Midterm case study

INFO7390 Advances in Data Sciences and Architecture

March 18, 2016

Group 9

* Neha Gilson
* Aastha Grover
* Shuxian Wu

**West Roxbury dataset**

## About the Data:

The dataset includes details about residential homes in Roxbury like the area of house, number of rooms, bedrooms, baths, kitchen and if it is remodeled. The only categorical variable was “Remodel”.

## Goal:

The goal of this case is to make a prediction for the total value of a home. The original dataset comes with following variables:

Input Variables:

|  |  |
| --- | --- |
| TAX | Tax bill amount based on total assessed value multiplied by the tax rate |
| LOT SQFT | Total lot size of parcel in square feet |
| YR BUILT | Year property was built |
| GROSS AREA | Gross floor area |
| LIVING AREA | Total living area for residential properties (ft2) |
| FLOORS | Number of floors |
| ROOMS | Total number of rooms |
| BEDROOMS | Total number of bedrooms |
| FULL BATH | Total number of full baths |
| HALF BATH | Total number of half baths |
| KITCHEN | Total number of kitchens |
| FIREPLACE | Total number of fireplaces |
| REMODEL | When house was remodeled (Recent/Old/None) |

Output Variable:

|  |  |
| --- | --- |
| TOTAL VALUE | Total assessed value for property, in thousands of USD |

## Initial assumptions:

We took a look at the dataset and got a sense of how these variables will work interactively. We made assumptions, such as the tax variable may not be valid, considering the total value since the tax is calculated based on the value of home with a constant tax rate. Secondly, the remodel variable needs to be transformed into numerical value for prediction in regression since it is categorical.

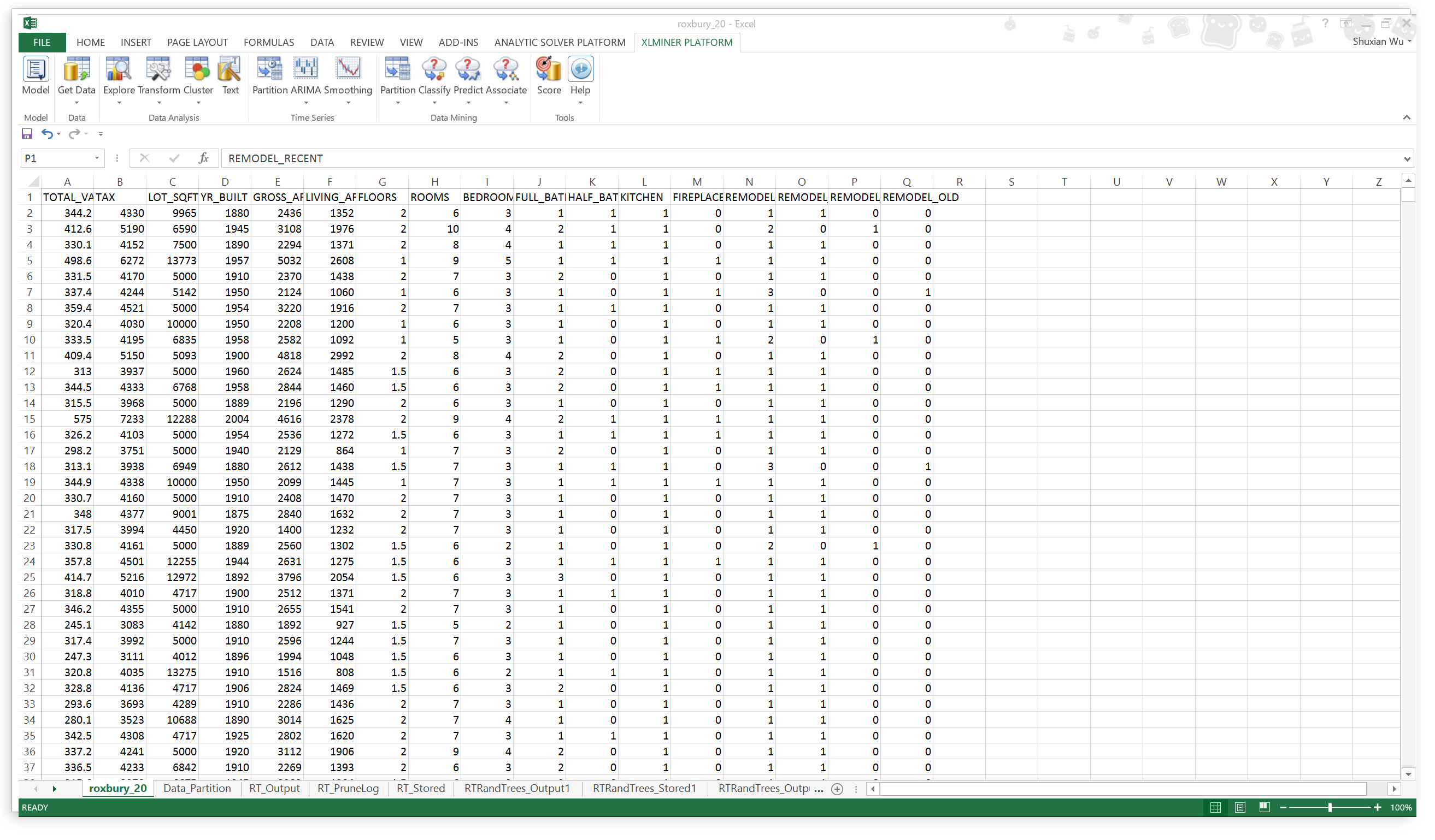
## Preprocessing:

We used Wrangler to clean the dataset by handling missing data, data format and data transformation.

For the remodel variable we created dummies in XLMiner:

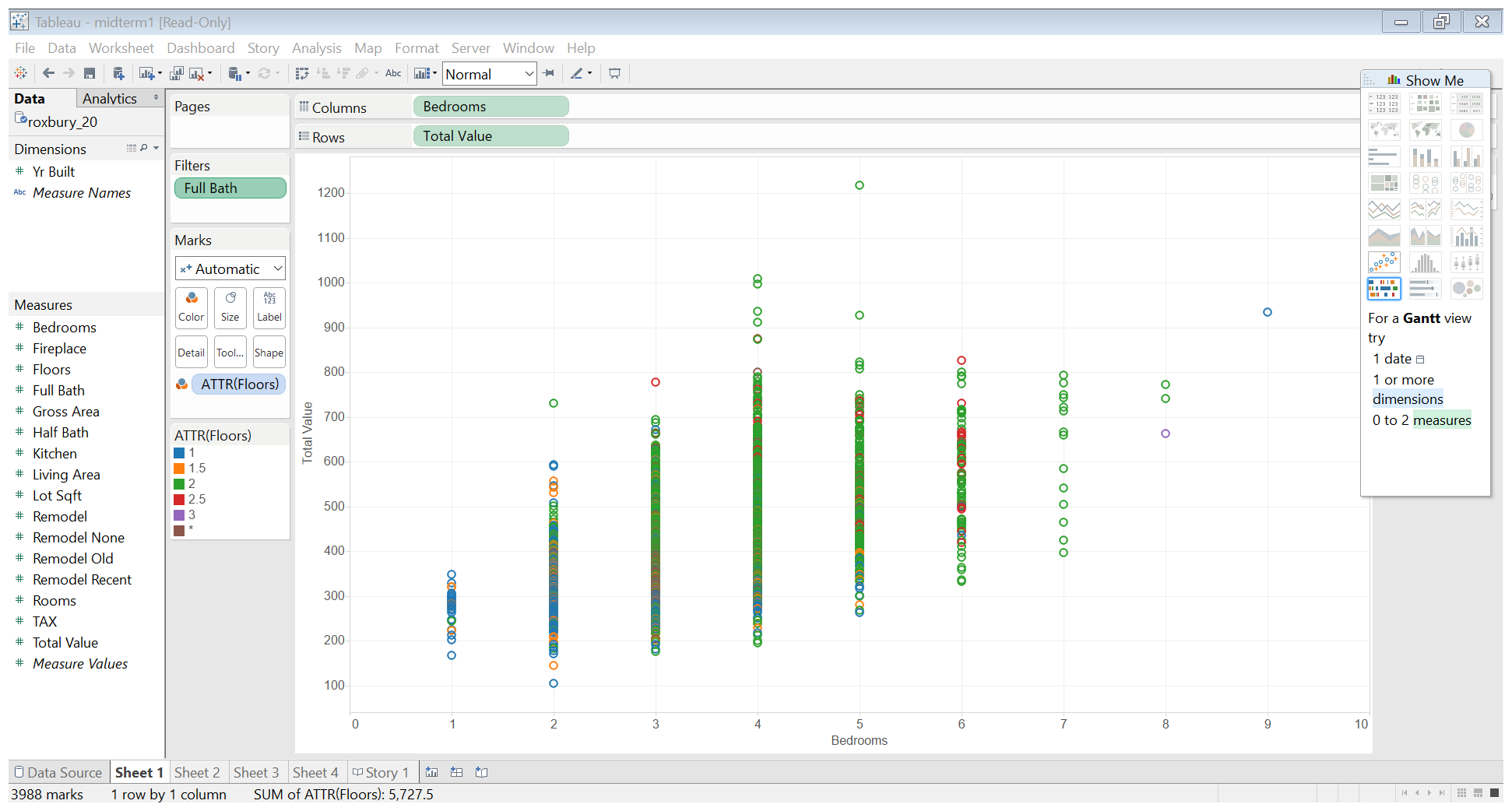
1. None
2. Recent
3. Old

This is how the data set looked after cleansing and transforming.



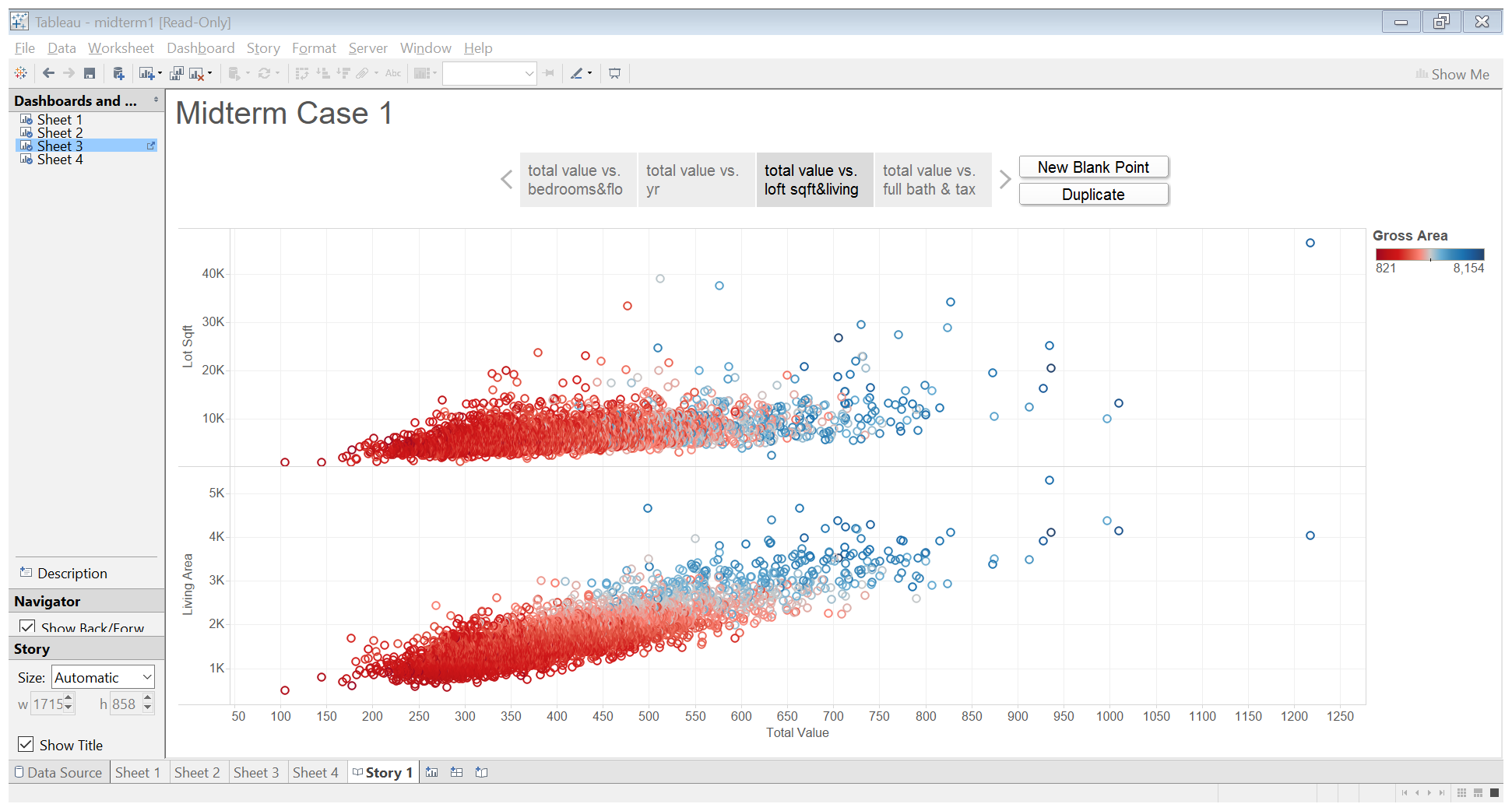
## Data Exploration:

Through exploration in Tableau, we found that an increase in the number of Bedrooms does not have a very heavy impact on the value of the home. As seen below, the highest value homes are not necessarily the ones with more number of bedrooms, though that is the trend.

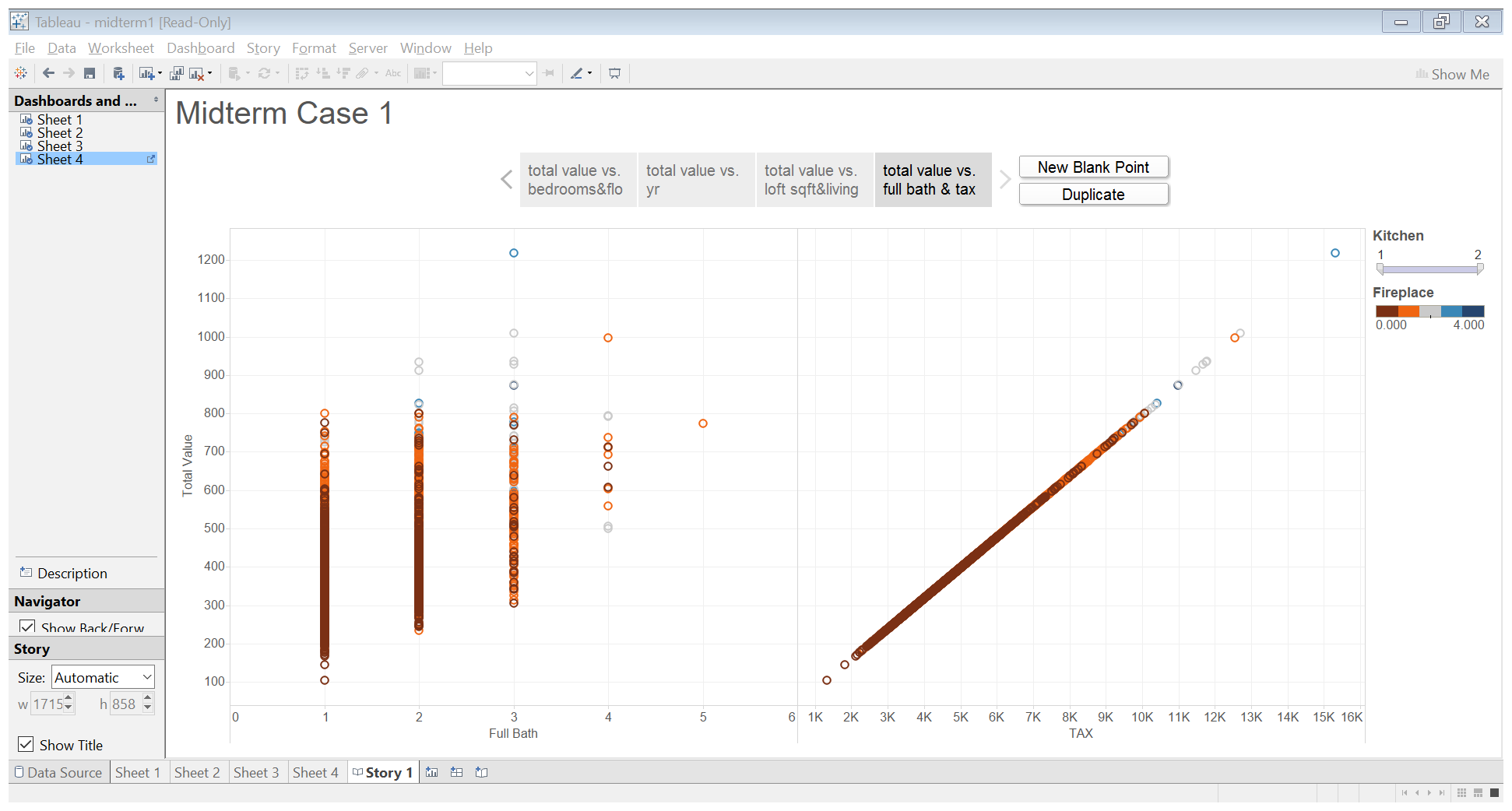


We also saw that most of the recently built houses that aren’t remodeled have higher value.

An interesting observation was the area of living area had more impact on the value of the home that the total lot area.



We saw that (on the right) Tax and total value have a very close correlation and therefore it will have lesser impact on the prediction model.

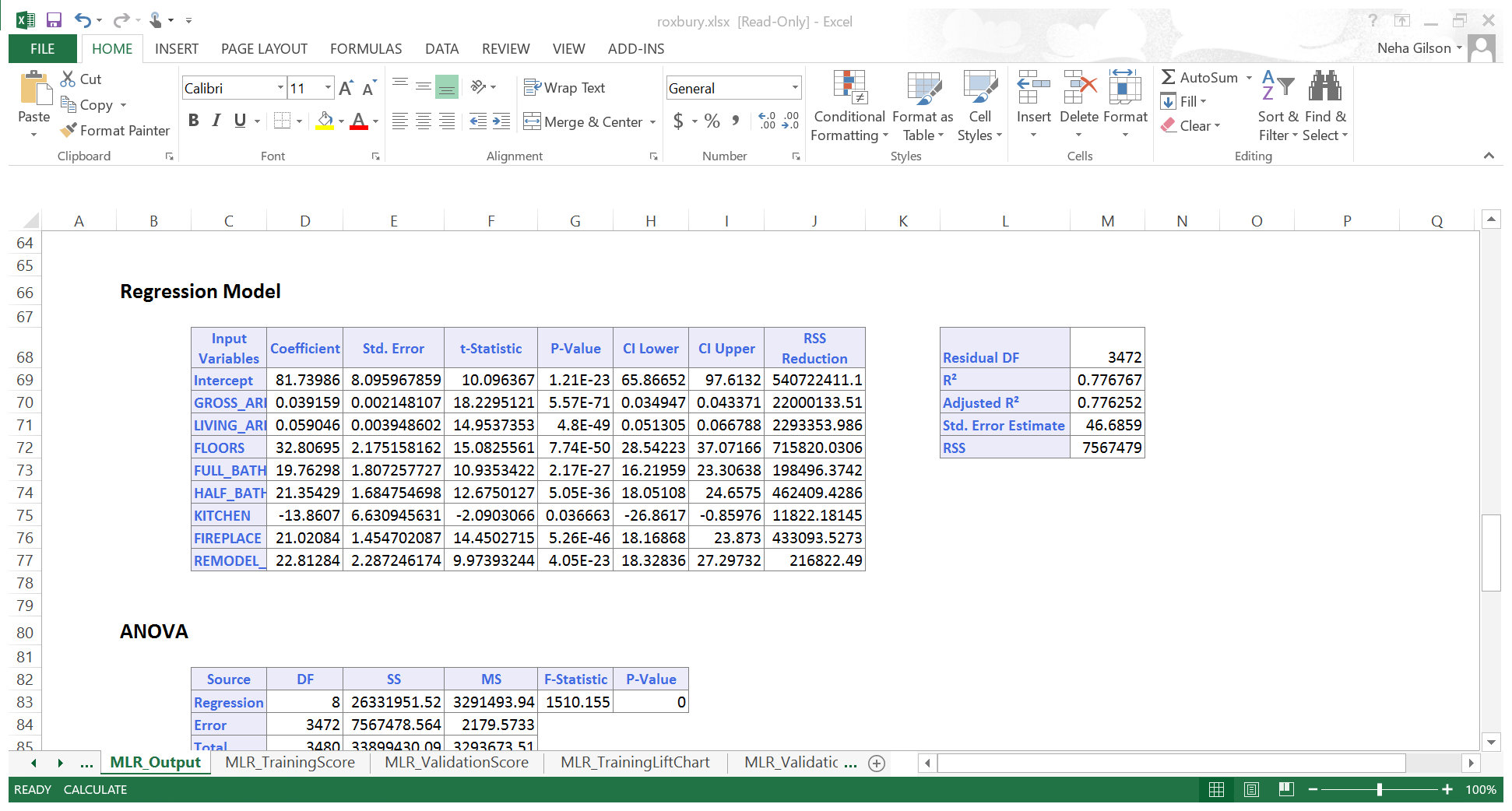


We then used this analysis for variable selection in XLMiner.

## Prediction Models

1. Regression:

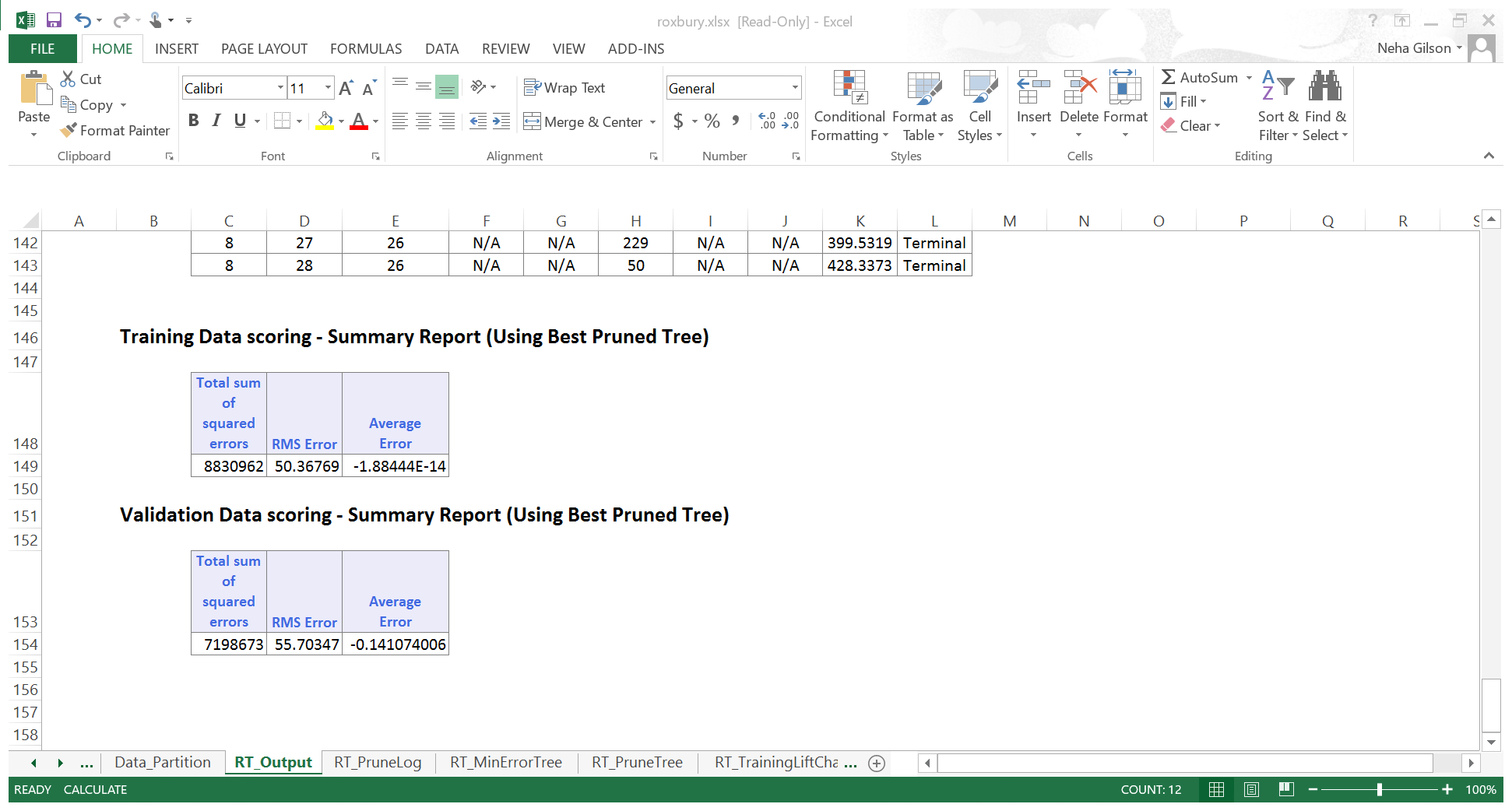
We used Multiple Linear Regression. The Regression Model and the R^2 values are as shown below for the best combination of selected features.



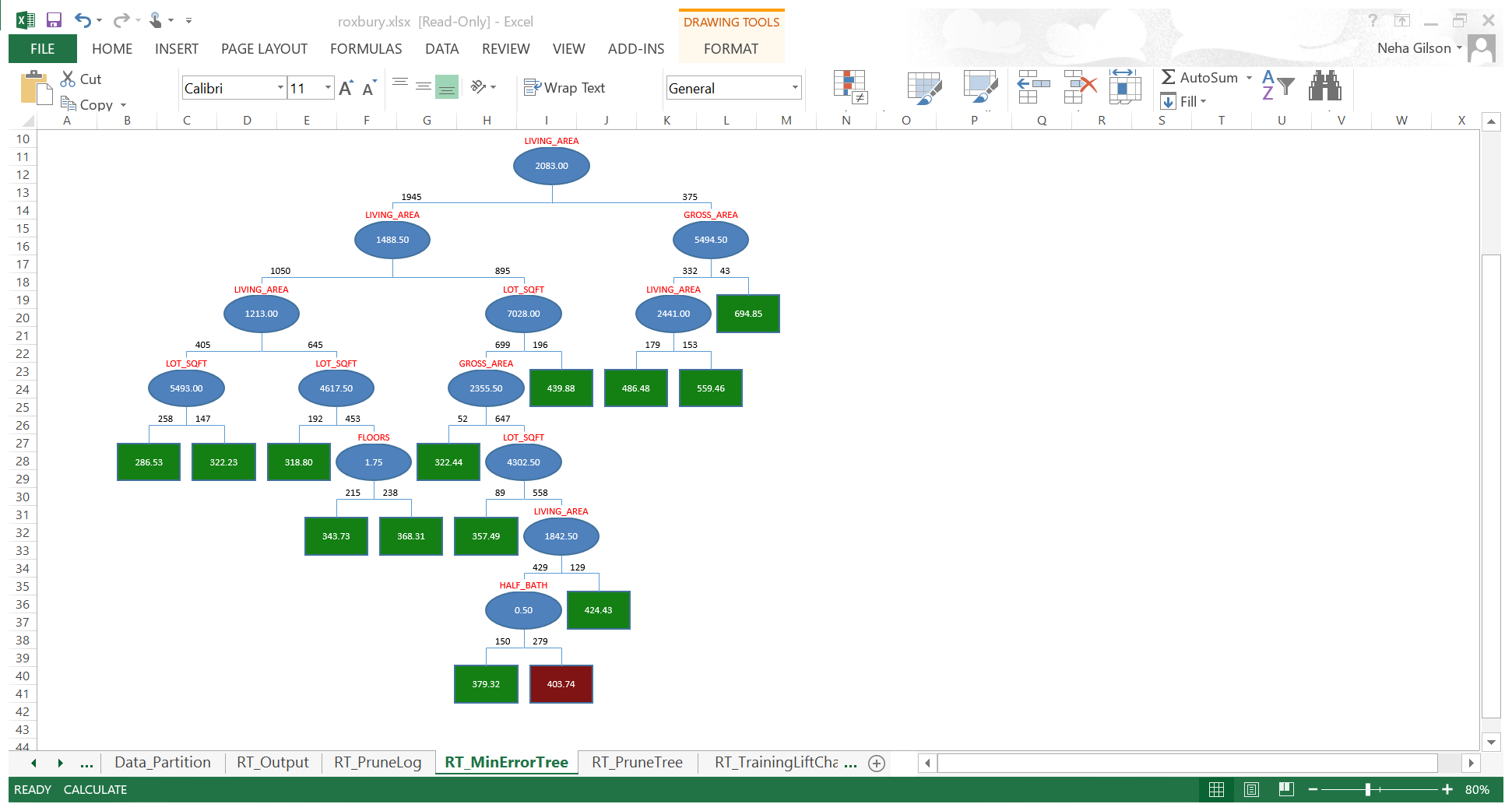
The RMSE error is as shown below for training and validation data.



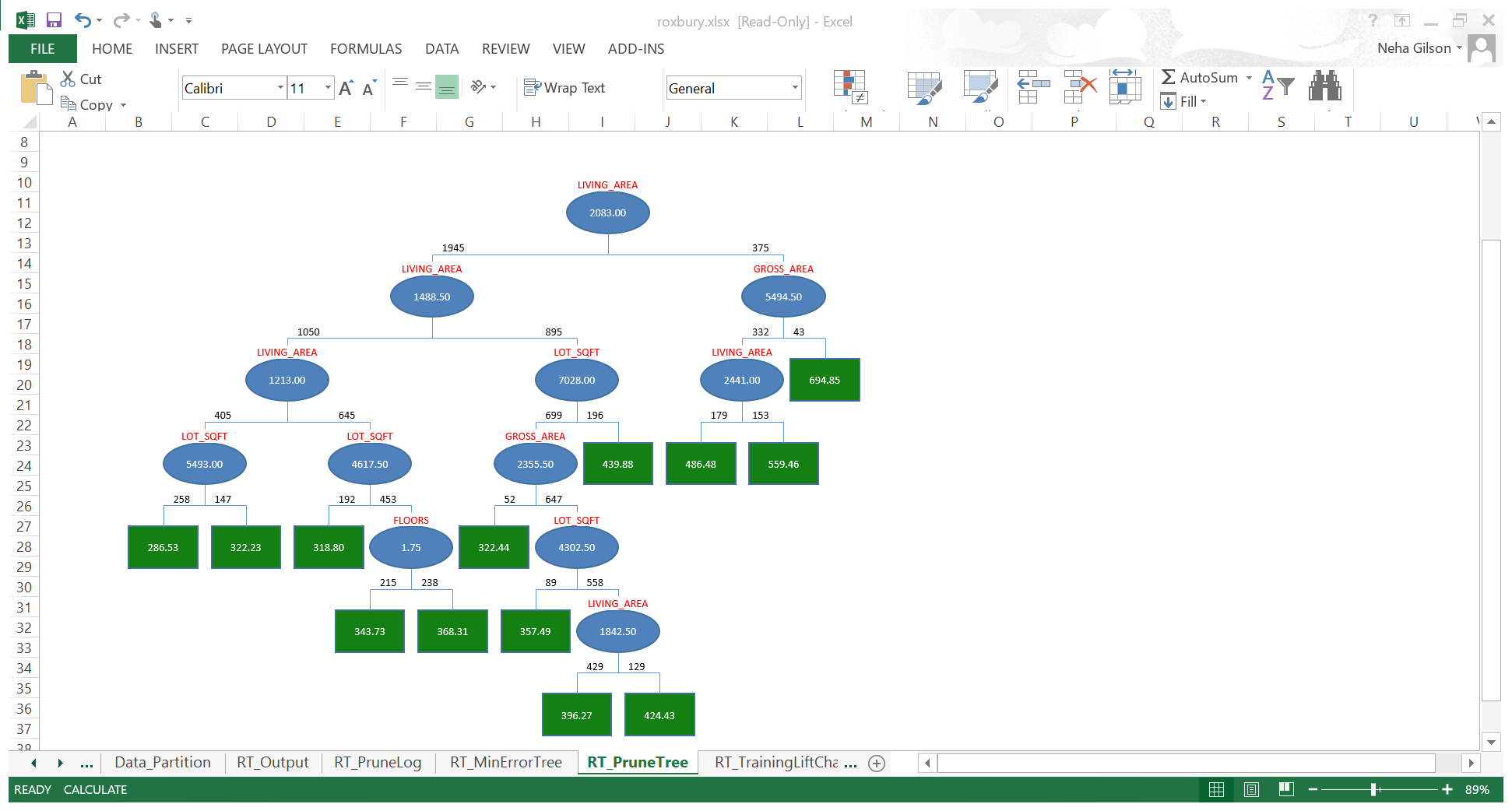
1. CART



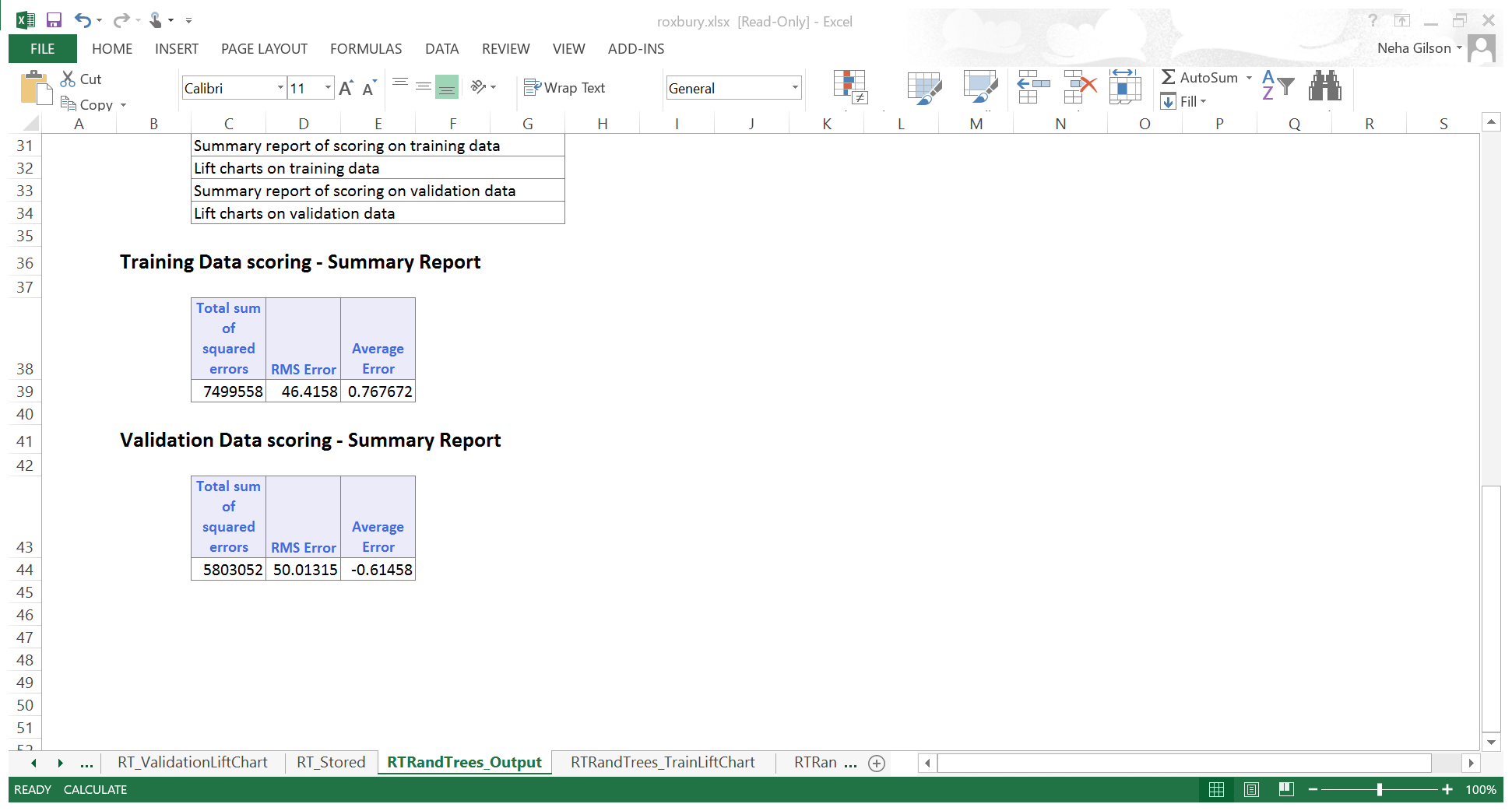
**Minmum Error Tree:**



**Best Pruned Tree:**



1. Random Forest



## Conclusions:

We compared the RMS errors and AUC in all the 3 above models and clearly, the best model is Multiple Linear Regression. RMSE has a significantly higher correlation to the distance from the ground truth on average than AUC.

We would recommend that for future prediction of home values the MLR model gives a more accurate result.

**Mortgage Defaults**

## About the data:

The dataset contains approved loans and the factors affecting whether these loans were defaulted or not.

## Goal:

The goal of this case is to classify whether any future approved loans will be default or non default.

Input Variables:

|  |  |
| --- | --- |
| Bo\_Age | Borrower age |
| Ln\_Orig | Value of loan, USD |
| Orig\_LTV\_Ratio\_Pct | Ratio of loan to home purchase price |
| Credit\_score | Borrower's credit score |
| First\_home | First time home buyer? (Y/N) |
| Tot\_mthly\_debt\_exp | Borrower's total monthly debt expense |
| Tot\_mthly\_incm | Borrower's total monthly income |
| orig\_apprd\_val\_amt | Appraised value of home at origination |
| pur\_prc\_amt | Purchase price for house |
| DTI\_ratio | Borrower debt to income ratio (Tot\_mthly\_debt\_exp/Tot\_mthly\_incm) |
| Status | Current loan status |
| State | US state in which home is located |
| Median\_state\_inc | Median household income by state 2002-2004 |
| UPB>Appraisal | Loan amount (Ln\_Orig) greater than appraisal (orig\_apprd\_val\_amt) 0-no, 1=yes |

Output Variable:

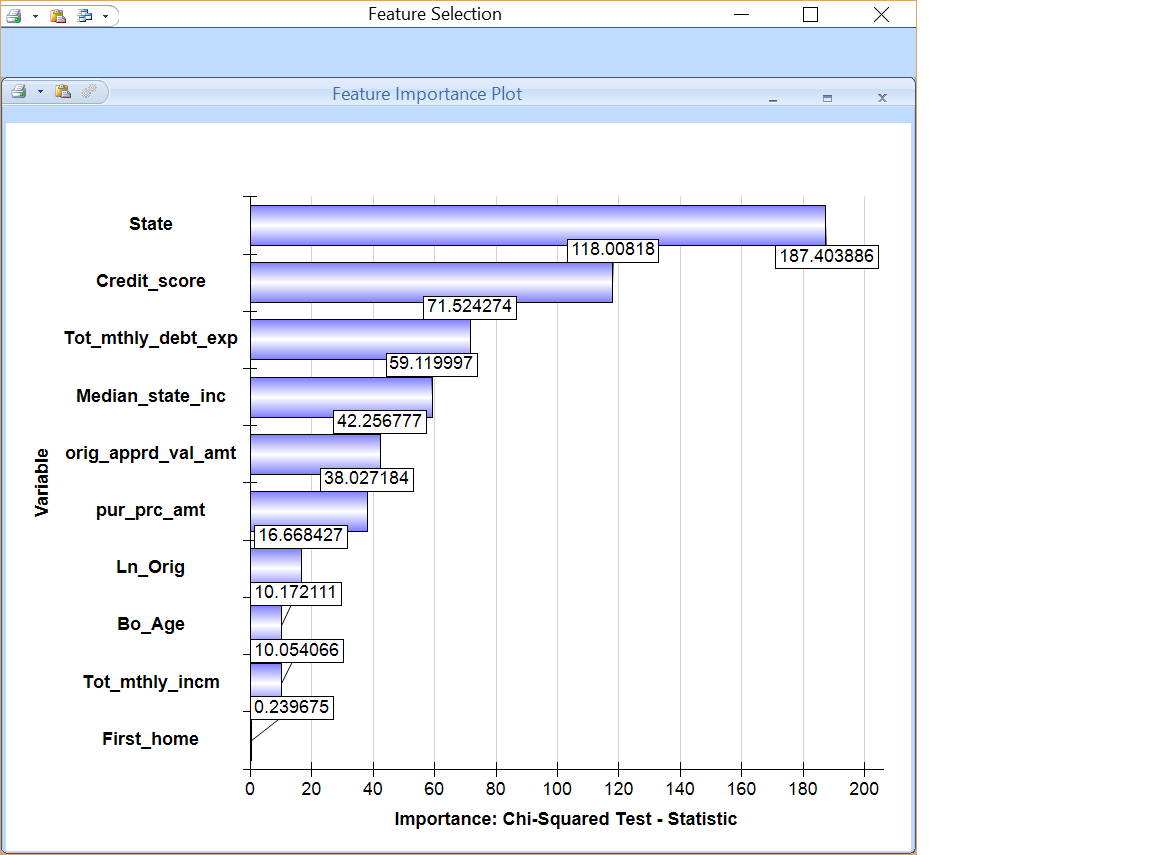
|  |  |
| --- | --- |
| OUTCOME | Binary version of "Status" (either default or non-default) |

## Initial assumptions:

We took a look at the dataset and got a sense of how these variables will work interactively. We made assumptions that since there are features that are derived from other features they could be ignored in the classification model. For e.g. DTI ration is derived from Total monthly debt and total monthly income of the borrower.

## Preprocessing:

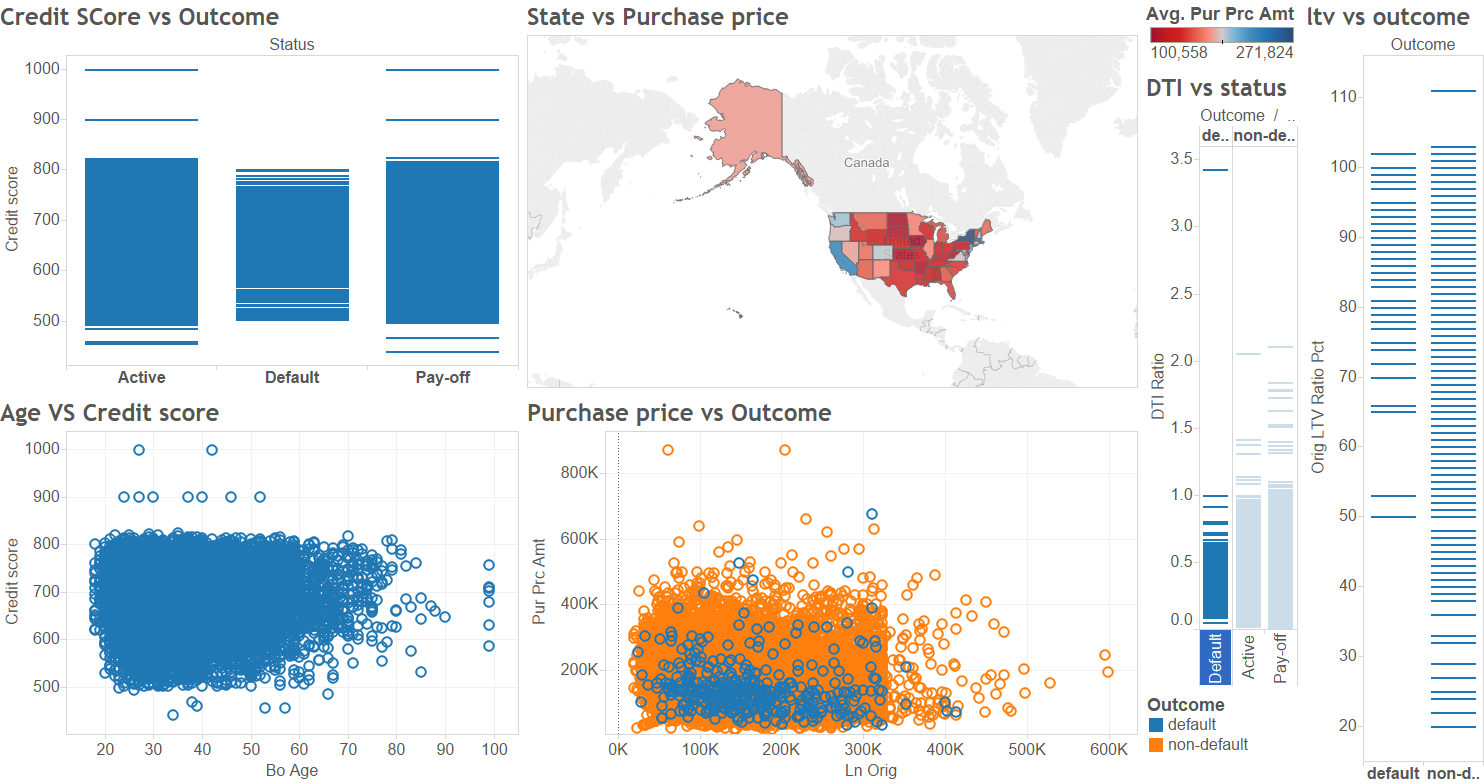
The last 2 columns of the data were irrelevant and they were removed. There was no missing data. There were some data that had Credit score above 850 which is an anomaly, so we removed them. There were a few categorical data for which we created dummy variables. Also, the data contained status of mortgage based on states. Since there were 50 such variables we used feature selection to decide which variables had more impact on the model.



When we built the model based on the above variables, we noticed that including or dropping all the states in the model did not have a major impact. So we dropped the states with the least impact.

## Data Exploration

The dashboard for data exploration in Tableau is as shown below:

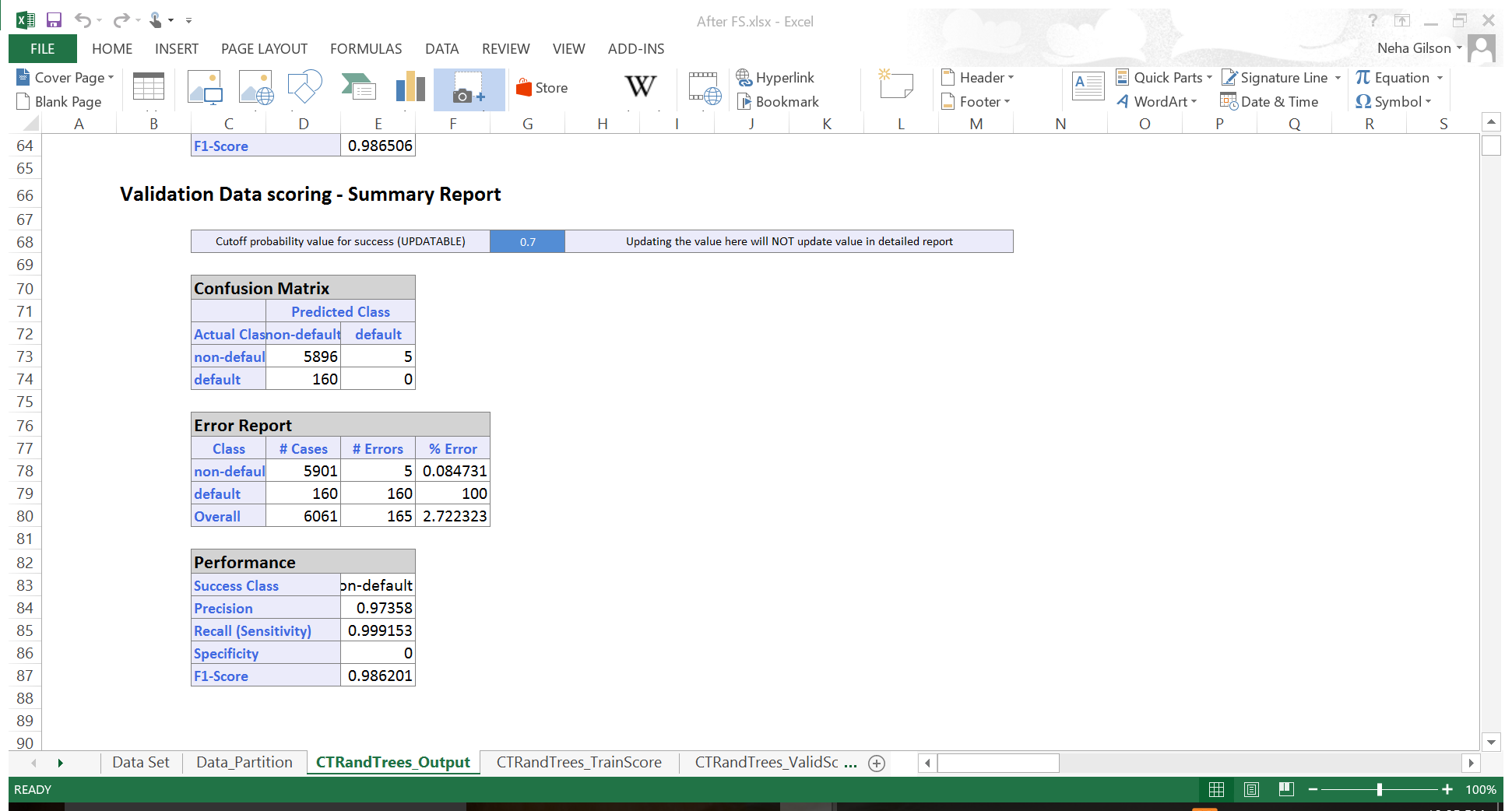


* The first graph shows the relation between Credit score and Outcome of the mortgage. We see that most of the defaulters fall in the range of 570 – 766.
* The second graph shows that relation between Age and Credit score. We see that most of the older age group people have better credit scores and thus in turn are non-defaulters as per the first graph.
* The third graph shows the purchase price of the homes through different states. We can see that New York state and California have the highest purchase prices.
* The fourth graph shows purchase price versus outcome. It is interesting to see that most of the defaulters are for lower priced homes and lower loan amount mortgages indicated by blue. Whereas as the purchase price increases the loan amount increases and the mortgage ends up in the non-default category.
* We also see from the fifth and sixth graph that most of the defaulters have a lower DTI ratio and higher LTV ratio.

These explorations help us understand the effect of different features in the model. We understand that a good credit score would lean towards the non-defaulter category. The higher purchase price of the homes combined with lower LTV ratio and good DTI Ratio will probably be a non-defaulter.

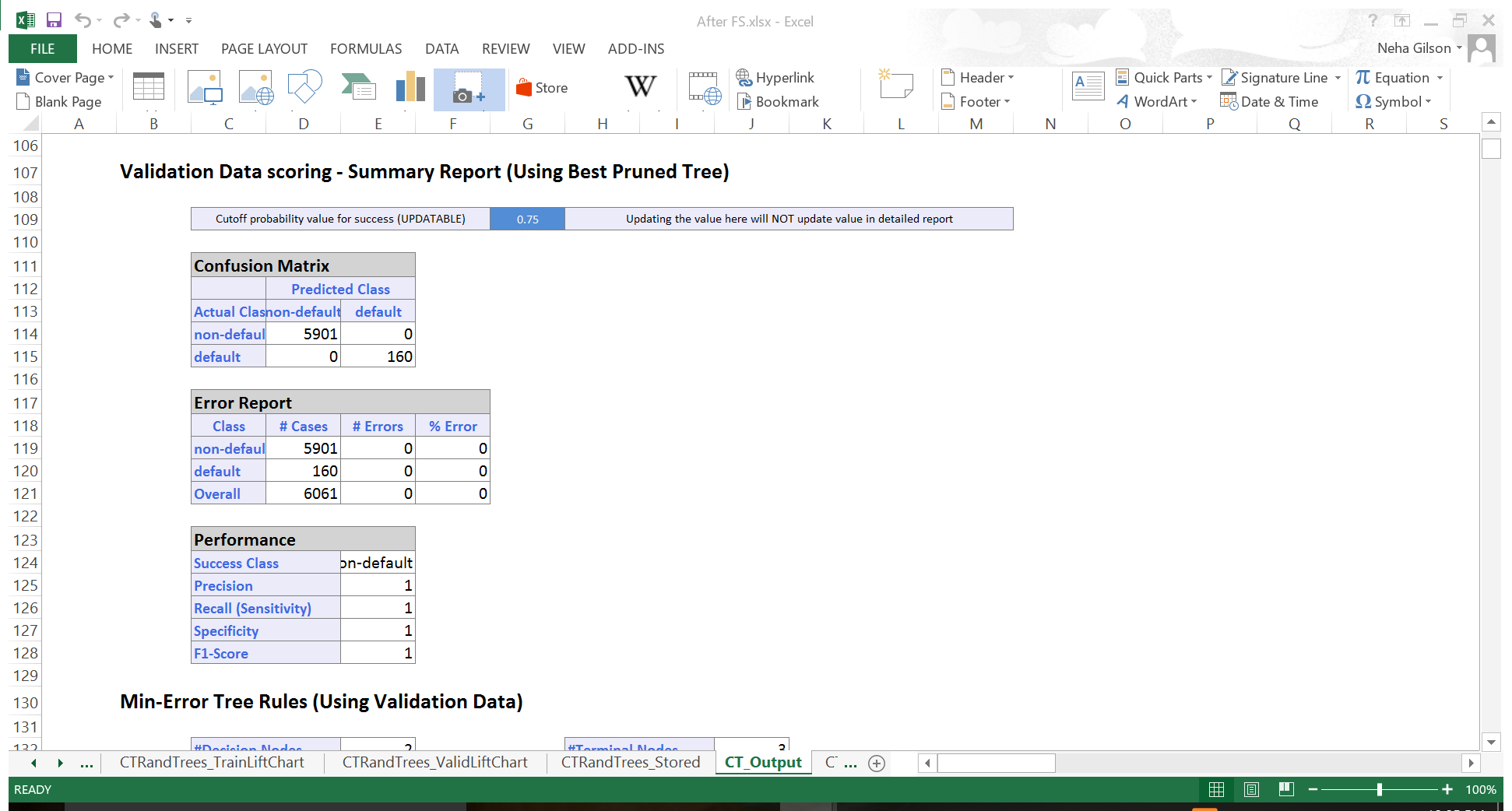
## Classification Model:

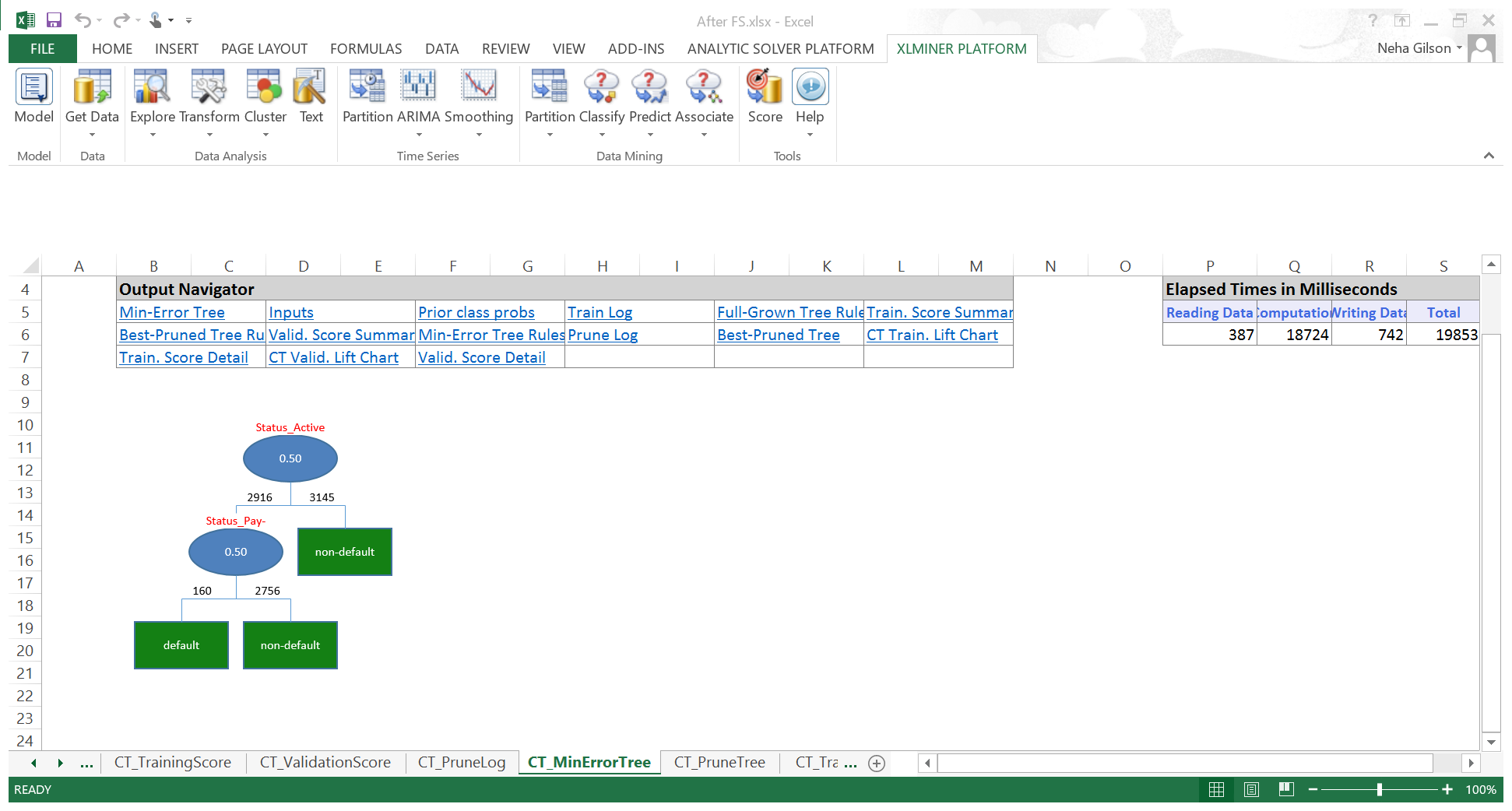
**Random Forest:**

****

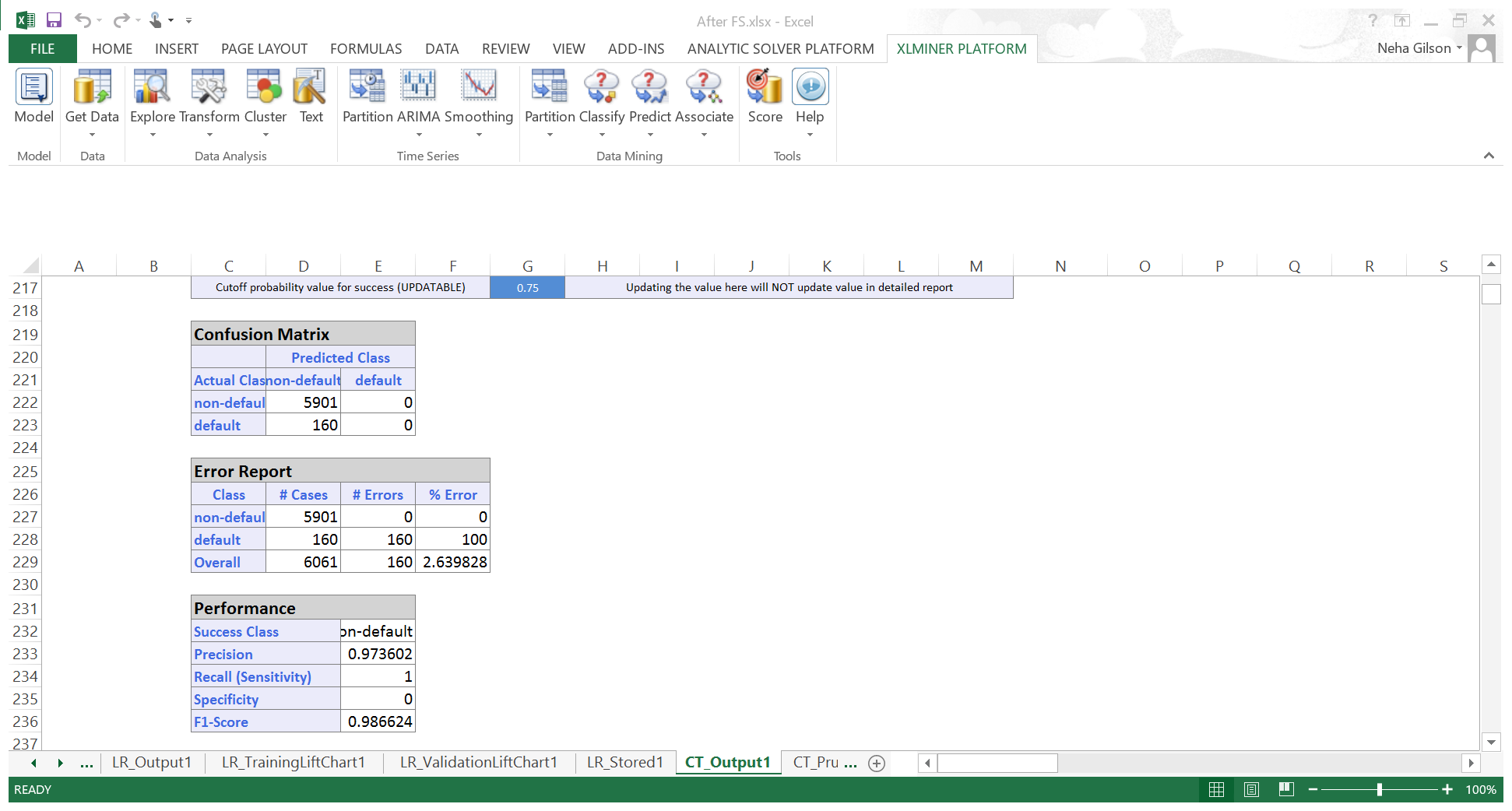
**CART:**

When considering status:

****

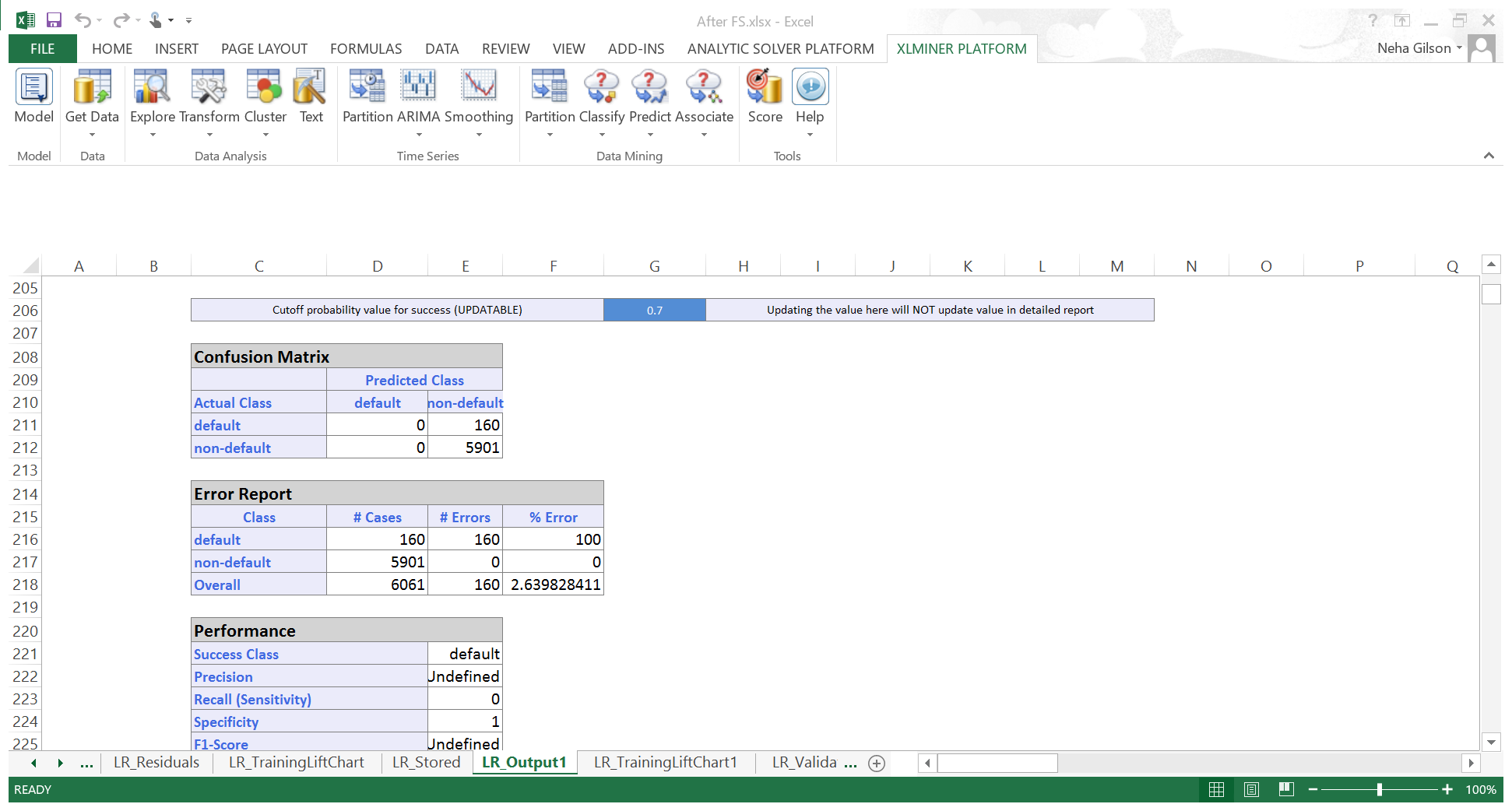


When not considering status:

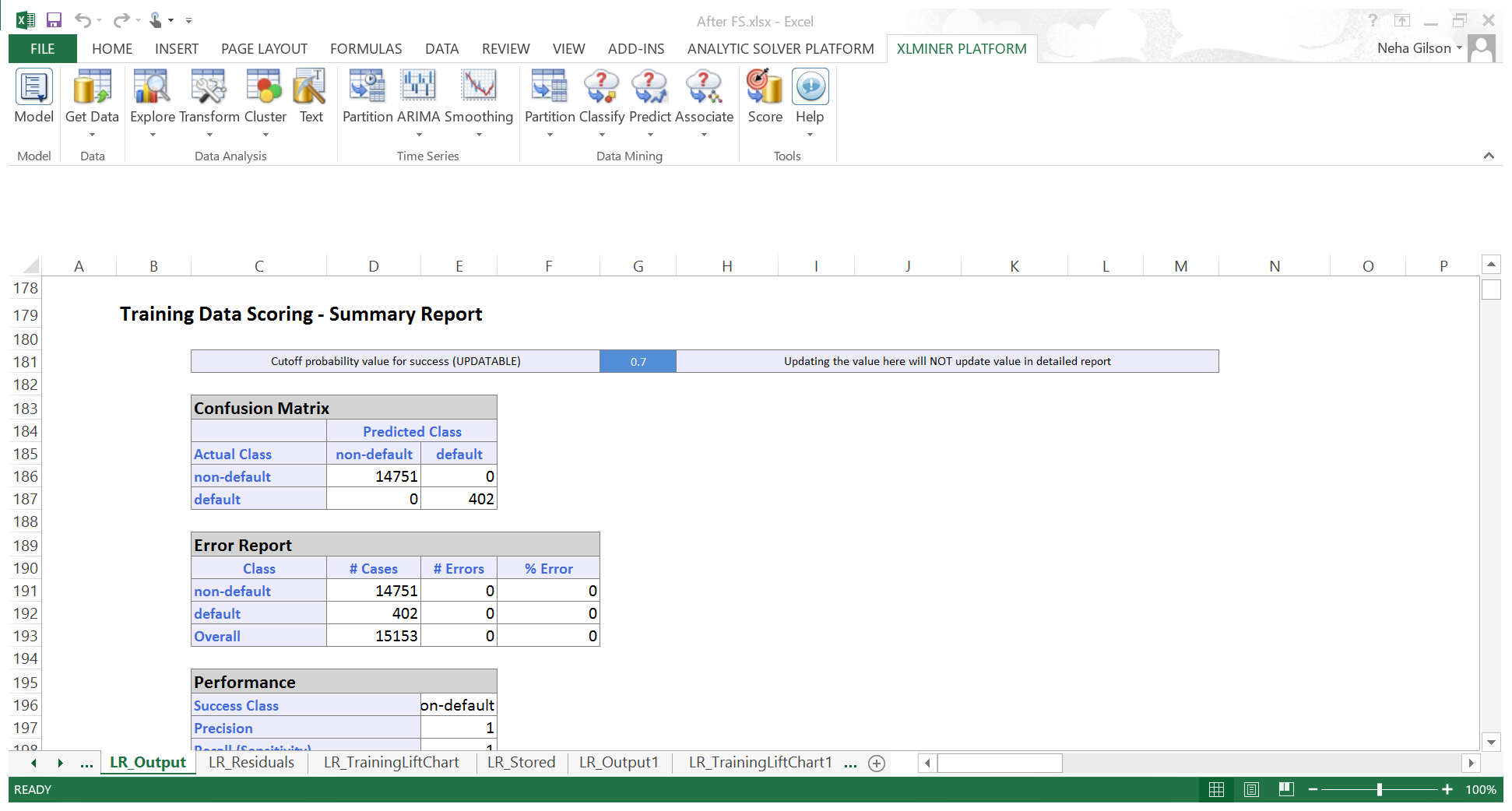


**Logistic Regression:**

When not considering status:

****

When considering status:



## Conclusion:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Logistic | Cart | Random Forest |
| Overall % Error | 2.63 | 0 | 2.7 |
| AUC | 0.73 | 1 | 0.64 |
| Cutoff Value | 0.7 | 0.75 | 0.7 |

The Random Forest Model gives a better classification of mortgage that will fall under non-default. But for classifying a default mortgage it is not a good model. But, if a mortgage falls under non-default and it is still in active status then we cannot say whether it will default in the future. So, Random Forest is not a very good classification model in this case.

When using status variable, CART and Logistic regression models give a perfect model to predict both the default and non-default mortgages. But since this variable is very closely related to the outcome in the sense that it is the binary variation of status, we chose to ignore this feature. When doing so, CART fails to develop a regression tree. Therefore, we conclude that Logistic regression gives a better model to predict the classification of mortgage into defaulters and non-defaulters.

Based on the confusion matrix and the AUC of ROC curve we come to a conclusion that Logistic regression is a better option.

**Detecting Spam**

## About the data:

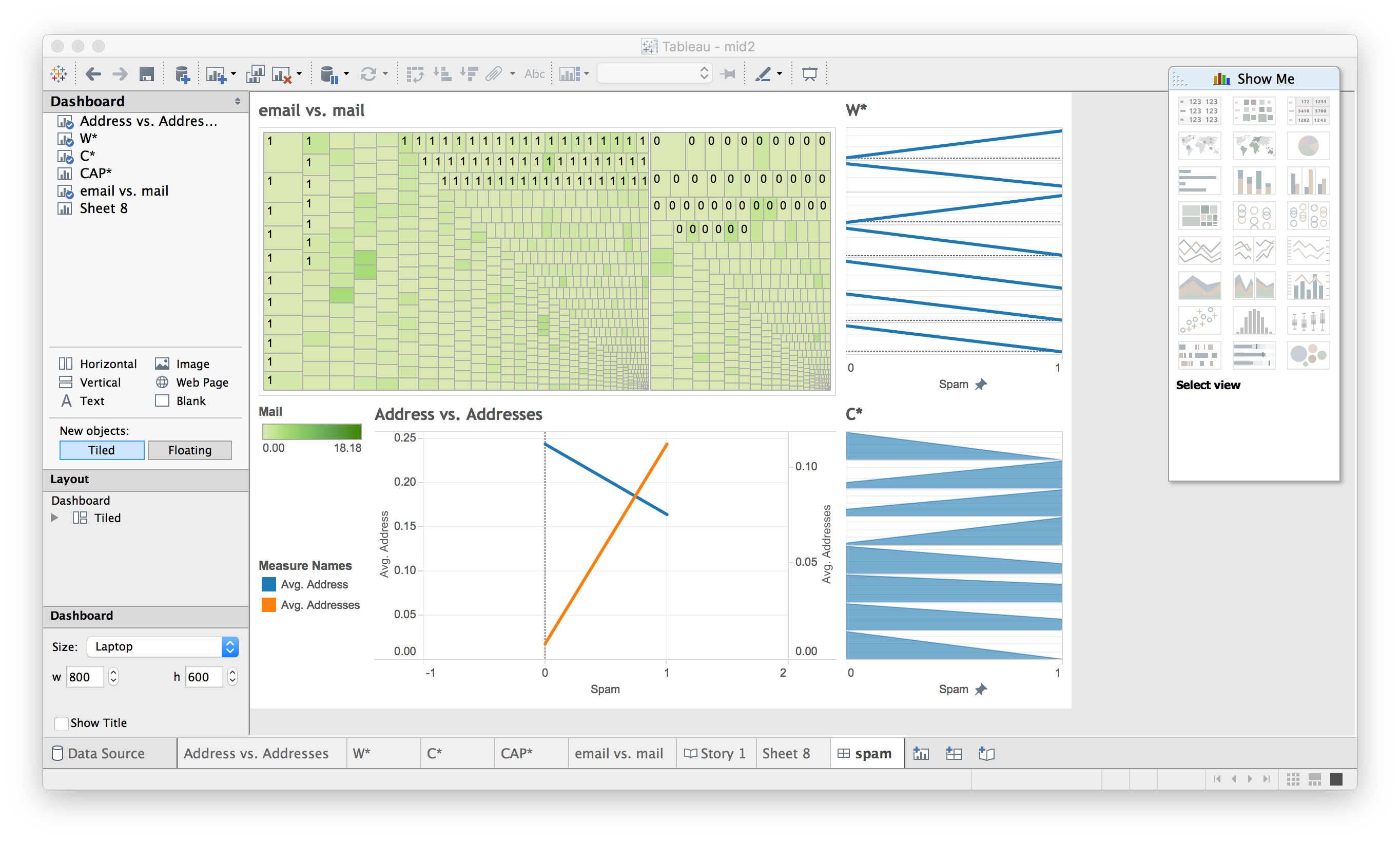
In the spam dataset, there are 57 input variables and the output variable is spam which has binary value for indicating whether an email is spam or not. Among the 57 input variables, we made assumptions that some of them can be group together to compare how they impact the value of spam.

## Goal:

Building a classification model to classify an email into spam or normal email.

## Data Exploration:

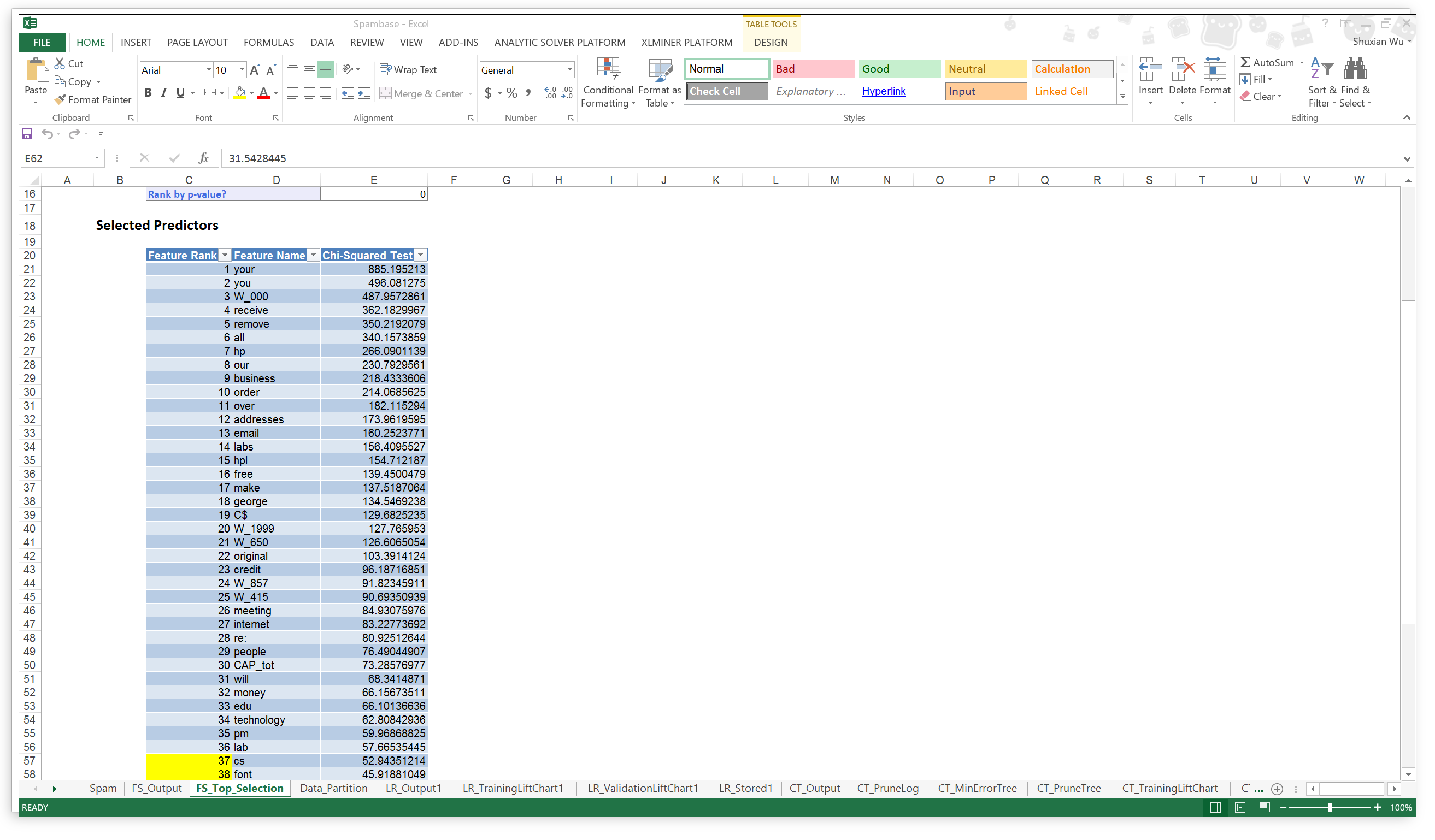
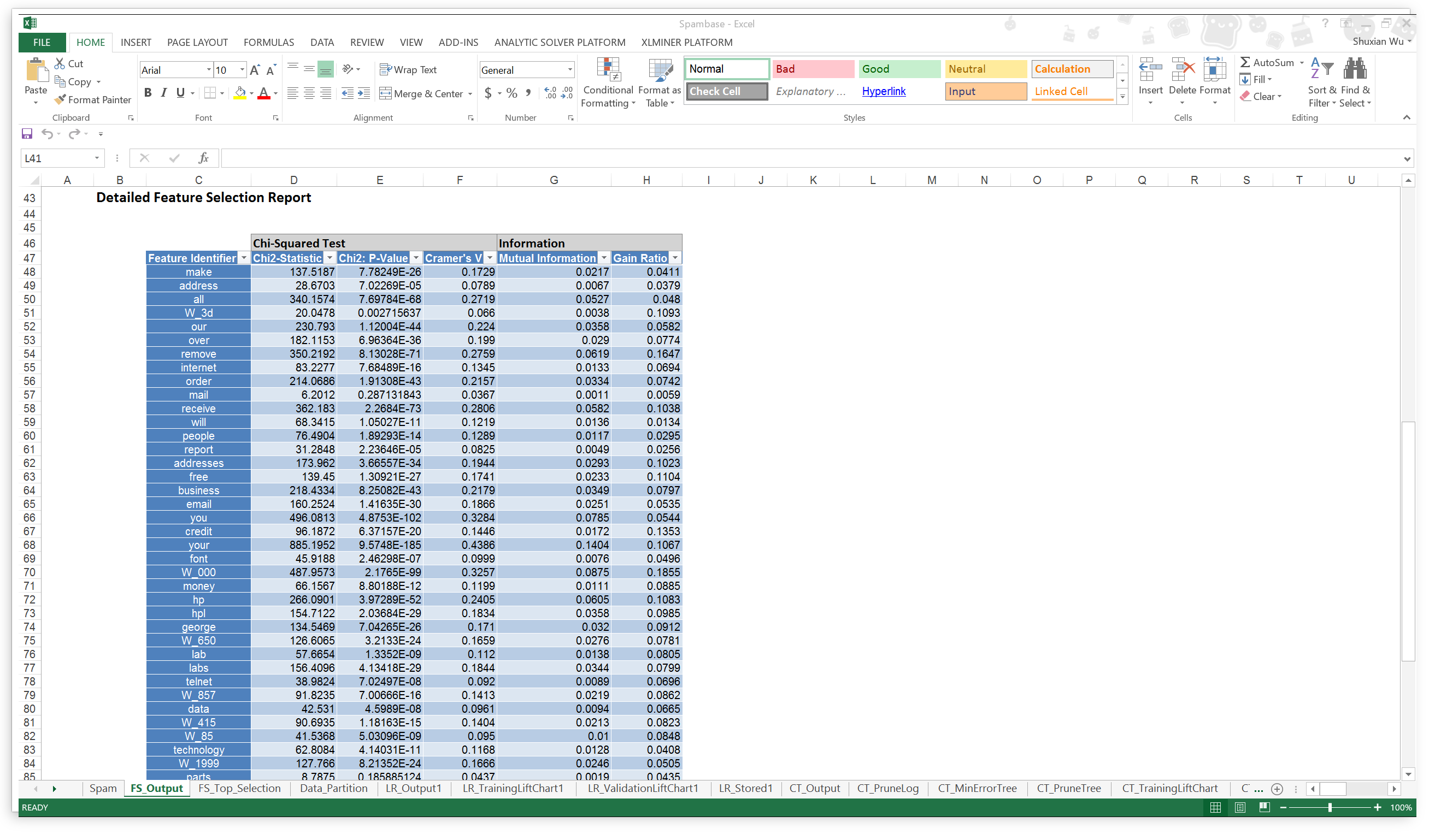
Since there are so many combinations of the input variables, we decided only some parts of them as examples in our report.



* email vs. mail: The graph indicates the relationship among variable email, mail and spam. We found that mail is less significant impact on spam and email tends to be more influential.
* W\*: Grouping variables starting with “W” following by numbers and characters, we compared the average of each variable so that we are able to assume the first and the third variables in the chart has higher significant level in classification model.
* Address vs. Addresses: Comparing address (blue line) and addresses (orange line), addresses has more records on indicating a spam email.
* C\*: Similar to W\* chart, we grouped the variables stating with “C” following by special symbols. We might use the 2nd, 3rd, 4th variables in our model.

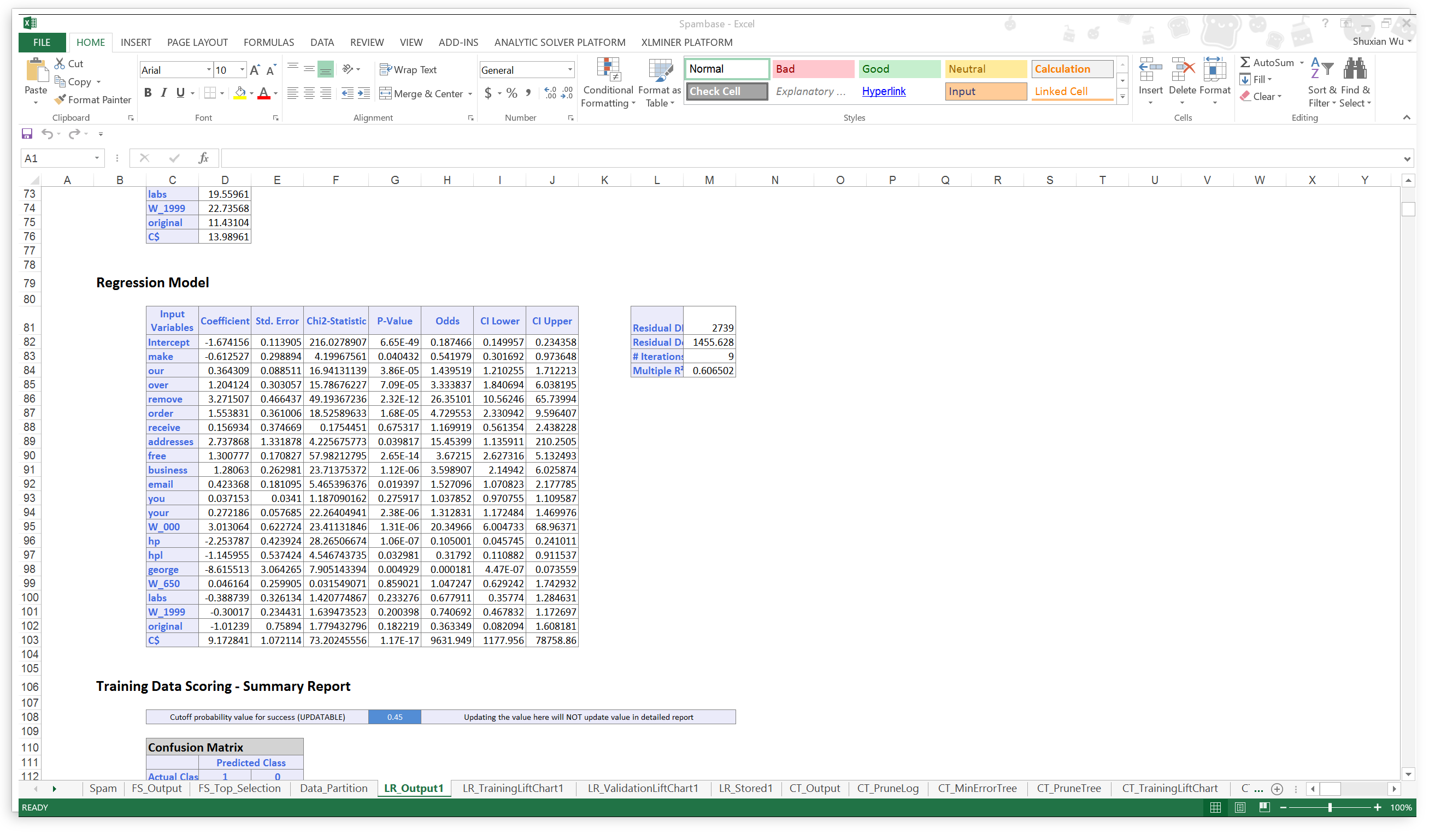
## Classification Model:

To test our assumption, we started with Feature Selection:



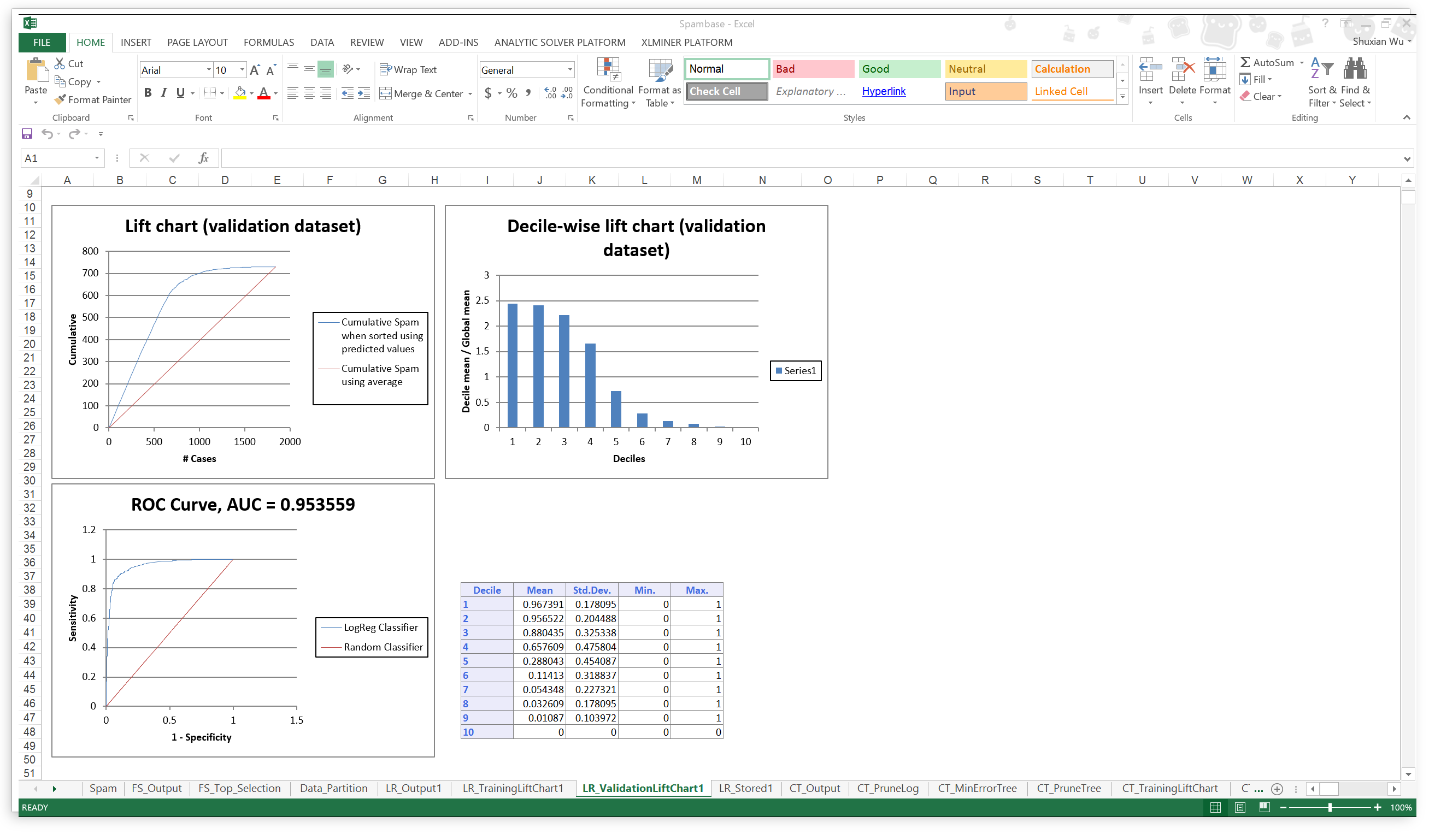
For our model, we decided to consider only top 22 out of 57 which has Chi-Squared Test value larger than 100.

**Logistic Regression:**



**R2=0.61**

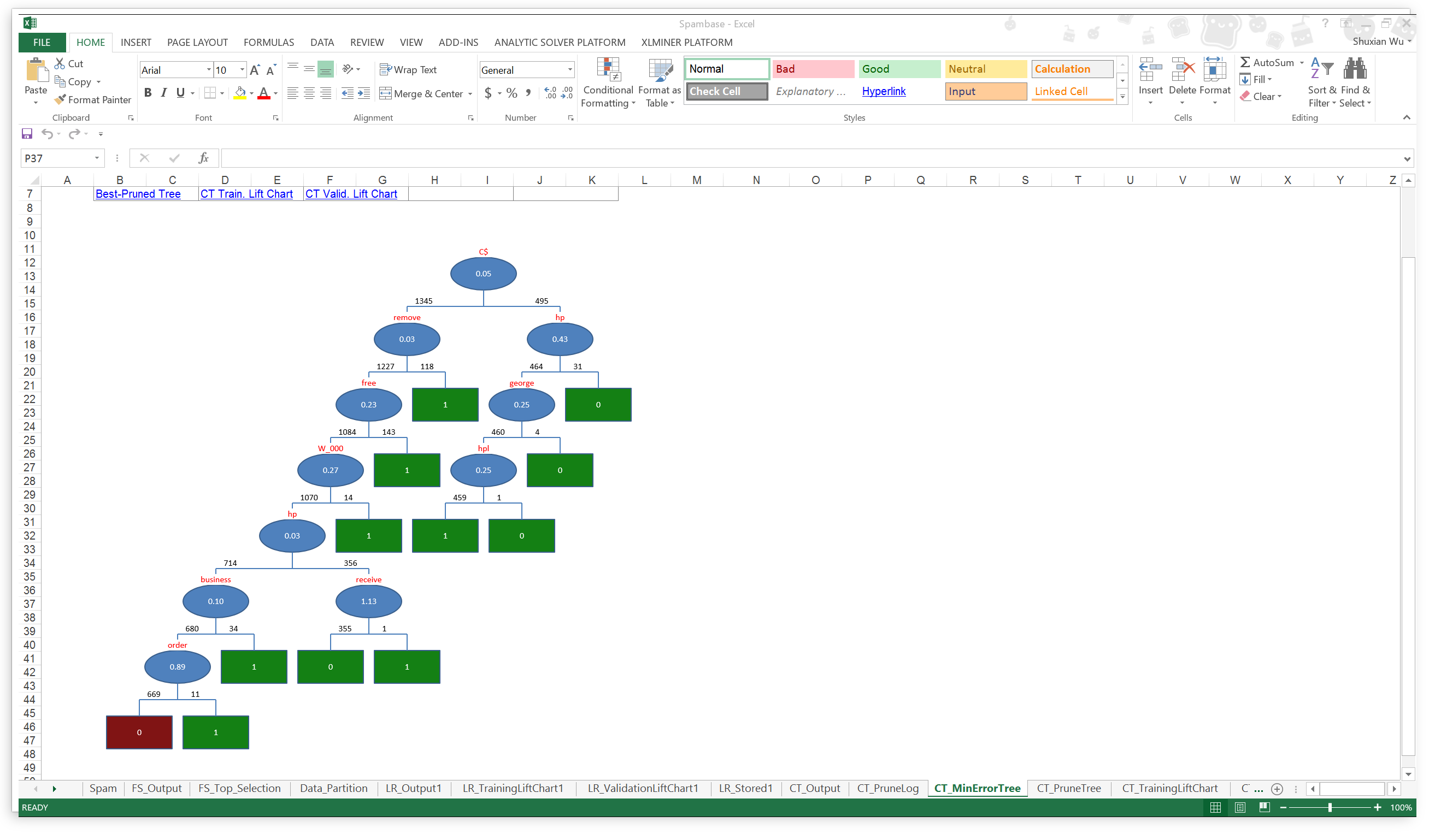
Performance Evaluation:



Under cutoff value of 0.45, we get lowest overall % Error and highest AUC which is close to 1.

**Cart:**

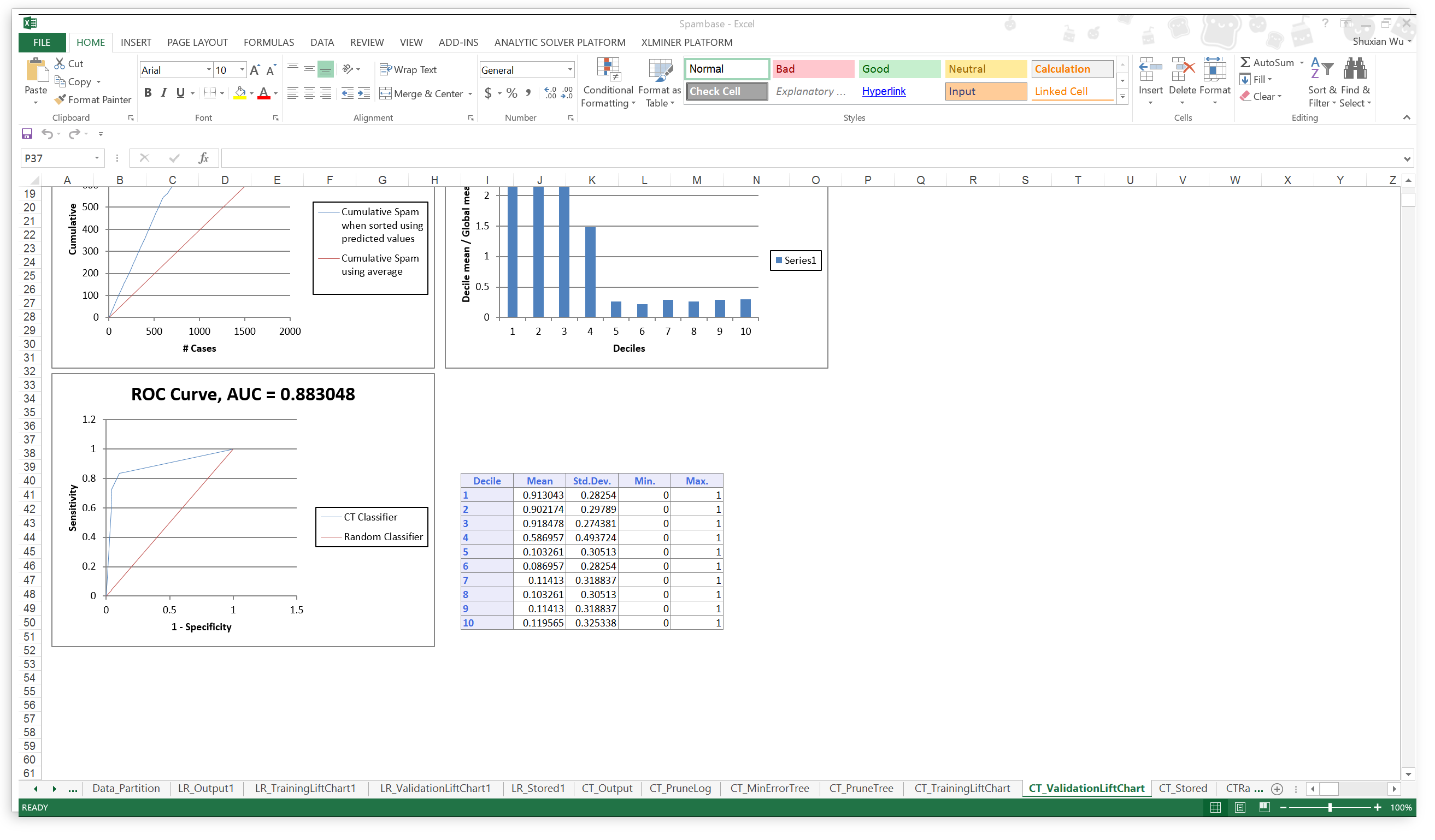
Minimum Error Tree:



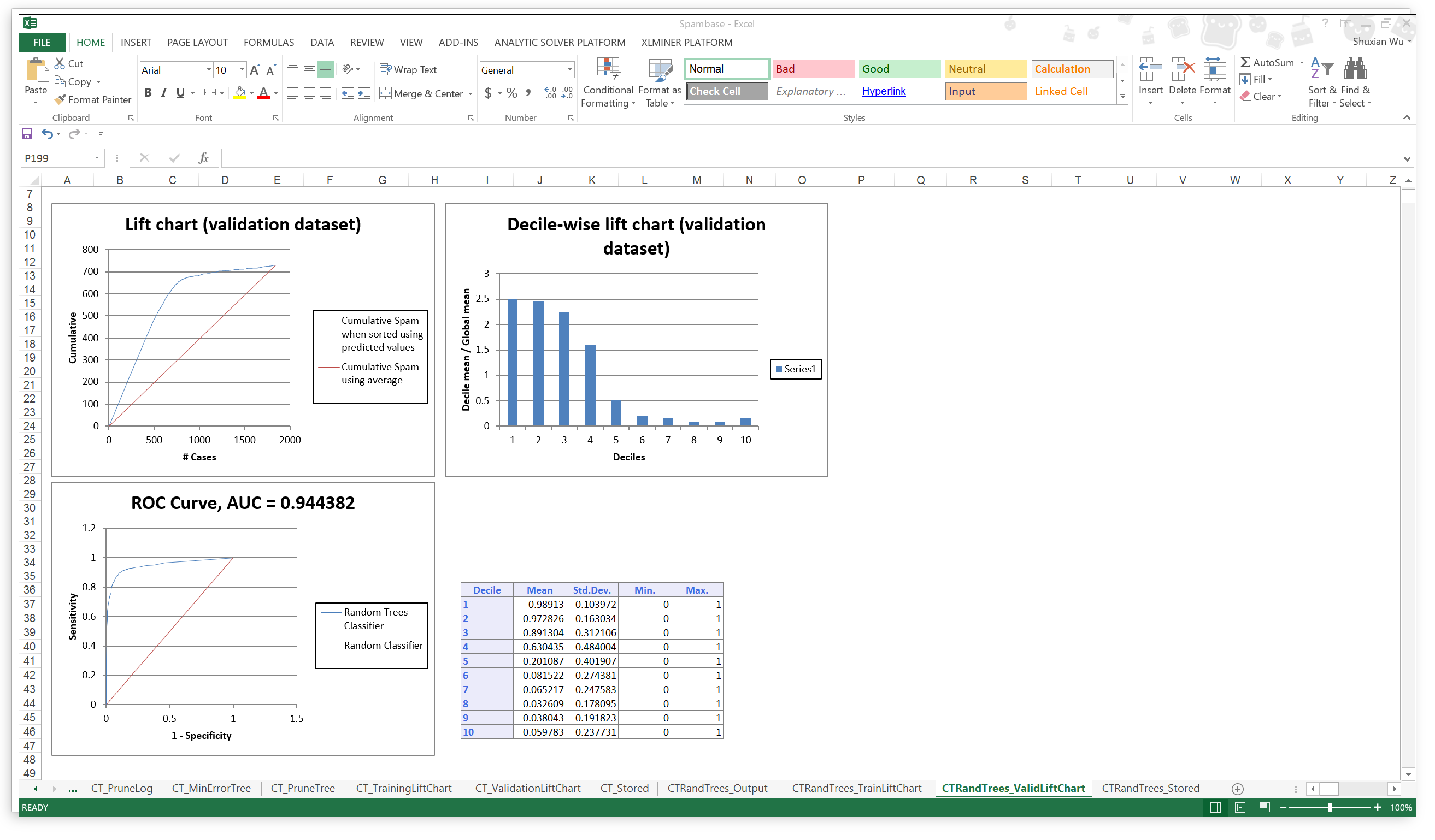
Best Pruned Tree:



Performance Evaluation:



**Random Forest:**



Performance Evaluation Comparison:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Logistic | Cart | Random Forest |
| Overall % Error | 9.89 | 12.35 | 10 |
| AUC | 0.9536 | 0.8830 | 0.9443 |
| Cutoff Value | 0.45 | 0.75 | 0.5 |

Conclusion:

Comparing the three value of each model listed in the table above, Logistic has the lowest Overall % Error and Highest AUC value, so that we think logistic regression is probably the best model to classify if an email is spam or not under certain selection of input variables.

**Blog feedback problem**

## About the data

Every dataset contains 281 columns. The rows represents different blog posts.

As mentioned in the question, we have to consider only the blog posts which were published 72 hours before the basetime.

Columns 263-269 are the binary indicator for the weekday of the basetime and columns 272-276 represent the binary indicator for the weekday of the date of publication.

So we make combinations of basetime and date of publication such that each combination has different of 3 days that is 72 hours.

The combinations of columns are as follows:

Both date of publication and basetime columns given in the data are weekdays.

|  |  |
| --- | --- |
| **Date of Publication of Blog Post** | **Base Time** |
| Column 275 [Saturday] | Column 263 [Monday] |
| Column 276 [Sunday] | Column 264 [Tuesday] |
| Column 270 [Monday] | Column 265 [Wednesday] |
| Column 271 [Tuesday] | Column 266 [Thursday] |
| Column 272 [Wednesday] | Column 267 [Friday] |
| Column 273 [Thursday] | Column 268 [Saturday] |
| Column 274 [Friday] | Column 269 [Sunday] |

* We make a separate model for each combination of *Date of Publication* & *Base Time*.
* There are Test datasets for each day of February 2012 and March 2012.
* Fit the model and predict the values for the corresponding Test dataset.
* Training dataset will be trained for each of the above combinations.
* For Each Combination we take only those blogs which have Date of publication- Basetime=72 hours

For example: Model trained for combination 270-265 will be applied to predict values on Test dataset with date 2012/03/28 which has a basetime of Wednesday. (All values of column 265 are 1) .

## Prediction Model

**Refer to the Example below:**

*# for Wednesday dataset 2012/3/28-Wednesday*

Command 1: testdatawednesday1<-subset(test.dataset.wednesday1, var265==1 & var270==1)

Command 2: testdatawednesday1

Command 3: data4<-subset(train.dataset, var270==1 & var265==1)

We built multiple linear regression model and tested it for various days using various input parameters. The combinations of input parameters are as follows:

* **Basic Features**
* **Basic+Weekday**
* **Basic+Parent**
* **Basic+Textual**
* **Bagging**

*The best one we found was Regression with Bagging*

# regression with bagging

length\_divisor<-2

iterations<-100

predictions<-foreach(m=1:iterations,.combine=cbind) %do% {

training\_positions <- sample(nrow(data2), size=floor((nrow(data2)/length\_divisor)))

train\_pos<-1:nrow(data2) %in% training\_positions

lm\_fit<-lm(var281 ~ var52+var57+var53+var58+var54+var59+var55+var60+var277+var278+var279+var280,data=data2[train\_pos,])

predict(lm\_fit,newdata=testdatafriday)

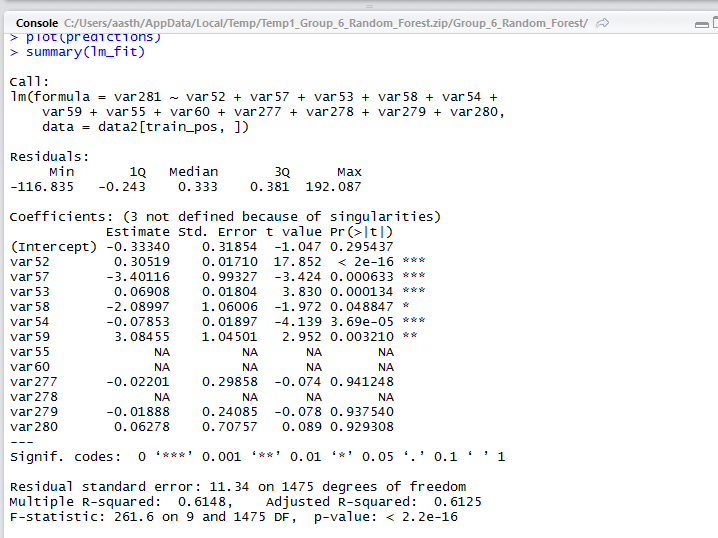
}

predictions<-rowMeans(predictions)

error<-sqrt((sum((testdatafriday$var281-predictions)^2))/nrow(testdatafriday))

RMSE.rtree =sqrt(mean((predictions - testdatafriday$var281)^2))

summary(lm\_fit)



**RMSE Error came out to be 8.12**

**Experiment 2: CART Regression**

We built CART Regression model and tested on the same dataset.

Refer to the example below:

trainset1<-subset(train.dataset, var267==1 & var272==1)

frmla =var281 ~ var52+var57+var53+var58+var54+var59+var55+var60

fit = rpart(frmla, method="anova", data=trainset1)

printcp(fit) # display the results

plotcp(fit) # visualize cross-validation results

summary(fit)

pred=predict(fit,testdatafriday)

summary(pred)

RMSE.rtree =sqrt(mean((pred - testdatafriday$var281)^2))

RMSE.rtree

MAE.rtree <- mean(abs(pred-testdatafriday$var281))

MAE.rtree

RMSE Error came out to be 8.10

**Experiment 3: Random Forest**

#randomforest

fitted <- randomForest(var281 ~ var52+var57+var53+var58+var54+var59+var55+var60, trainset1)

predicted= predict(fit,testdatafriday)

RMSE Error came out to be 17.32

**Evaluationg Model’s Performance:**

RMSE Errors Table:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Bagging(Regression) | Cart Decision Tree | Random Forest |
| Saturday 2012/03/31 | 28.70281 | 22.179 | 22.179 |
| Wednesday 2012/03/28 | 8.10 | 8.12 | 17.32 |
| Wednesday 2012/02/29 | 13.711 | 13.73 | 15.62 |
| Friday 2012/3/30 | 11.82 | 12.32 | 9.32 |

We can clearly see that Regression used with Bagging gives the minimum RMSE Errors.

Also the R^2 and adjusted R^2 values for Regression Bagging are high which means the predicted values are close to the real ones.

The T values for Bagging shows the used features as the most significant ones in this model

**Recommendation:**

After trying these 3 models on all the datasets for different days of the week, we found that the model with best performance was Regression with bagging.

*Reason:* Regression with bagging performs best because it takes 100 randomly selected subsets of the basic features and predicted values for all of these 100 subsets of features.

The bagging model takes the average of these 100 prediction values. Hence bagging model accounts for the maximum variance of information included for prediction.

We found the RMSE Error values to be lowest for the regression model used with Bagging.

*Hence we highly recommend this model.*