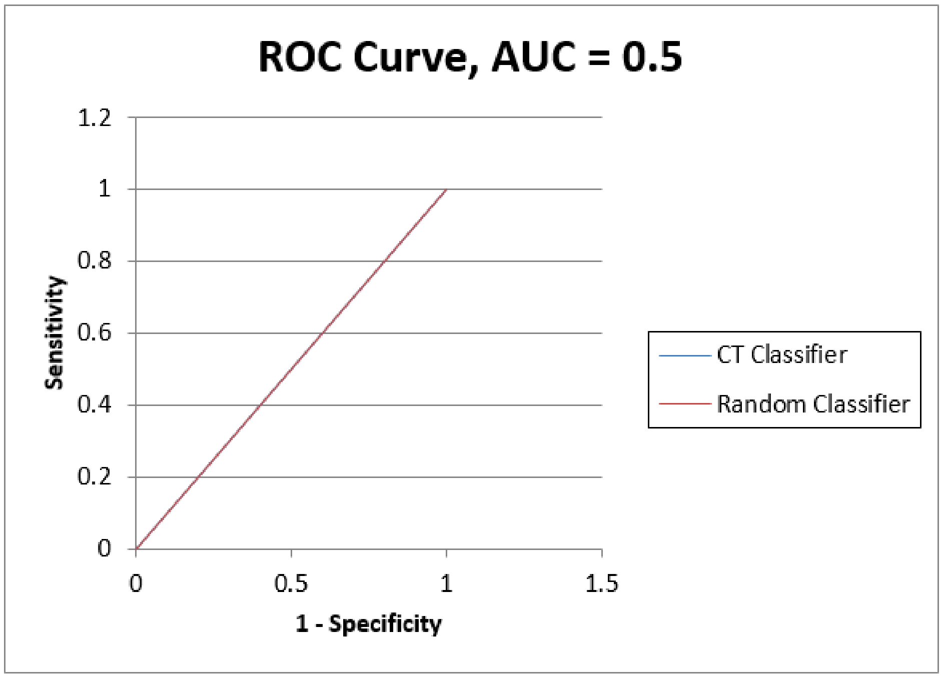




In above Lift chart baseline and lift curve almost overlapping each other.

ROC Curve:

The ROC curve plots the pair's sensitivity and 1- specificity as cut-off value increase from 0 and 1. Model reflects better performance if curve is closer to top left corner. Sensitivity measures the proportion of actual positives that are correctly identified as positive. Specificity measures proportion of negatives that are correctly identified as negative



Random Forest:

- In this model we partition data as **Training data=60% and Test data= 40%**
- After partition we started to build classification model using logistic regression
- We take Ln_orig, credit_score, Tot_mthly_debt_exp, Tot_mthly_incm, orig_apprd_val_amt, pur_prc_amt as **input variable** and OUTCOME as **output variable**

Confusion Matrix:

A Confusion Matrix is used to evaluate the performance of a classification method. This matrix summarizes the records that were classified correctly and those that were not. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class.

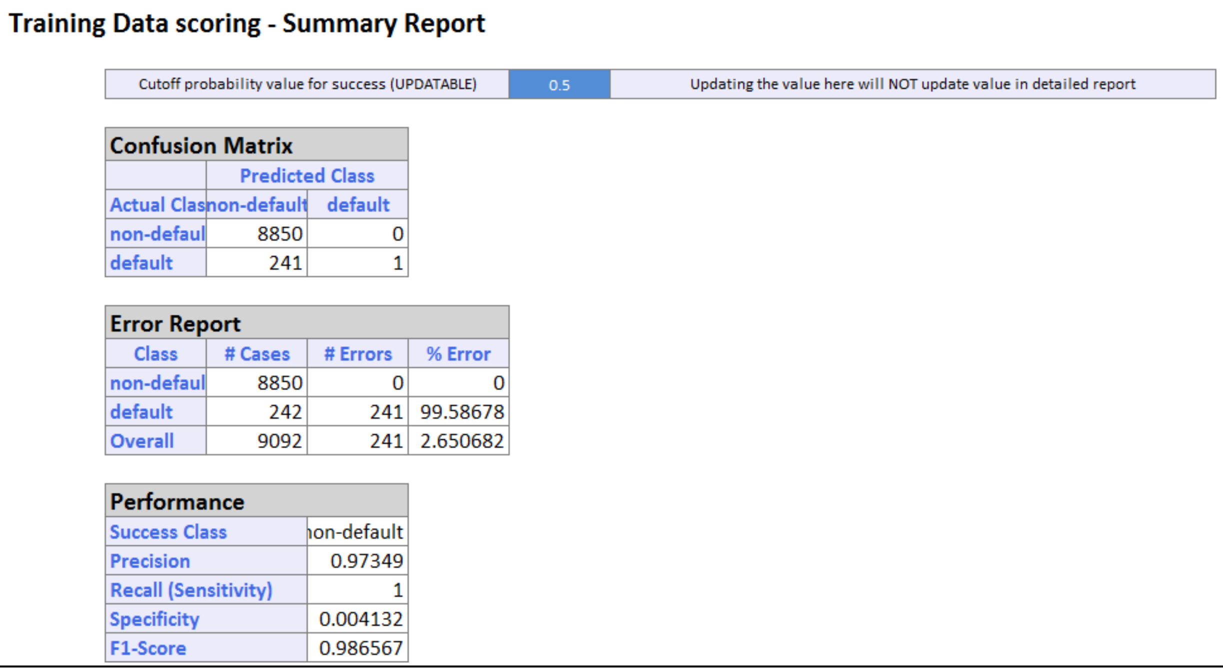
Error Report:

Error Report gives overall errors in our model.

Performance Table:

Performance matrix give us total overview of performance of algorithm depending upon several attributes. Success class specify which class is successful in algorithm. Precision gives us overall success rate of model. Sensitivity measures the proportion of actual positives that are correctly identified as positive. Specificity measures proportion of negatives that are correctly identified as negative

Training data scoring



Training Data scoring - Summary Report

Cutoff probability value for success (UPDATABLE)	0.5	Updating the value here will NOT update value in detailed report

Confusion Matrix		
	Predicted Class	
Actual Clas	non-default	default
non-defaul	8850	0
default	241	1

Error Report			
Class	# Cases	# Errors	% Error
non-defaul	8850	0	0
default	242	241	99.58678
Overall	9092	241	2.650682

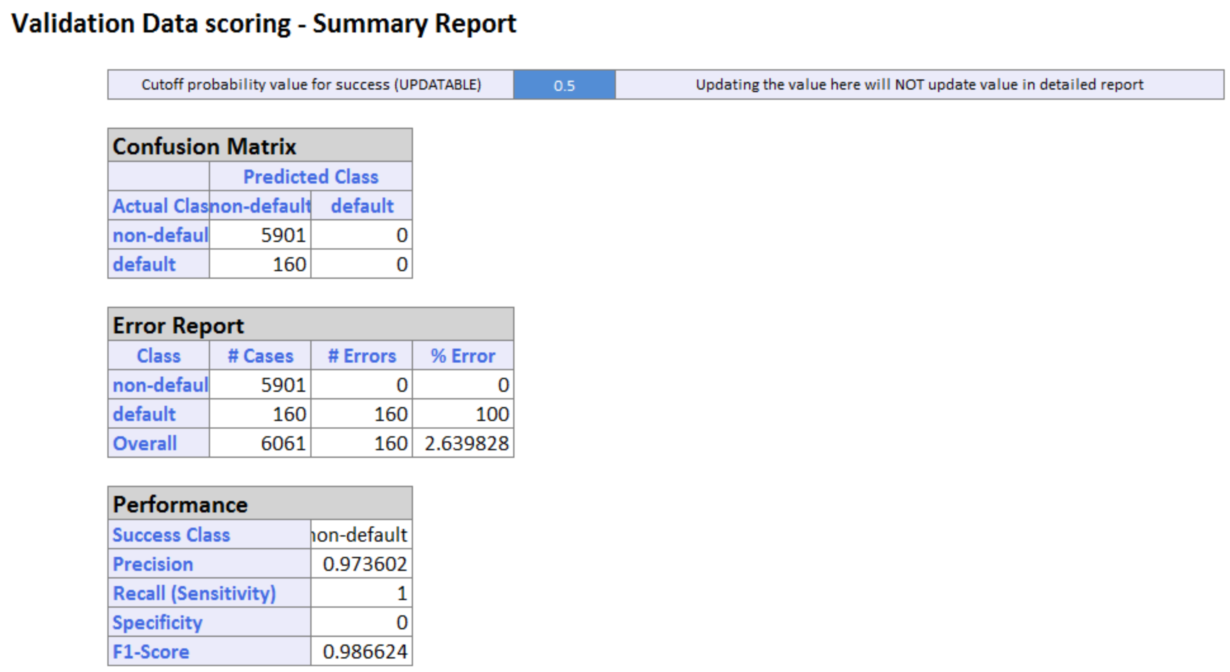
Performance	
Success Class	non-default
Precision	0.97349
Recall (Sensitivity)	1
Specificity	0.004132
F1-Score	0.986567

From the report we can see that there are total 9092 records in training dataset. **In confusion matrix** we can see that there are 8850 records are assigned to predicted non-default class which are really belongs to class non-default. Also 241 records assigned as non-default class which are actually belongs to default class and only one record from Actual default class assigned correctly predicted default class.

Error report shows that there are 0% errors in non-default class and 99.586% error in default class. Overall error rate in this model for training dataset is 2.650%

Performance table shows that success class is non-default, Precision value is 0.9734, and sensitivity is also 1, Specificity is 0.0041

Validation data scoring



Validation Data scoring - Summary Report

Cutoff probability value for success (UPDATABLE)	0.5	Updating the value here will NOT update value in detailed report

Confusion Matrix		
	Predicted Class	
Actual Clas	non-default	default
non-defaul	5901	0
default	160	0

Error Report			
Class	# Cases	# Errors	% Error
non-defaul	5901	0	0
default	160	160	100
Overall	6061	160	2.639828

Performance	
Success Class	non-default
Precision	0.973602
Recall (Sensitivity)	1
Specificity	0
F1-Score	0.986624

From the report we can see that there are total 6601 records in validation dataset. **In confusion matrix** we can see that there are 5901 records are assigned as non-default which are actually non-default. Also 160 records assigned as non-default, all default class value assigned as non-default class value.

Error report shows that there are 0% errors non default class and 100% error in default class. Overall error rate in this model for validation dataset is 2.63%

Performance table shows that success class is non-default, Precision value is 0.9736, and sensitivity is also 1, Specificity is 0

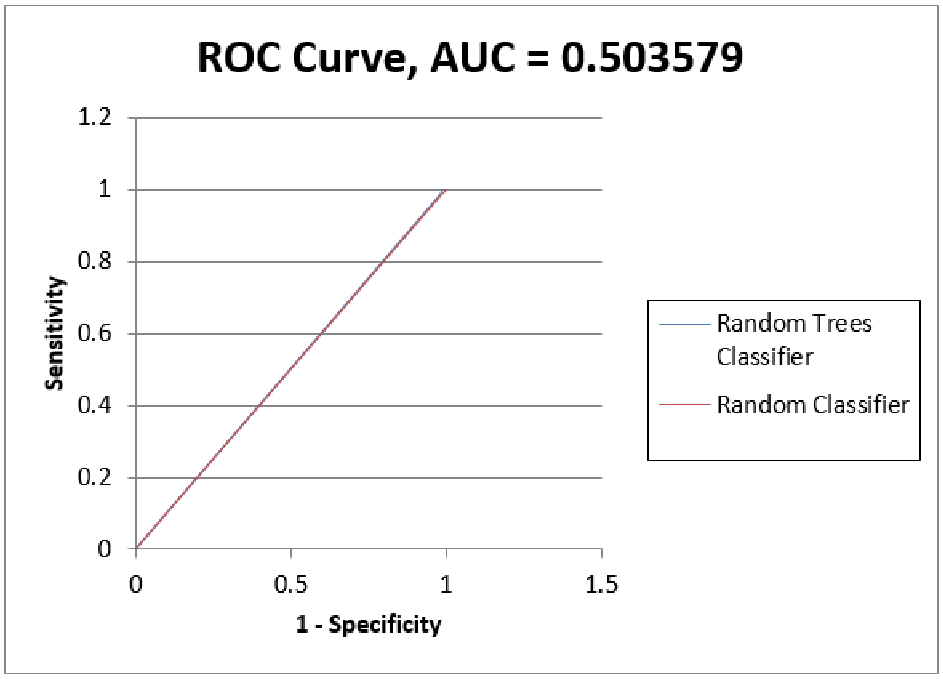
Lift Chart:

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. A graphical way to assess predictive performance is through a lift chart which compares performance to a base line model that has no predictors. Lift chart gives the highest cumulative predicted values when we are searching for a set of records that are relevant. Lift chart consist of lift curve and baseline. It said that, greater the distance between baseline and lift curve better the model

In above Lift chart baseline and lift curve almost overlapping each other.

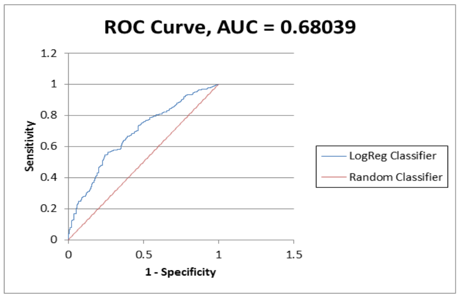
ROC Curve:

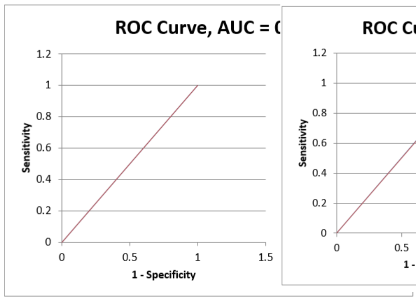
The ROC curve plots the pair's sensitivity and 1- specificity as cut-off value increase from 0 and 1. Model reflects better performance if curve is closer to top left corner. Sensitivity measures the proportion of actual positives that are correctly identified as positive. Specificity measures proportion of negatives that are correctly identified as negative

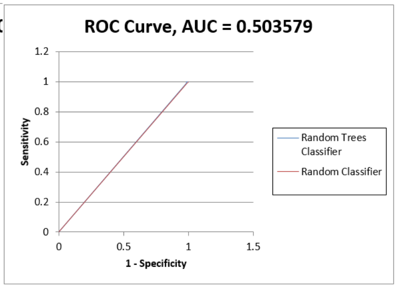


Model selection:

According to me for this dataset **logistic regression model** is best. If we compare ROC curve for three model.







Logistic Regression **Cart** **Rando**
m Forest

As you can see curve for logistic regression closer to upper left corner and AUC value for logistic regression is maximum. So I suggest logistic regression model for Mortgage Defaults dataset. If we compare error report for each model for validation data is almost same, so I use ROC curve for model selection.

Problem No 3: Detecting Spam (spamBase.xlxs):

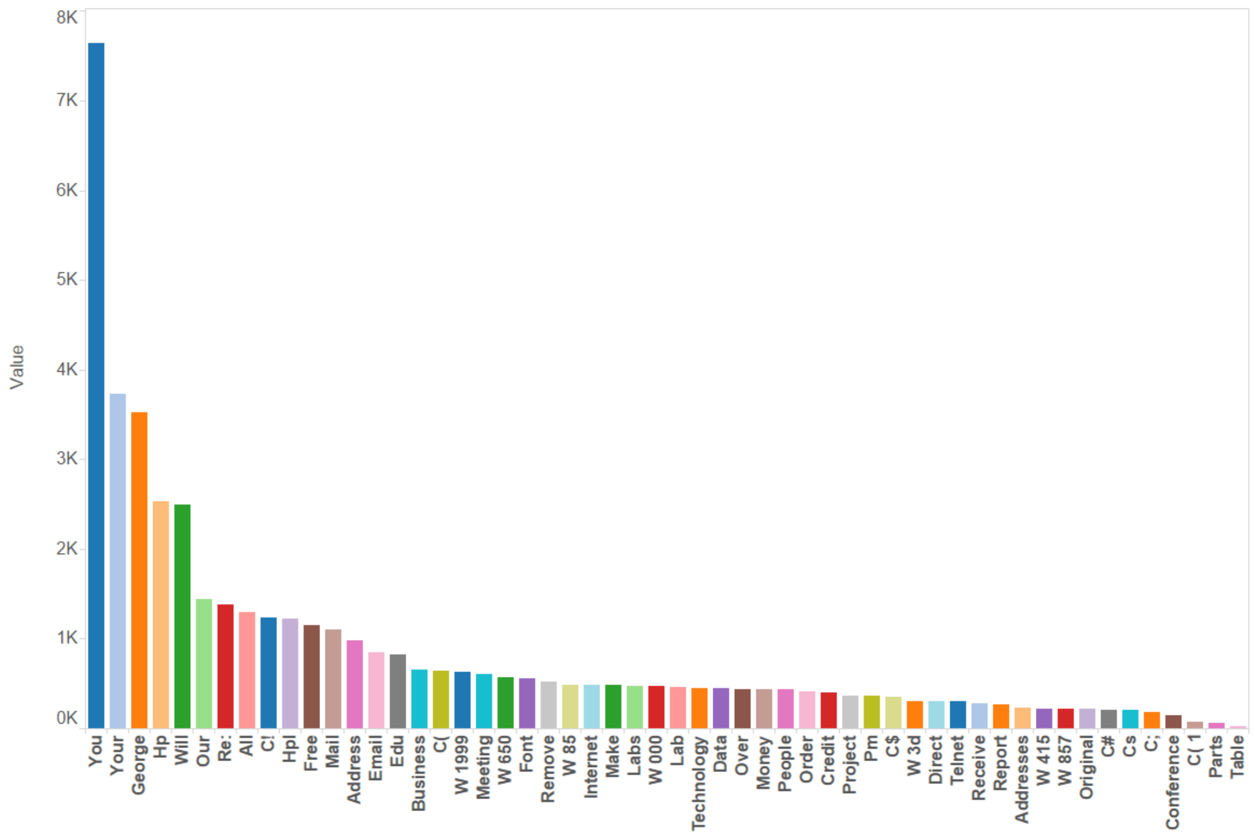
Objective: The main objective of the problem is to exploratory data analysis using TABLEAU and to perform logistic, CART and Random forest on the dataset and evaluate the most appropriate model which provides the most optimized solution for predictive analysis, whether the email is spam or not.

Nature of dataset: The dataset provided for the analysis contains 57 columns. The last columns tells us whether the email is a spam or not. If the value is 1 it's a spam email and if its 0 it's a non-spam email. Most of the attributes give us an insight whether a particular word or pattern or character is frequently occurring in the email.

Data Preprocessing: The dataset is always cleaned before the analysis. But there were no anomalies in the provided dataset so we didn't have to do any cleaning.

Exploratory Data Analysis using TABLEAU:

Figure 1: Figure 1 represents the frequency of occurances of different words in the email dataset. The X-axis contains all the words and the Y axis represents the number of occurences of each word.



Value
8K
7K
6K
5K
4K
3K
2K
1K
0K
You
Your
George
Hp
Will
Our
Re:
All
C!
Hpl
Free
Mail
Address
Email
Edu
Business
C(
W 1999
Meeting
W 650
Font
Remove
W 85
Internet
Make
Labs
W 000
Lab
Technology
Data
Over
Money
People
Order
Credit
Project
Pm
C$
W 3d
Direct
Telnet
Receive
Report
Addresses
W 415
W 857
Original
C#
Cs
C;
Conference
C(1
Parts
Table

Figure 1

Figure 2: Figure 2 represents the frequency of occurrence of different words in spam and non-spam emails. As you can see this graph has 2 parts. The 1st part contains the occurrence of different words in the non-spam emails and the 2nd part contains the occurrence of different words in spam emails.

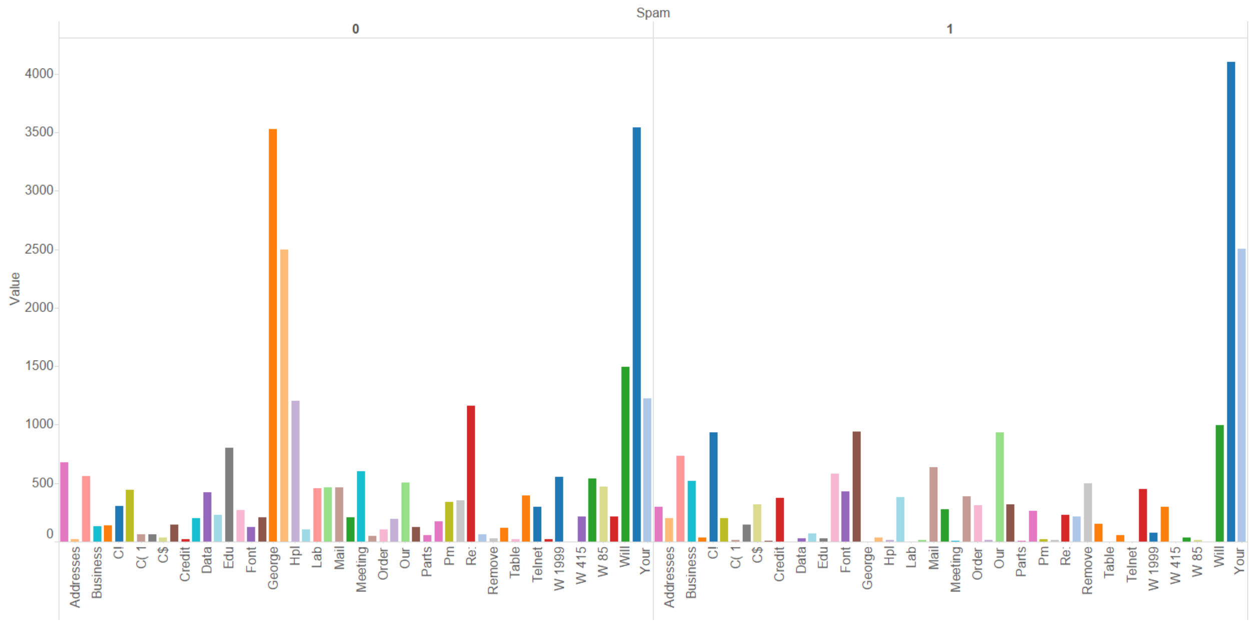
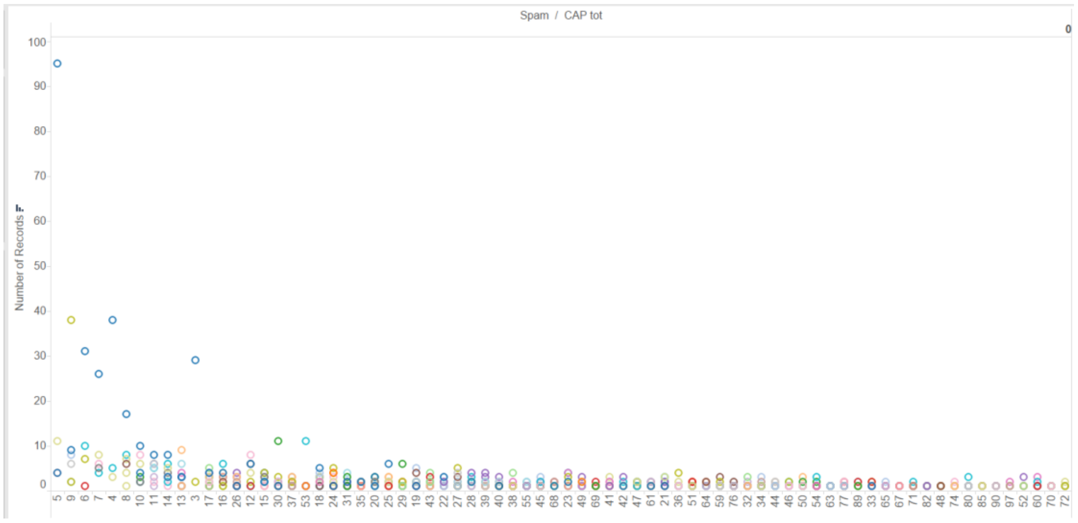


Figure 2

Figure 3: Figure 3 provides a pictorial representation of capAvg and CapTot for non-spam records with the details of total Number of records.



Analysis of Different Models:

Different metrics used for the evaluation of the best model are:

- **Confusion Matrix:** In the field of machine learning, a confusion matrix, also known as a contingency table or an error matrix [3] , is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice-versa).[2] The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).
- **Lift Chart:** The lift curve is a popular technique in direct marketing. One useful way to think of a lift curve is to consider a data mining model that attempts to identify the likely responders to a mailing by assigning each case a "probability of responding" score. The lift curve helps us determine how effectively we can "skim the cream" by selecting a relatively small number of cases and getting a relatively large portion of the responders. The input required to construct a lift curve is a validation dataset that has been "scored" by appending to each case the estimated probability that it will belong to a given class.

  For the given dataset the X axis represent the total number of input we get for analysis which is named as "case" where as in the Y axis represents the values of the number of spam emails depending on the value of X.
  The baseline (The curve in red) represents the average value of the spam emails without using a model while the lift curve (The curve in blue) represents the predicted result of the spam emails when the model is applied to the dataset. Depending on the distance of the curve from the base line the credibility of the model is determined. The model is considered to be more authentic if the distance is more from the baseline.

1. **Logistic Regression:**

   Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

   We applied logistic regression to the given dataset using XLMiner where the partition was 70-30. 70% was the training dataset and 30% was the validation dataset.

   PFB the O/P of logistic regression

Confusion Matrix:

(Training Dataset)

Confusion Matrix

	Predicted Class	
Actual Clas	1	0
1	1131	137
0	89	1864

(Validation Dataset)

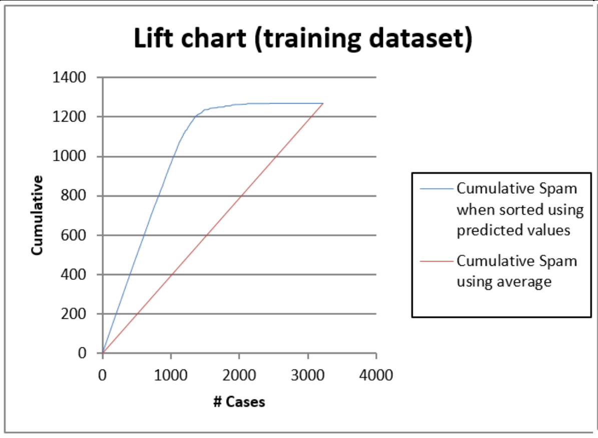
Confusion Matrix

	Predicted Class	
Actual Clas	1	0
1	483	62
0	46	789

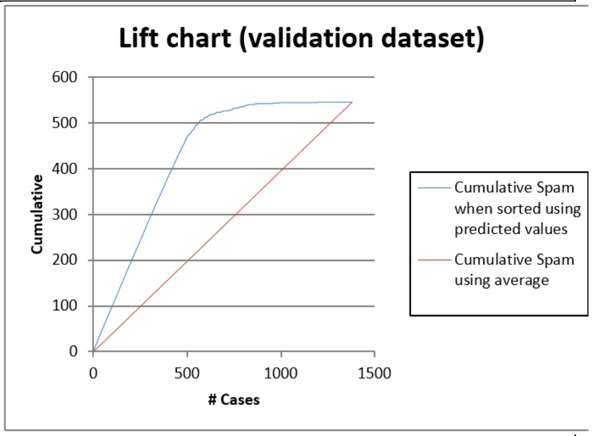
Confusion Matrix of Validation Dataset: From the confusion matrix of validation dataset we can see that the Logistic Regression Model predicted 483 correct values for class 1 and 62 wrong values for class 1. It predicted 789 correct values for class 0 and 46 wrong values for class 0.

Lift Chart:

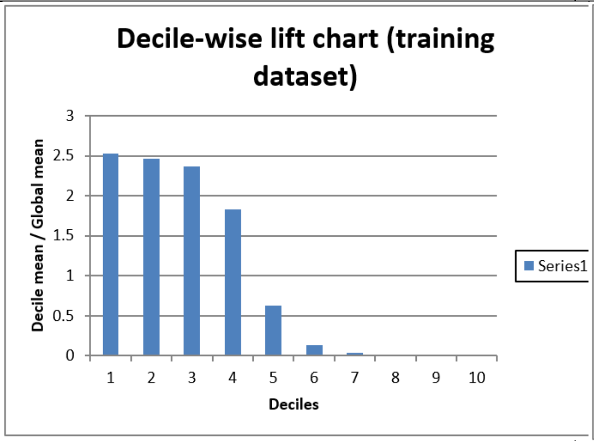
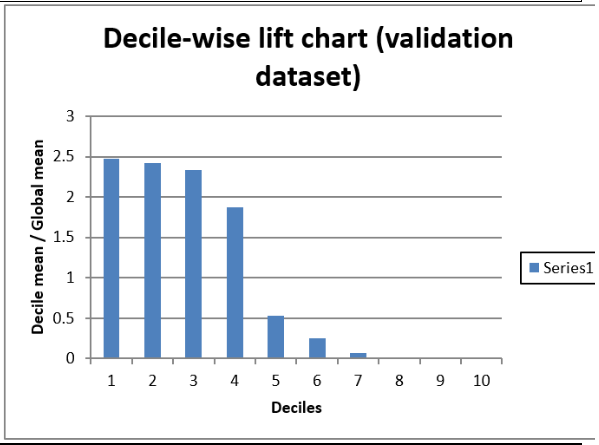
Training Dataset



Validation Dataset



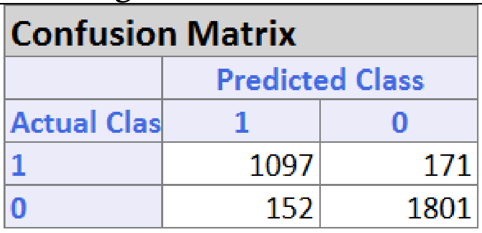
Decile-wise Lift Chart:

Training Dataset	**Validation Dataset**
	

2. **CART:**
   Classification and regression tree (CART) analysis recursively partitions observations in matched data set, consisting of a categorical (for classification trees) or continuous (for regression trees) dependent (response) variable and one or more independent (explanatory) variables, into progressively smaller groups. Each partition is a binary split based on a single independent variable.

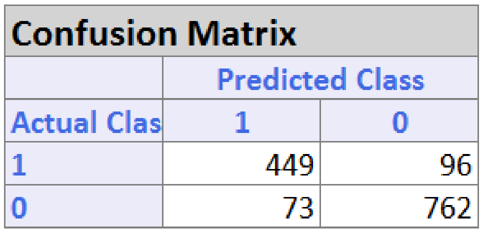
Confusion Matrix:

Training Dataset



Confusion Matrix		
	Predicted Class	
Actual Clas	1	0
1	1097	171
0	152	1801

Validation dataset

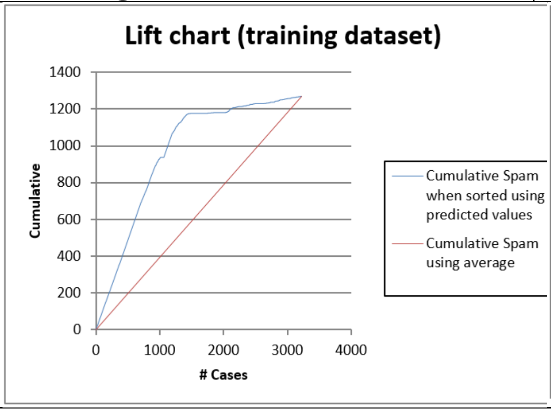


Confusion Matrix		
	Predicted Class	
Actual Clas	1	0
1	449	96
0	73	762

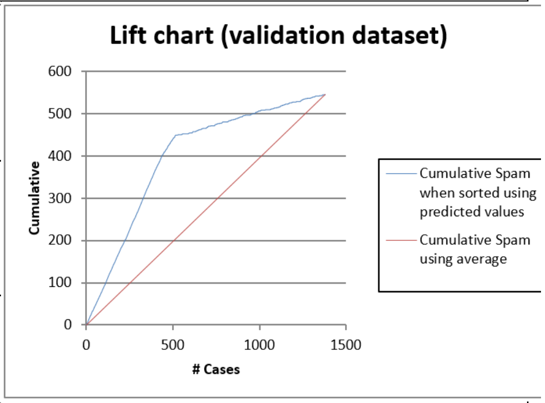
Confusion Matrix of Validation Dataset: From the confusion matrix of validation dataset we can see that the CART Model predicted 449 correct values for class 1 and 96 wrong values for class 1. It predicted 762 correct values for class 0 and 73 wrong values for class 0.

Lift Chart:

Training Dataset

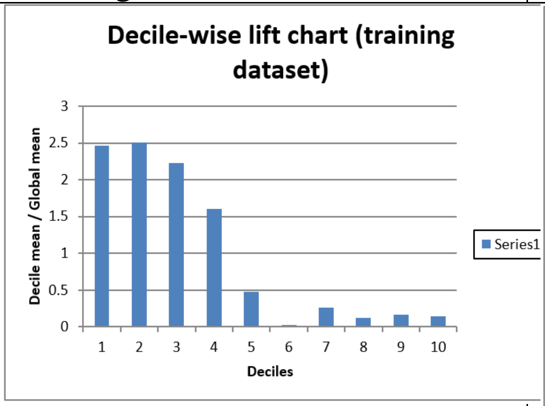


Validation Dataset

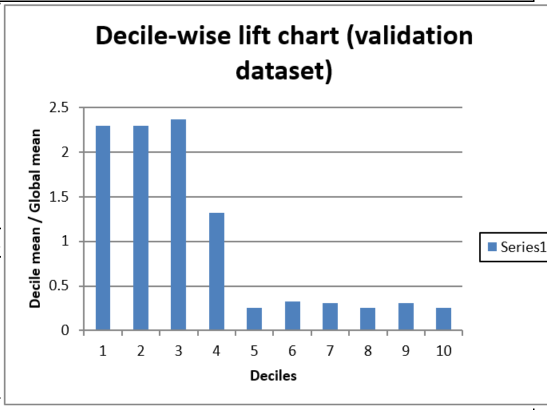


Decile-wise Lift Chart:

Training Dataset



Validation Dataset

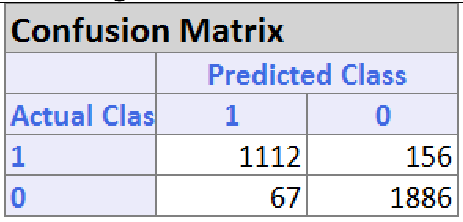


3. **Random Forest:**

Random forests is a notion of the general technique of random decision forests[1][2] that are an ensemble learning method
for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

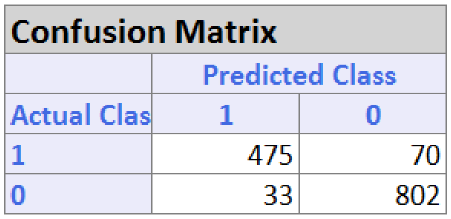
Confusion Matrix:

Training Dataset



Confusion Matrix		
	Predicted Class	
Actual Clas	1	0
1	1112	156
0	67	1886

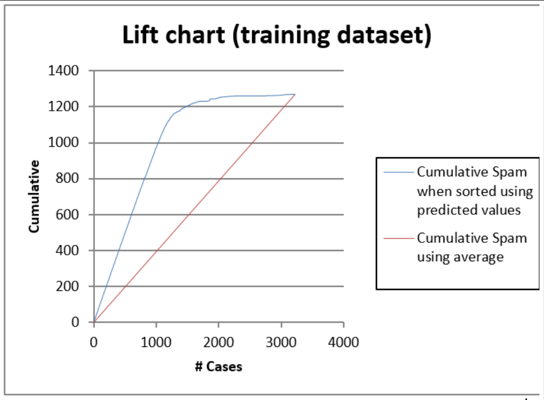
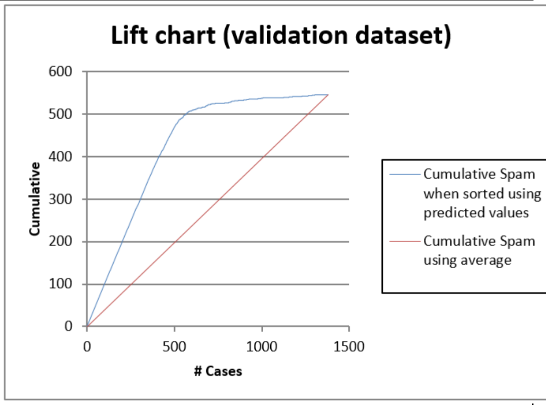
Validation Dataset



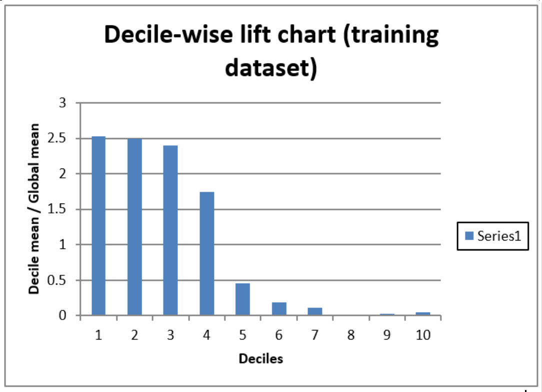
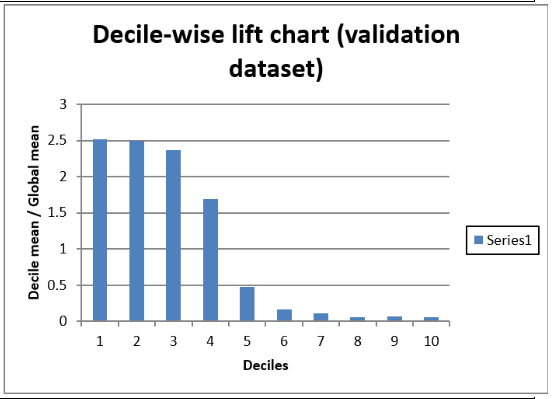
Confusion Matrix		
	Predicted Class	
Actual Clas	1	0
1	475	70
0	33	802

Confusion Matrix of Validation Dataset: From the confusion matrix of validation dataset we can see that the CART Model predicted 475 correct values for class 1 and 70 wrong values for class 1. It predicted 802 correct values for class 0 and 33 wrong values for class

Lift Chart:

Training Dataset	Validation Dataset
	

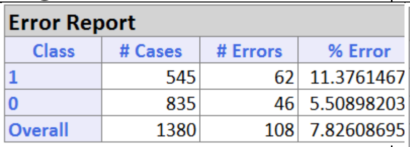
Decile-wise Lift Chart

Training Dataset	Validation Dataset
	

Comparison of the aforementioned Algorithms:

Logistic	CART	Random Forest

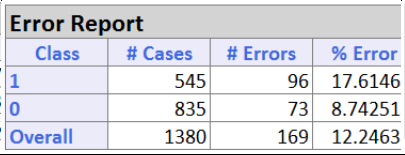
Logistic



Error Report

Class	# Cases	# Errors	% Error
1	545	62	11.3761467
0	835	46	5.50898203
Overall	1380	108	7.82608695

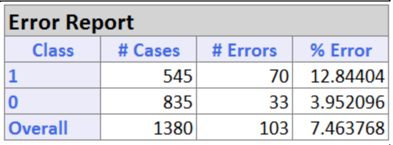
CART



Error Report

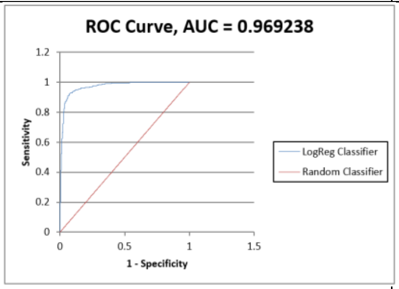
Class	# Cases	# Errors	% Error
1	545	96	17.6146
0	835	73	8.74251
Overall	1380	169	12.2463

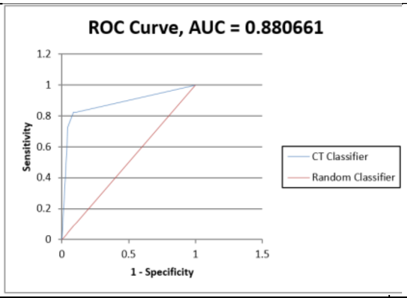
Random Forest

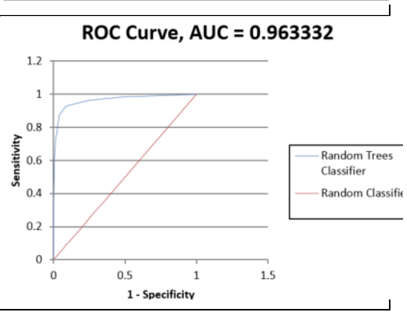


Error Report

Class	# Cases	# Errors	% Error
1	545	70	12.84404
0	835	33	3.952096
Overall	1380	103	7.463768







Conclusion:

If we look into the above table it's clearly seen that Random Forest provides the least value for overall error report as compared to Logistic and CART analysis. But if we compare the AUC values for all the 3 algorithms Logistic and Random Forest provides approximately same results with a difference of 0.005. So as per our studies I have chosen Random forest as the best analysis model for the given dataset.

Problem 4:

Random Forest:

Random forests is a notion of the general technique of random decision forest that are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

Working Principle:

Bootstrapping: Identification of a feature (column/ variable) that best splits the data i.e. the most predominant predictor variable in the data set

Randomization: The Random Forest algorithm randomly chooses predictor variables from data set to form decision trees.

Top Benefits of Random Forests

- Accuracy
- Runs efficiently on large data bases
- Handles thousands of input variables without variable deletion
- Gives estimates of what variables are important in the classification
- Generates an internal unbiased estimate of the generalization error as the forest building progresses
- Provides effective methods for estimating missing data
- Maintains accuracy when a large proportion of the data are missing
- Provides methods for balancing error in class population unbalanced data sets
- Generated forests can be saved for future use on other data

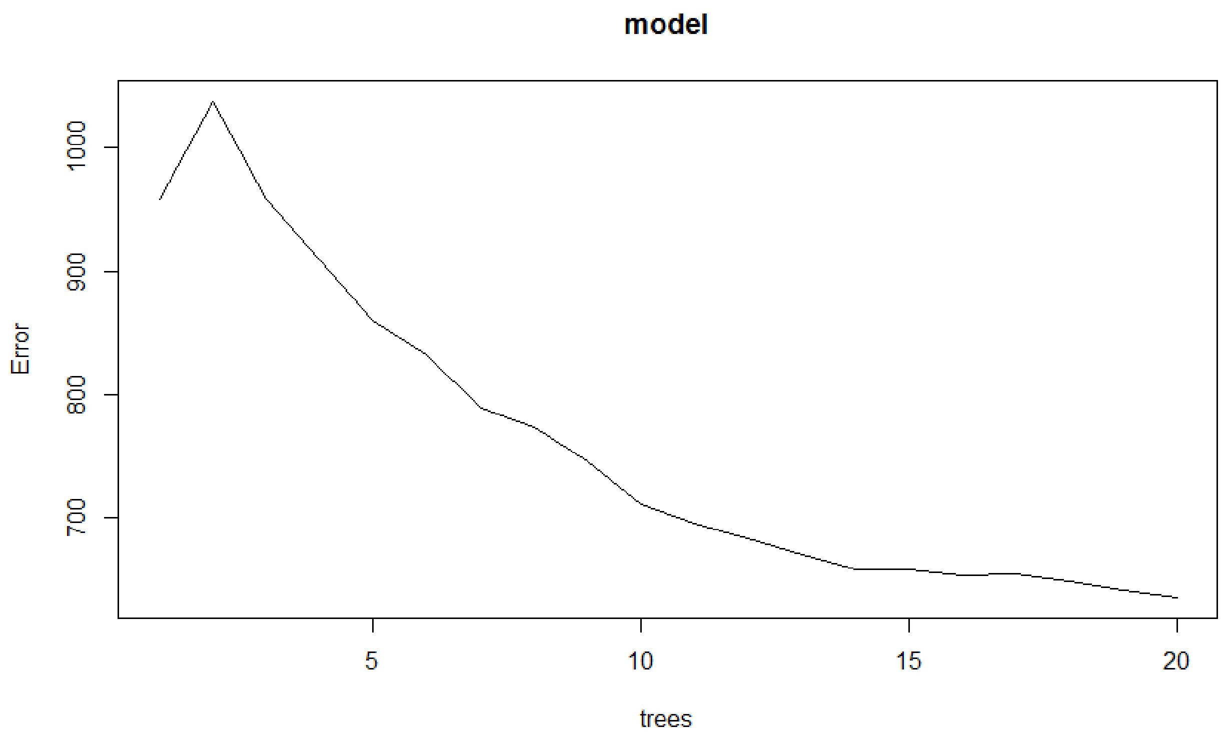
Steps taken while building random forest Algorithm:

- First we import train and test data into R
- Install **package randomForest** for this algorithm
- Then we build model on train data set by R code
  model <- randomForest(V281 ~ ., data=train, importance=TRUE, ntree=20)
  Here we use value of ntree is 20 which means algorithm build 20 tree and pick best tree amongst them
- Then we exclude Output data column for prediction
- Then we predict model by R code on test data
  blog_predict <- predict(model,test_blog)
- After predicting model we determine RMS value for this algorithm by R code
  rmse_blog <- sqrt(mean((blog_predict - test$V281)^2))
- **RMS** value for Random forest algorithm is **23.39957**
- Then We plot graph for RMS error for 20 trees using R code

R code for algorithm:

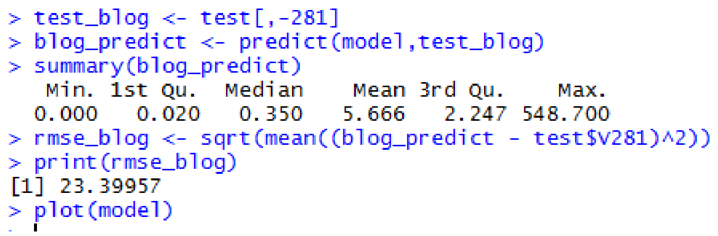
```
train <- read.csv(file="blogData_train.csv",header = FALSE)
train
install.packages("randomForest")
library(randomForest)

model <- randomForest(V281 ~ ., data=train, importance=TRUE, ntree=20 )

test <- read.csv(file="all.csv",header= FALSE)
summary(model)

#predict_test
test_blog <- test[,-281]
blog_predict <- predict(model,test_blog)
summary(blog_predict)

#rmse - test dataset
rmse_blog <- sqrt(mean((blog_predict - test$V281)^2))
print(rmse_blog)
plot(model)
```

Plot Graph:



Above graph explains RMS error against number of trees. We can see that for first five tree RMS error maximum. After first 3 trees value of RMS error start decreasing. At tree 20 error is minimum.

RMS error in R:



```
> test_blog <- test[,-281]
> blog_predict <- predict(model,test_blog)
> summary(blog_predict)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   0.020   0.350   5.666   2.247 548.700 
> rmse_blog <- sqrt(mean((blog_predict - test$V281)^2))
> print(rmse_blog)
[1] 23.39957
> plot(model)
```

CART:

Classification and Regression Tree Analysis, CART, is a simple yet powerful analytic tool that helps determine the most "important" (based on explanatory power) variables in a particular dataset, and can help researchers craft a potent explanatory model. This technical report will outline a brief background of CART followed by practical applications and suggested implementation strategies in R for the given dataset.

CART methodology consists of three parts:

1. Construction of maximum tree

2. Choice of the right tree size

3. Classification of new data using constructed tree

Advantages of CART:

- CART is nonparametric: Therefore this method does not require specification of any functional form
- CART does not require variables to be selected in advance: CART algorithm will itself identify the most significant variables and eliminate non-significant ones
- CART results are invariant to monotone transformations of its independent variables.
- CART can easily handle outliers
- CART has no assumptions and computationally fast.
- CART is flexible and has an ability to adjust in time.

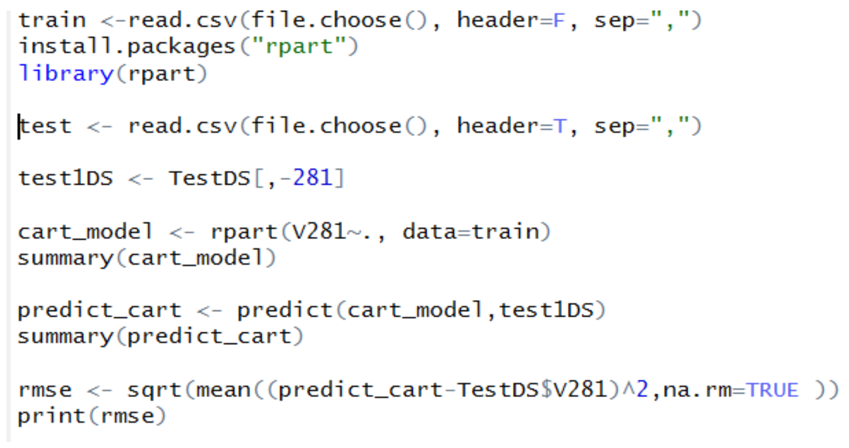
Disadvantages of CART

- CART may have unstable decision trees
- CART splits only by one variable

Steps taken while building random forest Algorithm:

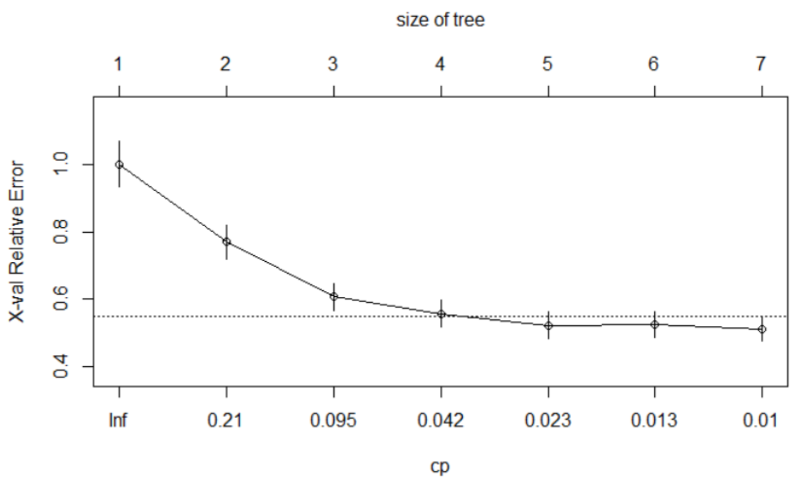
- First we import train and test data into R
- Install **package rpart** for this algorithm
- Then we build model on train data set by R code
  cart_model <- rpart(V281~., data=train)
- Then we exclude Output data column for prediction
- Then we predict model by R code on test data
  predict_cart <- predict(cart_model,test1DS)
- After predicting model we determine RMS value for this algorithm by R code
  rmse <- sqrt(mean((predict_cart-TestDS$V281)^2,na.rm=TRUE))RMS value for Random forest algorithm is **24.3881926881315**
- Then We plot graph for Cross Validation Error
  plotcp(cart_model)
- Next we plot the tree : **plot(cart_model) text(cart_model)**

R Code for CART:

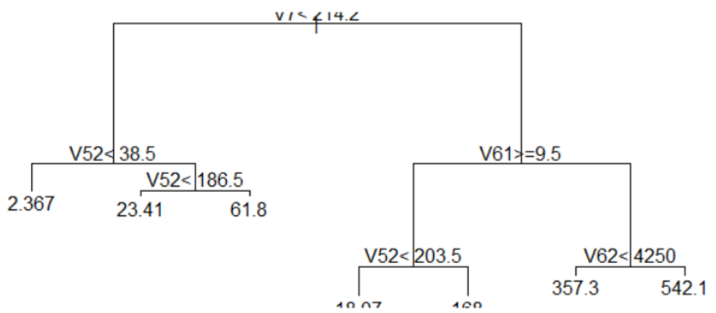


```
train <-read.csv(file.choose(), header=F, sep=",")
install.packages("rpart")
library(rpart)

test <- read.csv(file.choose(), header=T, sep=",")

test1DS <- TestDS[,-281]

cart_model <- rpart(V281~., data=train)
summary(cart_model)

predict_cart <- predict(cart_model,test1DS)
summary(predict_cart)

rmse <- sqrt(mean((predict_cart-TestDS$V281)^2,na.rm=TRUE ))
print(rmse)
```

Cross Validation Error Plot:



Tree Plot:



Multiple Linear Regression:

The multiple liner regression model was built using the variable V281 (**the number of comments in the next 24 hours**) as the outcome variable. The rest of the variables were fed in to the model as independent variables for prediction.

The following variables had a significant relation with the outcome variable:

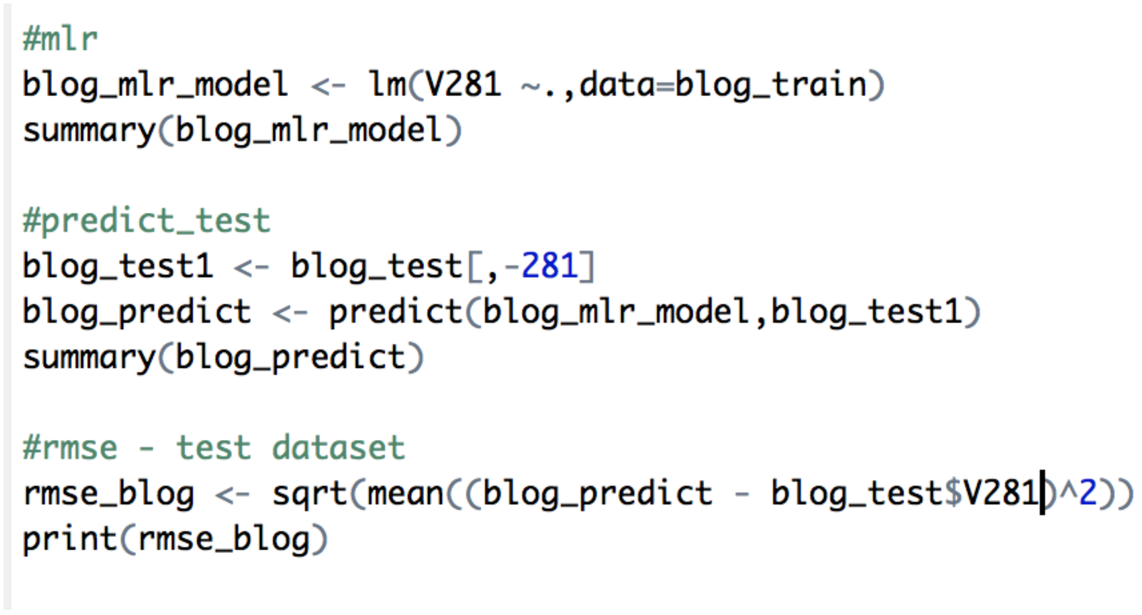
V51: Total number of comments before basetime

V52: Number of comments in the last 24 hours before the basetime

V53: Let T1 denote the datetime 48 hours before basetime, Let T2 denote the datetime 24 hours before basetime. This attribute is the number of comments in the time period between T1 and T2

V54: Number of comments in the first 24 hours after the publication of the blog post, but before basetime

R code for the Multiple Linear Regression:


```
#mlr
blog_mlr_model <- lm(V281 ~.,data=blog_train)
summary(blog_mlr_model)

#predict_test
blog_test1 <- blog_test[,-281]
blog_predict <- predict(blog_mlr_model,blog_test1)
summary(blog_predict)

#rmse - test dataset
rmse_blog <- sqrt(mean((blog_predict - blog_test$V281)^2))
print(rmse_blog)
```

#mlr

blog_mlr_model <- lm(V281 ~.,data=blog_train)

summary(blog_mlr_model)

The summary of the model is as given below:

Residuals:

```
    Min      1Q  Median      3Q     Max

-270.51   -5.09   -1.10     2.77 1342.40
```

Residual standard error: 30.12 on 52148 degrees of freedom

Multiple R-squared: 0.3648,   **Adjusted R-squared:** 0.3617

F-statistic: 120.7 on 248 and 52148 DF, p-**value:** < 2.2e-16

```
#predict_test
blog_test1 <- blog_test[,-281]
blog_predict <- predict(blog_mlr_model,blog_test1)
summary(blog_predict)

> summary(blog_predict)
```

Min. 1st Qu.   Median   Mean 3rd Qu.   Max.

```
-107.600   -1.765    2.167    5.263    6.519  278.800
```

```
#rmse - test dataset

rmse_blog <- sqrt(mean((blog_predict - blog_test$V281)^2))

print(rmse_blog)


[1] 25.44774
```

Conclusion:

Random Forest RMSE: 23.399

CART RMSE: 24.388

Multiple Linear Regression RMSE: 25.447

From the above RMSE values we conclude that Random Forest is the best model for prediction of the number of comments in the next 24 hours.