**FINAL PROJECT**

**OPIM 5604 – FALL, 2016**

**University of Connecticut – School of Business**

**“Predicting the Alertness of the driver – Ford Dataset”**

**“The work contained and presented here is our work and our work alone”**

**TEAM #7**

**CLASS SECTION – EVENING**

**TEAM MEMBERS:**

Srishyla, Ashwini

Vadrevu, Pranavanshu

Mutnuru, Muralidhar

Saha, Debaleena

Tennur Narayanan, Tejas Naren

# **Executive Summary**

## **Problem Statement**

* To develop insights on the alertness of a driver from Ford dataset which can be used to reduce the number road accidents. Alertness of a driver is an important factor which makes the roads safer to ride. Driving while being distracted, fatigued or drowsy may lead to accidents.
* The Insights generated from the model can be used by the taxi companies to evaluate their drivers. The insurance companies can categorize the premium based on the alertness of the driver. The insights primarily include the finding of parameters contributing to increase or decrease the alertness of the driver.

## **Modelling Approach**

* The approach includes cleaning of the data which would primarily include screening of the dataset for missing patterns and outliers. The dataset would be categorized into two. One with outliers and another without outliers.
* A categorical parameter along with a continuous parameter would be predicted from developing models for each of the dataset. The categorical parameter would essentially contain values of “Yes” and “No” while the continuous parameter provides a value for the “Accident Likelihood”.

## **Challenges & Learning**

* The dataset had columns without names and the absence of data dictionary was the primary challenge in the modelling as we could not rely on the physical intuition.
* Considering, the fact that, each member had his/her own perspective of the data, it was difficult to collaborate the techniques and standards used by each member leading to a single output.
* We pondered hard on whether or not outliers should be removed, ultimately, we modelled on both datasets with and without outliers.
* Model evaluation criterion required multiple iterations to finalize.

## **Results**

The modeling carried out on the dataset yielded few insights on the factors contributing to the alertness of the driver. The insights primarily included the prediction of alertness which is a categorical response and the Accident Likelihood, a continuous response. The Business can use the models recommended by this report to understand the factors contributing to the alertness of the driver in detail.

**Table of Contents**

[**Executive Summary** 2](#_Toc468210237)

[Problem Statement 2](#_Toc468210238)

[Modelling Approach 2](#_Toc468210239)

[Challenges & Learning 2](#_Toc468210240)

[Results 2](#_Toc468210241)

[**1.0 Dataset description** 6](#_Toc468210242)

[**2.0 Data preprocessing** 7](#_Toc468210243)

[2.1 Data Cleaning and Sampling 7](#_Toc468210244)

[2.2 Data Exploration and Modification 9](#_Toc468210245)

[2.3 Distribution Analysis 10](#_Toc468210246)

[2.4 Dimensionality Reduction 12](#_Toc468210247)

[2.5 Exploring Outliers 14](#_Toc468210248)

[2.6 Data Scaling 15](#_Toc468210249)

[**3.0 Model Selection Process** 16](#_Toc468210250)

[3.1 Model Selection Process Flow 18](#_Toc468210251)

[3.2 Model Evaluation Criteria and Model Selection Matrix 18](#_Toc468210252)

[**4.0 Model Analysis of Continuous Response (with Outliers)** 21](#_Toc468210253)

[4.1 Model Tweaking 22](#_Toc468210254)

[4.2 Least Squares 22](#_Toc468210255)

[4.2.1 Observations 22](#_Toc468210256)

[4.2.2 Inference 26](#_Toc468210257)

[4.3 Stepwise 27](#_Toc468210258)

[4.3.1 Observations 27](#_Toc468210259)

[4.3.2 Inference 29](#_Toc468210260)

[4.4 Neural Network 29](#_Toc468210261)

[4.4.1 Observations 30](#_Toc468210262)

[4.4.2 Inferences 34](#_Toc468210263)

[4.5 Boosted Tree 34](#_Toc468210264)

[4.5.1 Observations 35](#_Toc468210265)

[4.5.2 Inference 35](#_Toc468210266)

[4.6 Ensemble Model 36](#_Toc468210267)

[4.6.1 Observation 36](#_Toc468210268)

[4.6.2 Inference 37](#_Toc468210269)

[4.7 Final Evaluation 38](#_Toc468210270)

[4.7.1 Recommendation – Continuous Response 39](#_Toc468210271)

[**5.0 Model Analysis of Categorical Response (Without Outliers)** 40](#_Toc468210272)

[5.1 Model Tweaking 40](#_Toc468210273)

[5.2 Decision Tree 40](#_Toc468210274)

[5.2.1 Observation 40](#_Toc468210275)

[5.2.2 Inference 43](#_Toc468210276)

[5.3 Discriminant Analysis 44](#_Toc468210277)

[5.3.1 Observation 44](#_Toc468210278)

[5.3.2 Inference 46](#_Toc468210279)

[5.4 Neural Network 46](#_Toc468210280)

[5.4.1 Observation 46](#_Toc468210281)

[5.4.2 Inference 49](#_Toc468210282)

[5.5 Nominal Logistic Regression 50](#_Toc468210283)

[5.5.1 Observation 50](#_Toc468210284)

[5.5.2 Inference 53](#_Toc468210285)

[5.6 Ensemble Model 1 54](#_Toc468210286)

[5.6.1 Observation 54](#_Toc468210287)

[5.6.2 Inference 56](#_Toc468210288)

[5.8 Ensemble Model 3 62](#_Toc468210291)

[5.8.1 Observation 62](#_Toc468210292)

[5.8.2 Inference 64](#_Toc468210293)

[5.9 Ensemble Model 4 66](#_Toc468210294)

[5.9.1 Observation 66](#_Toc468210295)

[5.9.2 Inference 67](#_Toc468210296)

[5.10 Final Evaluation 68](#_Toc468210297)

[5.10.1 Recommendation – Categorical Response 69](#_Toc468210298)

[**6.0 Data Visualization and Insights** 70](#_Toc468210299)

[**7.0 Project Conclusion & Recommendation** 72](#_Toc468210300)

[**8.0 Appendix** 73](#_Toc468210301)

# 

# **1.0 Dataset description**

We obtained the data for this problem from Kaggle challenge website. The website provided two data files – one for training and one for testing. The training data file consists of 604,329 observations and the test file contains 120,840 observations from real time scenarios. These are measurements from total of 510 real time driving session where each driving session takes 2 minutes. Both training and test dataset have 32 variables. The following are the parameters or the columns in the dataset:

|  |  |
| --- | --- |
| **Column Header** | **Column Description** |
| P1 to P8 | Physiological data |
| E1 to E11 | Environmental data |
| V1 to V11 | Vehicular data |
| IsAlert | Indicates if the driver is alert (to be predicted) |
| Accident Likelihood | The likelihood value of whether an accident will occur or not (to be predicted) |

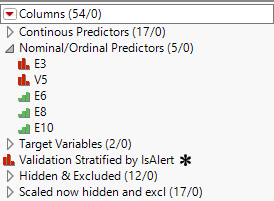
Table 1.1

# **2.0 Data preprocessing**

This section details the activities conducted as part of data preprocessing. The below mentioned are the steps involved in the preprocessing activity.

* Data cleaning and sampling
* Univariate analysis
* Bivariate and Multivariate analysis
* Dimensionality Reduction

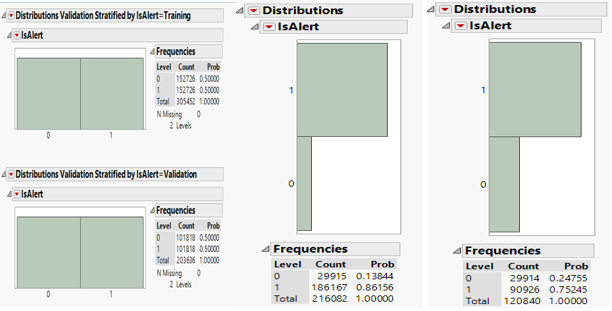
## **2.1 Data Cleaning and Sampling**

In some rare scenarios JMP misclassifies the data type of the variables. In our project, the outcome variable ‘IsAlert’ was changed from continuous to nominal. Since we didn’t have the data dictionary to explain any other variables, we used ‘recode’ to find what data type the variable might belong to. We were easily able to classify all the data types except variables E3, E6, E8, E10 and V5. We decided while modelling we will input them as nominal/ordinal predictors and compare results with, when we used them as continuous predictors and shall decide the datatype based on whichever was giving us the best results.

*Figure 1.1*

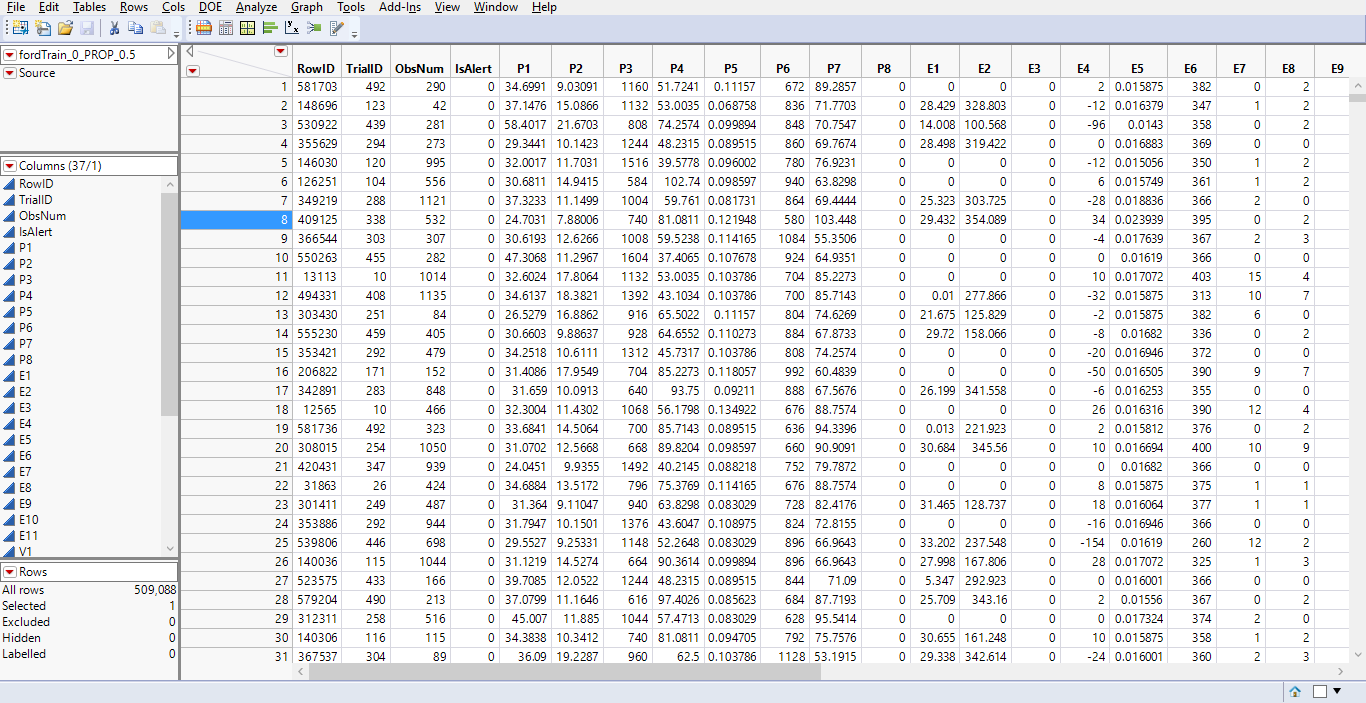
This section illustrates the activities performed as a part of data sampling. The following enlisted are the activities:

1. The given dataset at source was only Training, which we classified into two groups of data namely Training and Validation. This was done by “Stratified Balance Split” with a value 0.5 for both training and validation respectively. Since we had limited number of 0s in the ‘IsAlert’ column, the team decided to go with stratified sampling of Training and Validation rather than random sampling of the data.



*Figure 1.2*

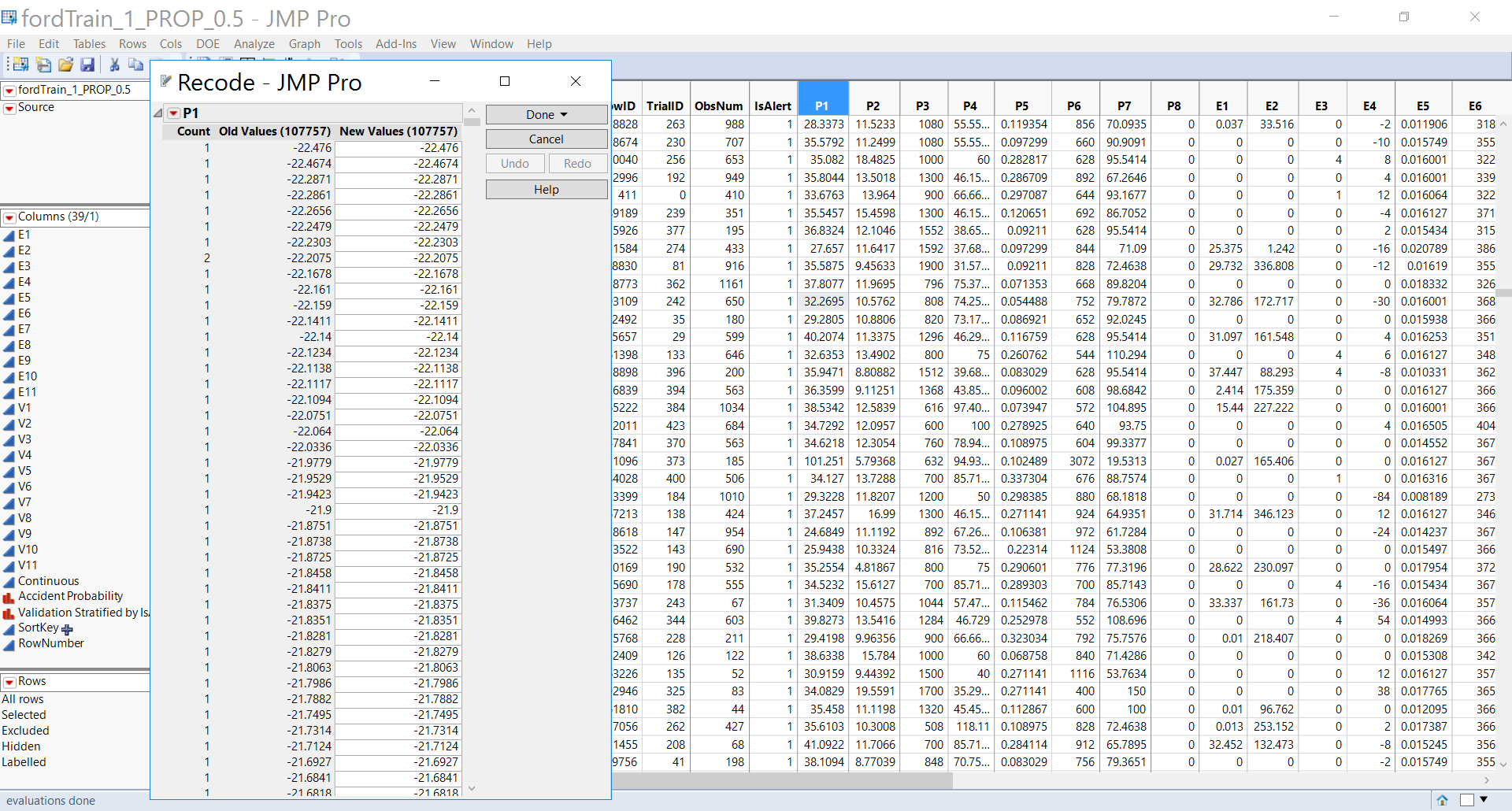
1. The stratified split had left out 95,241 Rows, which we appended to the test data set, such that there is no data loss. Thus, the test data count increased from 120840 to 216082.
2. The test dataset, obtained as a separate file from the source, was appended to the train and validation dataset and a single whole dataset was thus created with 725,170 observations.



*Figure 1.3*

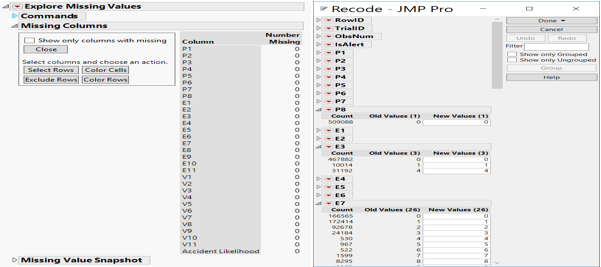
## **2.2 Data Exploration and Modification**

This section illustrates the data exploration and modification process conducted on the dataset. The Exploration part included the finding of missing patterns in each fields of the dataset. The following screenshot illustrates the process involved in the identification of missing values in the dataset with respect to a particular field.



*Figure 1.4*

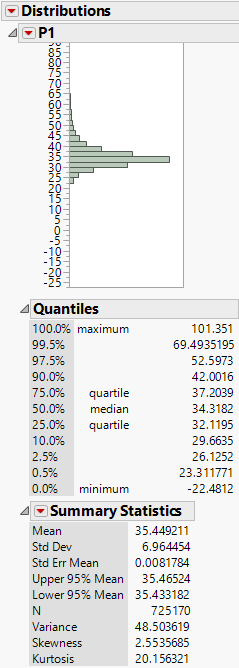
The process indicated above was conducted on all the fields in the dataset to explore the missing patterns and irregular nomenclature. From our analyses, we found that our data had no missing patterns, N/A’s or NA, any junk values or spelling errors.



*Figure 1.5*

## **2.3 Distribution Analysis**

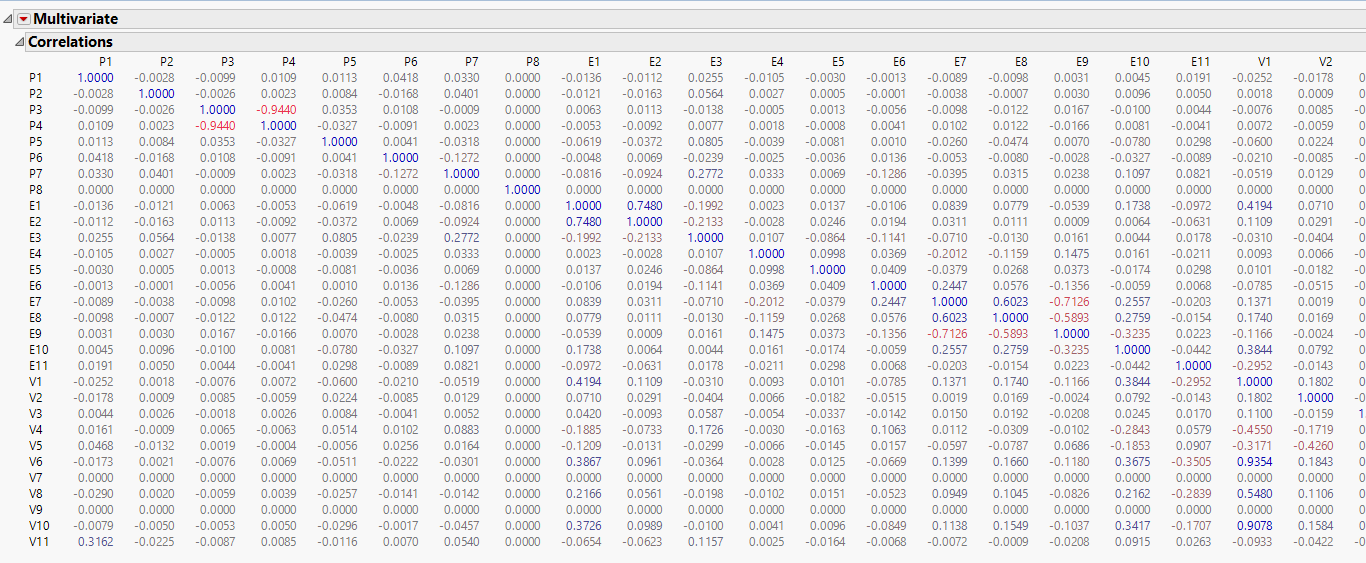
The Distribution of the data was checked to find any interesting patterns in the skewness levels. The following figure details the distribution of data for the “P1” column.



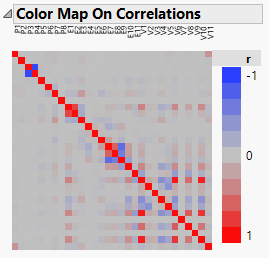
*Figure 1.6*

The above figure throws light on the skewness, kurtosis, Mean, Median, Max value and Min Value of the data with respect to the column “P1”. There are no outliers in this field. The same process is repeated for all the columns to review the central tendency measures of the dataset. But since we did not have the data dictionary from the business, we decided to leave the distributions untouched as we do not know the significance of these variables and their skewness.

## **2.4 Dimensionality Reduction**

With the total number of measured features being 33, it was desirable to reduce this number significantly. We began by inspecting the variables for correlations and non-contributing variables, i.e. variables that do not change value over the whole dataset or do not add any predictive power to the data.

*Figure 1.7*

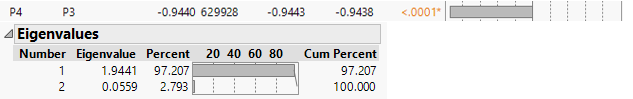
*Figure 1.8*

As can be seen from the above analyses, there were few variables which had very high correlation between them. Out of two variables which had high correlation, we took a PCA of both, since we could not decide which to keep and which to discard as the carriable names were not given.



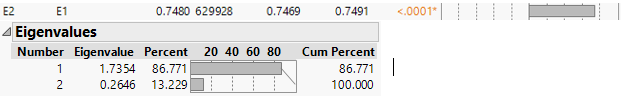
*Figure 1.9*

**PrinP4P3**



*Figure 1.10*

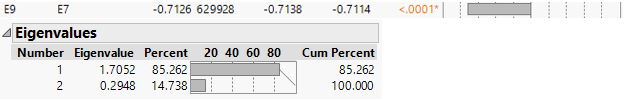
**PrinE2E1**



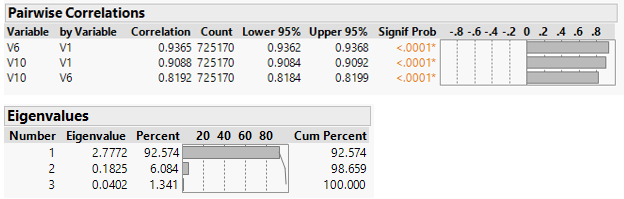
*Figure 1.11*

**PrinE9E7**

*Figure 1.12*

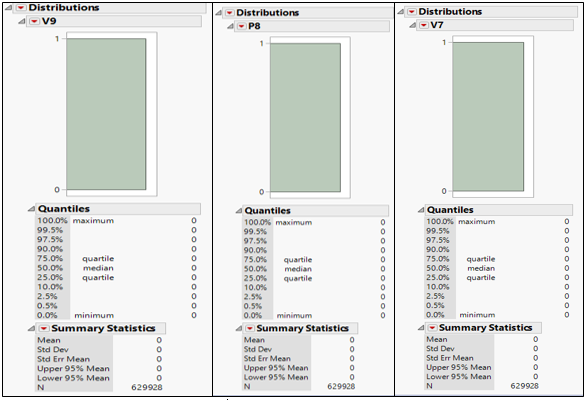


**PrinV1610**



*Figure 1.13*

V1, V6, and V10 were then discarded and PrinV1610 was included as a predictor variable.



*Figure 1.14*

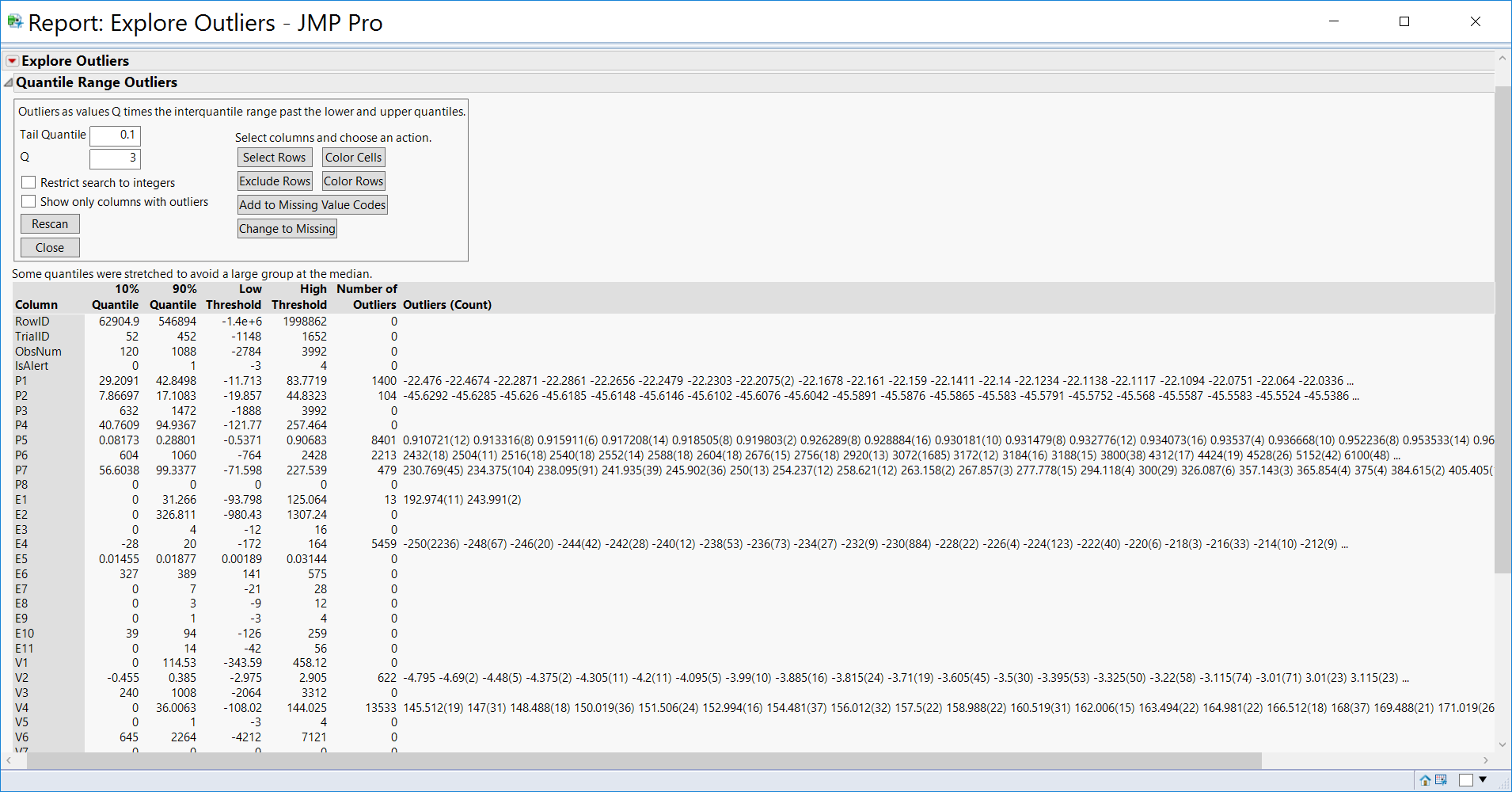
P8, V7 and V9 have no variation whatsoever and cannot provide any predictive power to the data. Hence, they were excluded.

## **2.5 Exploring Outliers**

Before we proceed to exploring outliers, we would separate out the test and validation dataset from the training dataset. Such that we know how well our model performs when it encounters outliers in test and validation datasets and is more representative of the real-world scenario.

A series of activities are performed to identify outliers in each of the columns. Outlier Box plots or Quantile range outliers is used to identify the outliers. The following figure projects the outliers in the dataset with respect to each column along with its range. After much deliberation, the team decided to do the following.

1. Create two separate datasets one with the outliers and one without the outliers.
2. At the time of modelling we would run our model on both data sets and see which one performs better and choose the best.



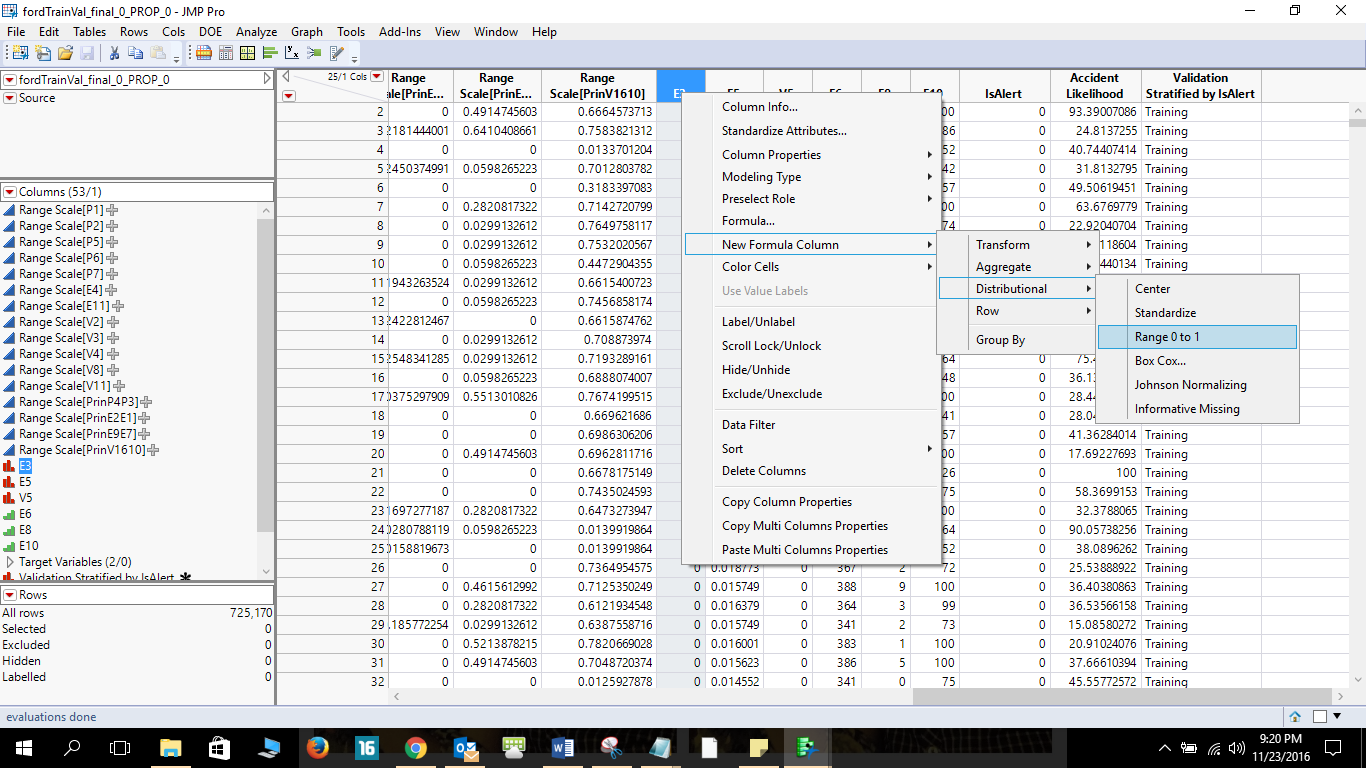
*Figure 1.15*

Post this, we append the test and validation datasets back to the two datasets created above, to have two modelling datasets.

There were in all 17748 outliers in the training dataset. The final dataset with outliers excluded now has 287704 training elements and the one with outliers included has 305452 training elements.

## **2.6 Data Scaling**

All the continuous columns in both the data sets are scaled between 0 and 1. Such that while building a model larger weights aren’t given to estimates. To avoid this problem.



*Figure 1.16*

# 

# **3.0 Model Selection Process**

This section details the activities carried out as part of data modeling. Models would be developed for two datasets, one with outliers and another without outliers. The prediction was done for two parameters. One is the “Accident Likelihood” which is a continuous variable and the other is a categorical variable which is “IsAlert”. The following matrix provides an overview on the models which were developed as part of this activity.

|  |  |  |
| --- | --- | --- |
| **Dataset Type** | **Dataset with Outliers** | **Dataset without Outliers** |
| **Dependent Variable Type** |
| **IsAlert (Categorical)** | Partition | Partition |
| Neural Networks | Neural Networks |
| Discriminant Analysis | Discriminant Analysis |
| Nominal Logistic | Nominal Logistic |
| Ensemble Model | Ensemble Model |
| **Accident Likelihood (Continuous)** | Partition | Partition |
| Least Squares | Least Squares |
| Boosted Trees | Neural Networks |
| Neural Networks | Stepwise |
| Stepwise | Ensemble |
|  | Ensemble |  |

*Table 2.1*

The above displayed matrix provides a list of all the models developed in the process of prediction. The step followed to choose the model for prediction are as follows. All the models which are developed and not part of the final model can be found in the appendix.

## **3.1 Model Selection Process Flow**

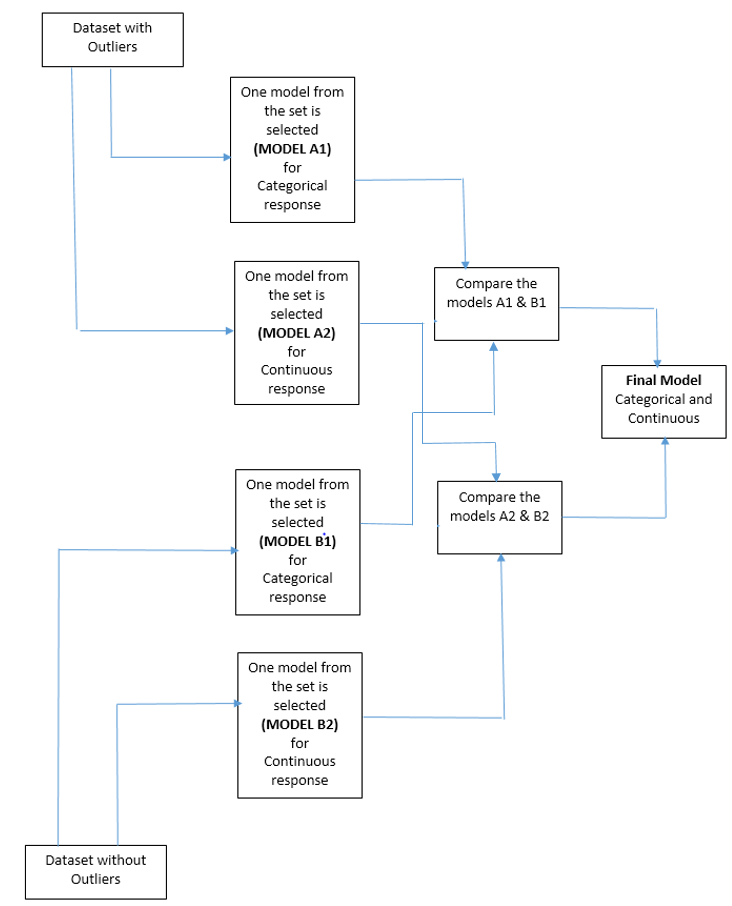


Figure 2.1

## **3.2 Model Evaluation Criteria and Model Selection Matrix**

1. Profit/Cost rates as opposed to each correct/wrong prediction for continuous/categorical.

|  |  |
| --- | --- |
| **Categorical** | **Value (in $)** |
| Benefit of True Positive | +$10.00 |
| Cost of False Positive | -$50.00 |
| Benefit of True Negative | +$50.00 |
| Cost of False Negative | -$10.00 |

|  |  |
| --- | --- |
| **Continuous** | **Value (in $)** |
| Benefit of within threshold predictions | +$7.00 |
| Cost of Out of threshold Errors | -$3.00 |
| Cost of Max/Min Errors | -$1.00 |

*Table 2.2*

**Continuous**

We have set a threshold of +/-5 for each prediction. If the prediction exceeds this threshold there is a cost associated with it. There is also small cost associated with predicting outside of the range of 0-100. We benefit by predicting within range and within threshold.

**Categorical**

It is extremely vital that we predict a non-alert driver as Not Alert, i.e. a true negative as a true negative, as the life of the driver and the safety of passengers is associated with it, hence the +$50 benefit and for predicting the alert as alert a +$10 benefit. Vice-versa applies for cost.

1. Model Evaluation Criteria

Net revenue is a major criterion in both continuous/categorical. RMSE<RMSE(Baseline) has been provided by the business, since the baseline fails to perform on criteria 2 and 3, it is vital that a model passes all 3 criterions for it to be selected. Accuracy of the categorical model should be better than the baseline.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Criterion 1** | **Criterion 2** | **Criterion 3** |
| **Continuous** | Net Revenue | RMSE< RMSE(Baseline) | No Trend in Residual Plot |
| **Categorical** | Net Revenue | Accuracy>Accuracy(Baseline) |  |

*Table 2.3*

1. Baseline Performance

Continuous:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Cost of the model** | **Count of Out of threshold Errors** | **Count of within threshold predictions** | **Count of Max/Min Errors** | **Net Revenue** |
| BASELINE | $0 | 195900 | 20182 | 0 | -$446,426.00 |

*Table 2.4*

Categorical:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Name** | **Cost of the Model** | **Total Profit from the Model** | **Net Revenue from the Model (Profit – Cost)** |
| Baseline Model (All values predicted as 0) | $0 | $10,804,100 | $10,804,100 |
| Baseline Model (All values predicted as 1) | $0 | $2,160,820 | $2,160,820 |

*Table 2.5*

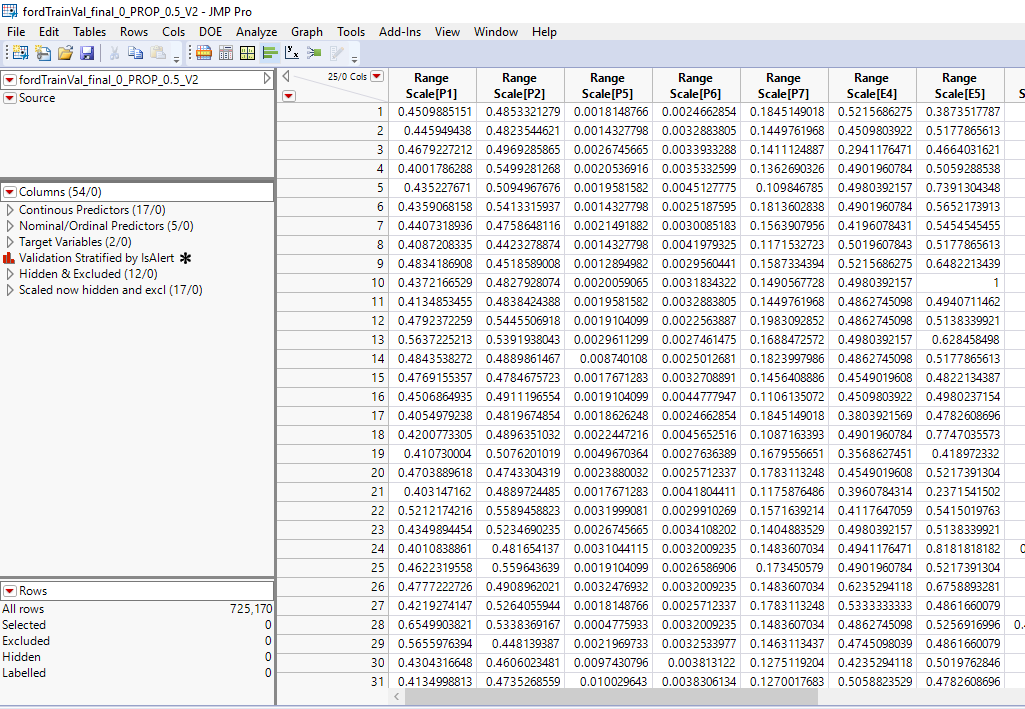
1. Model Selection Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dataset** | **Target Variable** | **Selected Model** | **Net Revenue** | **RMSE** | **Residual Plot** |
| With Outliers | Continuous | Neural Nets | $1,326,832.00 | Acceptable | Acceptable |
| Categorical | Nominal Logistic Regression @ 0.2 cut-off | $17,21,640 | N/A | N/A |
| Without Outliers | Continuous | Neural Nets | $1,320,996.00 | Acceptable | Acceptable |
| Categorical | Nominal Logistic Regression @ 0.3 cut-off | $17,69,050 | N/A | N/A |

*Table 2.6*

We have highlighted the best models, fulfilling our aforementioned criteria. We notice that modelling on two separate datasets with/without outliers has its benefits, and we would have lost out on our best model, if we just got rid of the outliers.

# **4.0 Model Analysis of Continuous Response (with Outliers)**



*Figure 3.1*

We begin with grouping all columns such that, running a model is easier and organized. We are going to run a total of 5 continuous prediction models.

1. Least Squares
2. Stepwise
3. Neural Networks
4. Boosted Tree
5. Ensemble

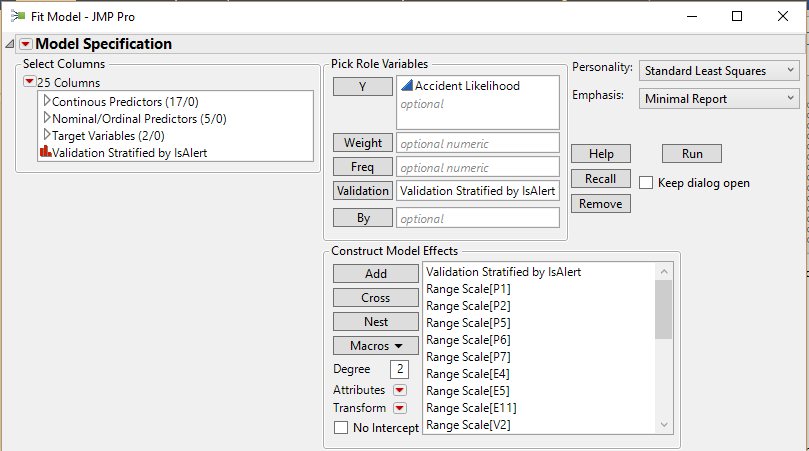
## **4.1 Model Tweaking**

For every continuous model, we do the following tweaks to fine tune our model and run multiple iterations to get better results.

1. Remove the variable from the predictors that are causing singularity. This is only a problem for the least squares model.
2. Choose only the variables that are significant.
3. We use transformations, full factorials, and response surface between variables. Especially between those that are not significant.
4. We consider E3, E6, E8, E10 and V5 to be continuous as well as nominal and see if results improve.

## **4.2 Least Squares**

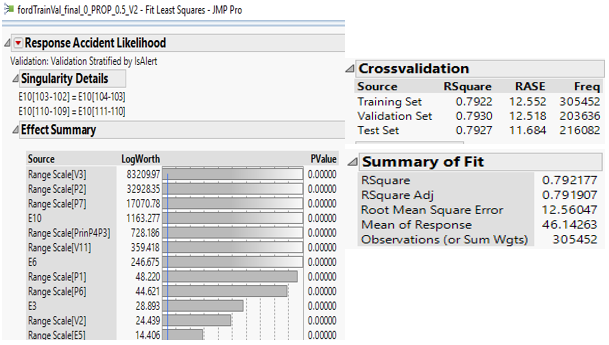
We run the Least squares model with the following parameters. Also, as discussed in the pre-processing step, we are considering variables E3, E6, E8, E10 and V5 as nominal/ordinal predictors and later compare model performance with the time when we use them as continuous predictors.



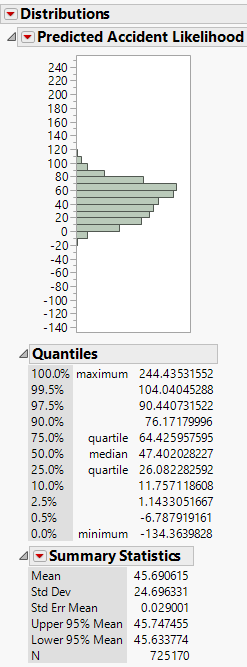
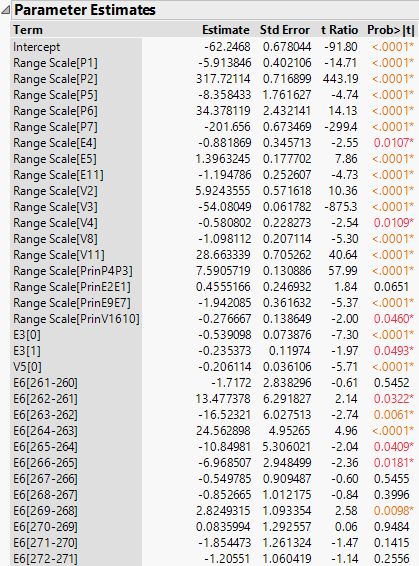
*Figure 3.1*

## **Observations**

We get the following report:



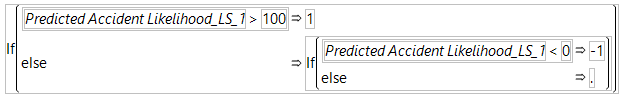
*Figure 3.2*

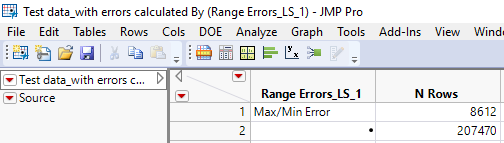


*Figure 3.3*

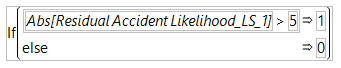
We notice a singularity component here, which happens when one or more of the explanatory variables are linear functions of other explanatory variables.

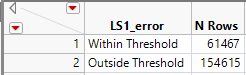
Also, we add a column to identify values outside the range of 0-100 and the ones outside of our threshold of +/- 5. We have attached a summary table highlighting the number of errors.





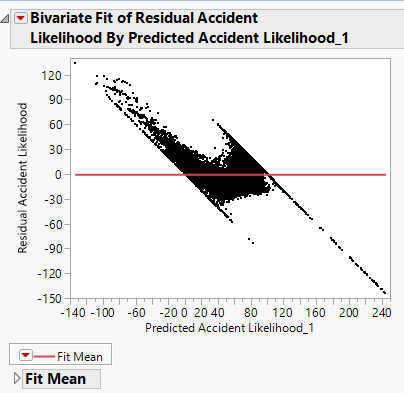
*Figure 3.4*





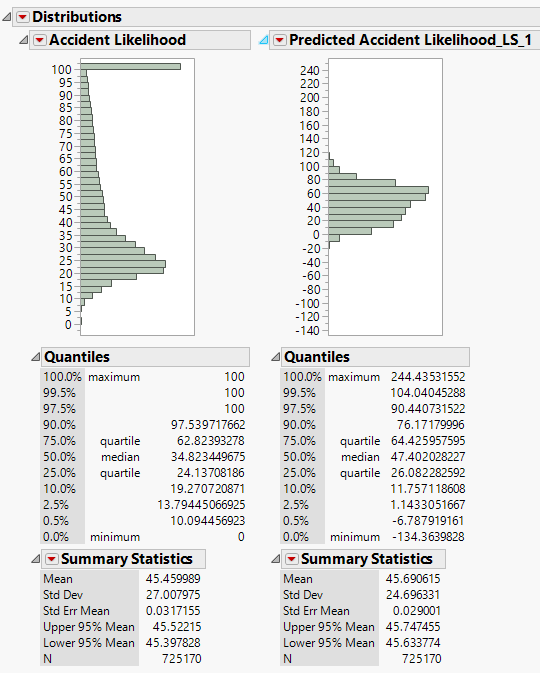
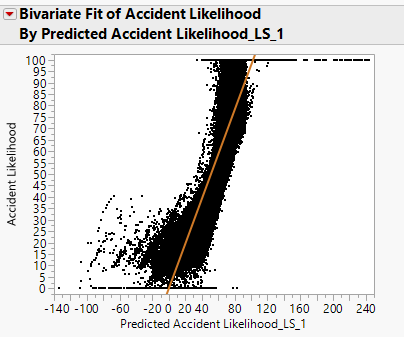
*Figure 3.5*

We also plot the residuals using fit Y by X with the predicted values below. Here we notice a time series data trend that includes a positive trend in the target variable, that is moving up over time, the model underestimates early and overestimate late. We need to be able to build a model that estimates consistently over all time periods. This would have been easier if we had considered the time intervals for modeling. We did not consider the time interval since, time series was out of scope.



*Figure 3.6*

Below we have compared the distributions of the actuals versus the predicted to see the shape and the line of fit. The line of fit is not very linear and the scales on x axis and y axis look similar but the predictions are still below par and the distribution not representative.



*Figure 3.7*

## **Inference**

On performing the above combinations and iterations we notice that the first trial of the least squares with the given parameters is the most successful as it is better than the baseline, has the highest net revenue and passes the RMSE criterion. However, it still shows a residual trend.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NAME** | **Cross & Transformed Variables Included** | **Singularity** | **Only Significant Variables** | **Type of E3, E6, E8, E10 and V5** | **Over-fitting?** | **R Square (Train/Validation/Test)** | **Residuals Plot?** | **RMSE** |
| LS1 | No | Yes | No | Nominal/Ordinal | No | 0.7921 0.7930 0.7927 | Visible Trend | 12.56 |
| LS2 | Yes | No | Yes | Nominal/Ordinal | No | 0.7880 0.7889 0.7854 | Visible Trend | 12.68 |
| LS3 | No | No | No | Continuous | No | 0.7868 0.7880 0.7911 | Visible Trend | 12.56 |
| LS4 | Yes | No | Yes | Continuous | No | 0.7857 0.7869 0.7893 | Visible Trend | 12.74 |

*Table 3.1*

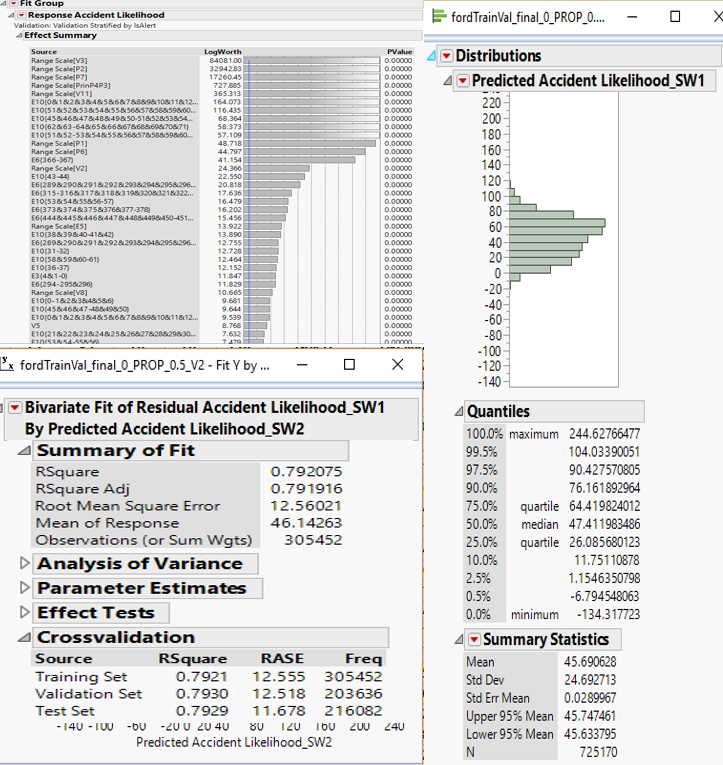
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Cost of the model** | **Count of Out of threshold Errors** | **Cost of Out of threshold Errors** | **Count of within threshold predictions** | **Benefit of within threshold predictions** | **Count of Max/Min Errors** | **Cost of Max/Min Errors** | **Net Revenue** |
|  | **(A)** |  | **(B)** |  | **(C')** |  | **(D)** | **A+B+C+D** |
| LS1 | -$4,000.00 | 154615 | -$463,845.00 | 61467 | $430,269.00 | 8612 | -$8,612.00 | -$46,188.00 |
| LS2 | -$4,000.00 | 156338 | -$469,014.00 | 59744 | $418,208.00 | 8689 | -$8,689.00 | -$63,495.00 |
| LS3 | -$4,000.00 | 155919 | -$467,757.00 | 60163 | $421,141.00 | 8757 | -$8,757.00 | -$59,373.00 |
| LS4 | -$4,000.00 | 157016 | -$471,048.00 | 59066 | $413,462.00 | 8526 | -$8,526.00 | -$70,112.00 |

*Table 3.2*

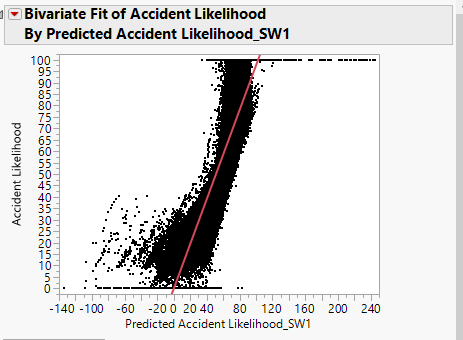
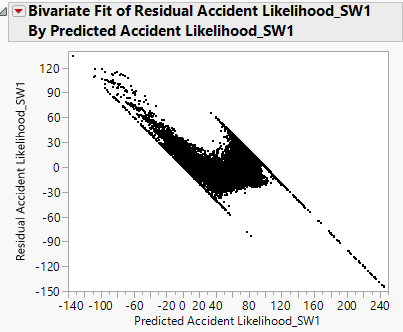
## **4.3 Stepwise**

We run the Stepwise model generally and using the tweaks suggested in the model tweaking section. Also, as discussed in the pre-processing step, we are considering variables E3, E6, E8, E10 and V5 as nominal/ordinal predictors and later compare model performance with the time when we use them as continuous predictors. Here we aren’t taking significance of variables into consideration, since this is a stepwise model and we will stop the selection of variables once the significance of it reduces.

## **Observations**



*Figure 3.8*

*Figure* *3.9*

## **Inference**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Cross & Transformed Variables Included** | **Stopping Rule** | **Singularity** | **Only Significant Variables** | **Type of E3, E6, E8, E10 and V5** | **Overfitting?** | **R Square (Train/Validation/Test)** | **Residuals Plot?** | **RMSE** |
| SW1 | No | Max Validation R Square | No | Yes | Nominal/Ordinal | No | 0.7921 0.7930 0.7929 | Visible Trend | 12.56 |
| SW2 | Yes | P Value Threshold | No | Yes | Continuous | No | 0.9476 0.9462 0.9432 | Visible Trend | 12.68 |

*Table 3.3*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Cost of the model** | **Count of Out of threshold Errors** | **Cost of Out of threshold Errors** | **Count of within threshold predictions** | **Benefit of within threshold predictions** | **Count of Max/Min Errors** | **Cost of Max/Min Errors** | **Net Revenue** |
|  | **(A)** |  | **(B)** |  | **(C')** |  | **(D)** | **A+B+C+D** |
| SW1 | -$4,200.00 | 154654 | -$463,962.00 | 61428 | $429,996.00 | 8601 | -$8,601.00 | -$46,767.00 |
| SW2 | -$4,200.00 | 71208 | -$213,624.00 | 144874 | $1,014,118.00 | 6807 | -$6,807.00 | $789,487.00 |

*Table 3.4*

## **4.4 Neural Network**

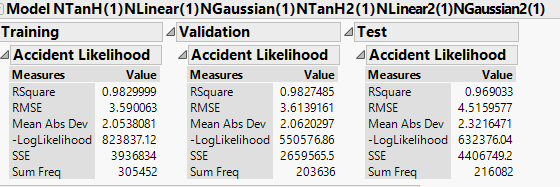
We run the Neural Networks model generally and using the tweaks suggested in the model tweaking section. Also, as discussed in the model tweaking section, we are not considering variables E3, E6, E8, E10 and V5 as nominal/ordinal and continuous predictors and redoing the

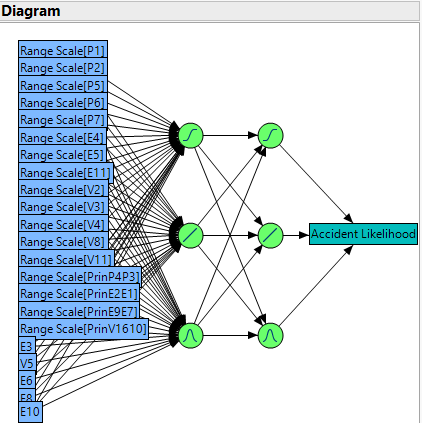
analysis as it hasn’t yielded any different results so far. We would be changing the number of layers and try to analyze the results.

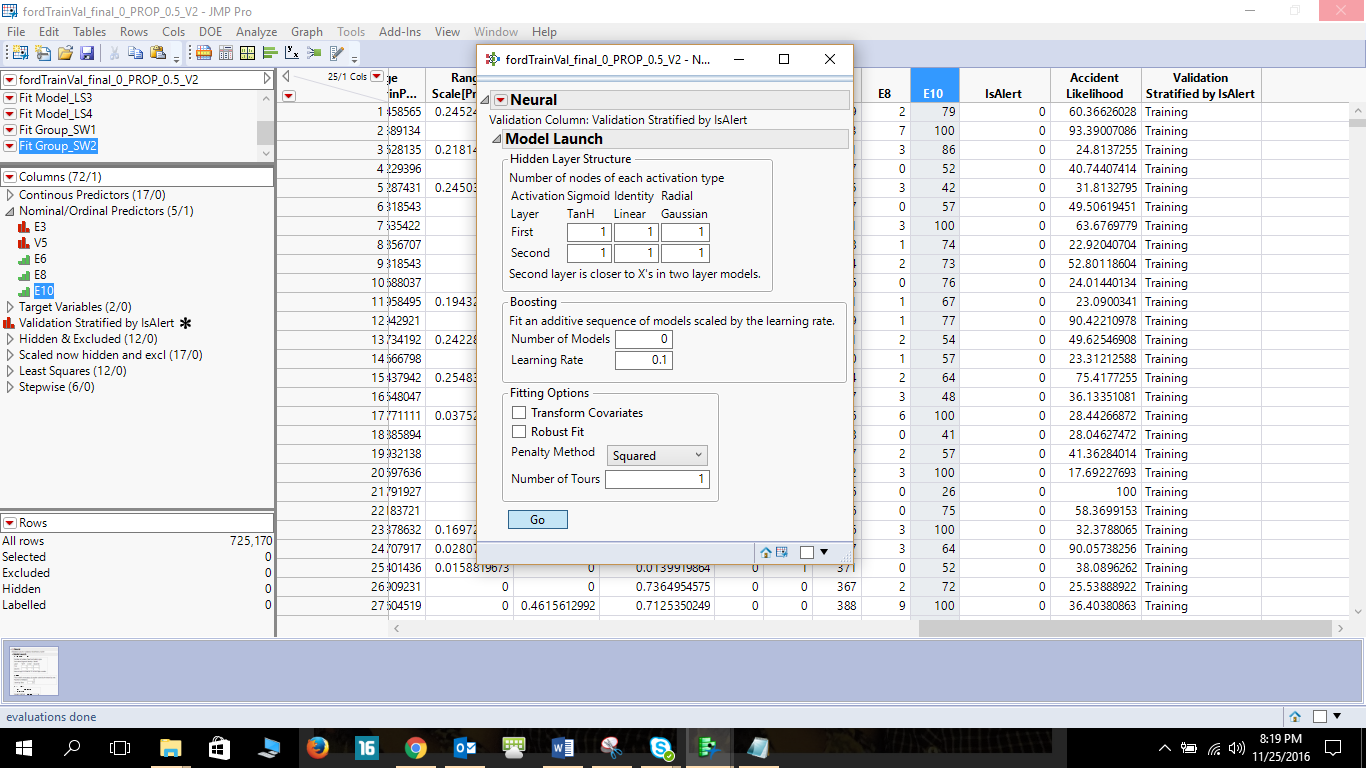
*Figure 3.10*

## **Observations**

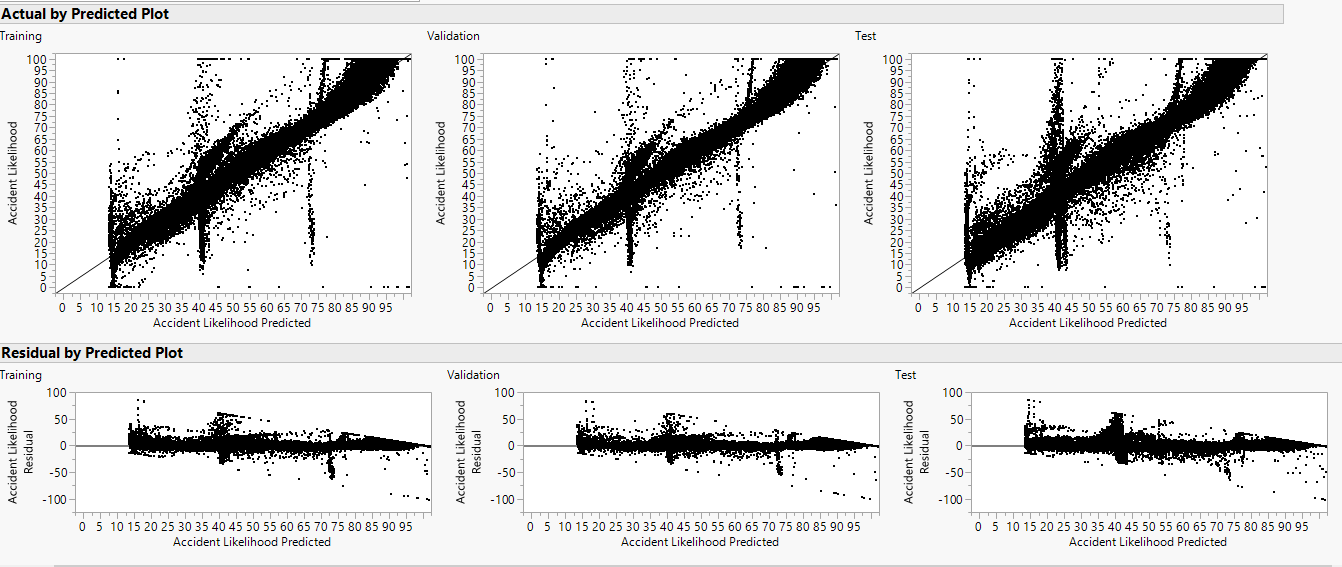
We notice that the R-square values are extremely good for this and does not point to overfitting.



*Figure 3.11*

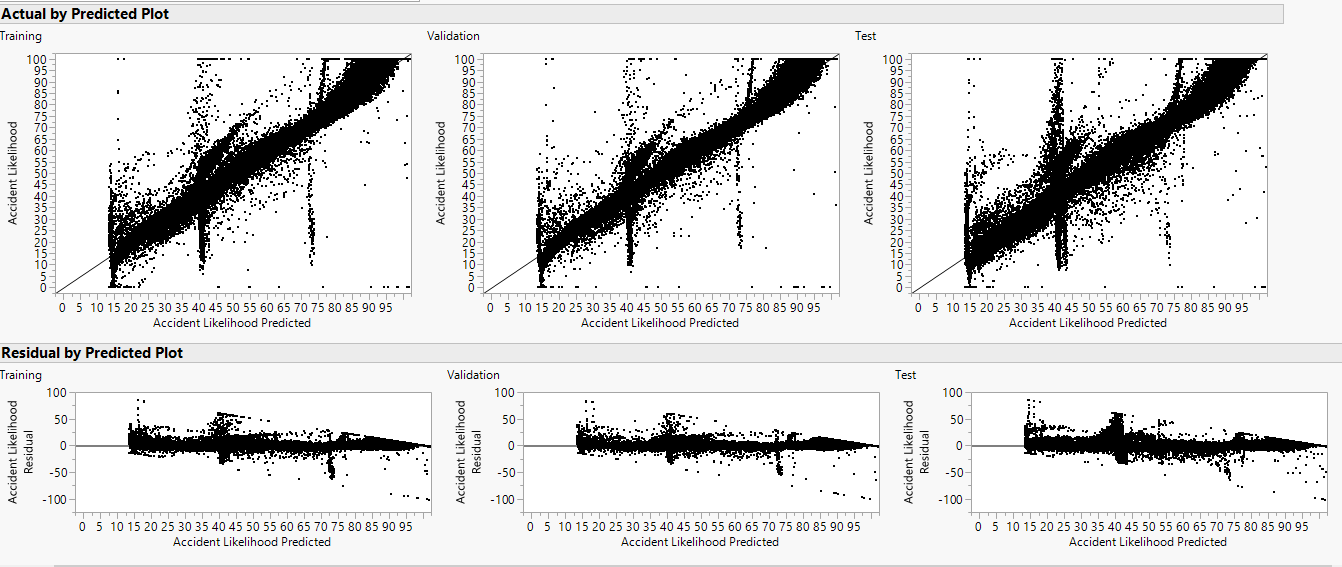


The bivariate looks more linear now leaning towards the 45o line.



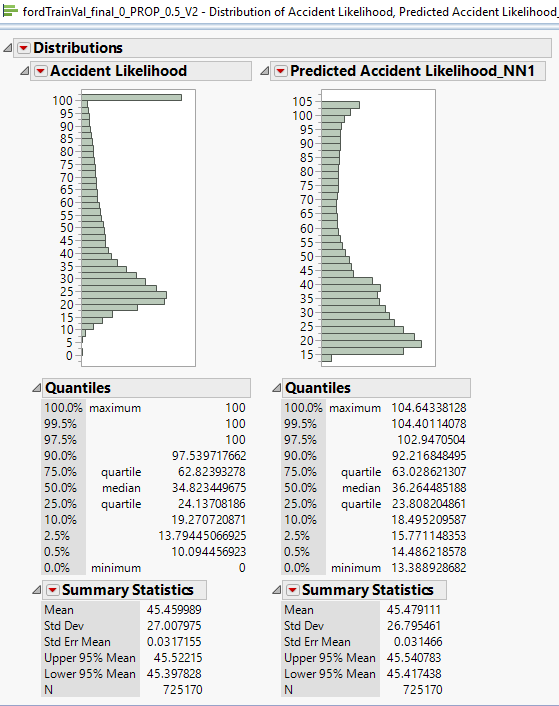
*Figure 3.12*

Here we finally notice no unusual trend in the residuals vs predicted plot and most of the points are near the mean line.



*Figure 3.13*

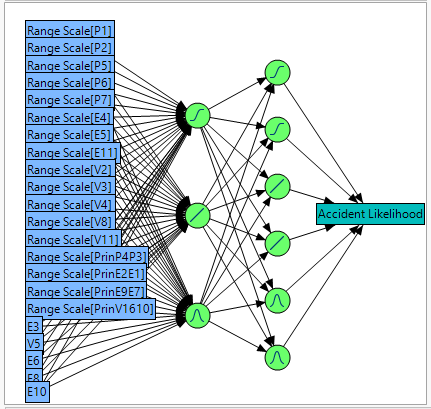
Observing the distribution, we notice tremendous improvement and increasing similarities.



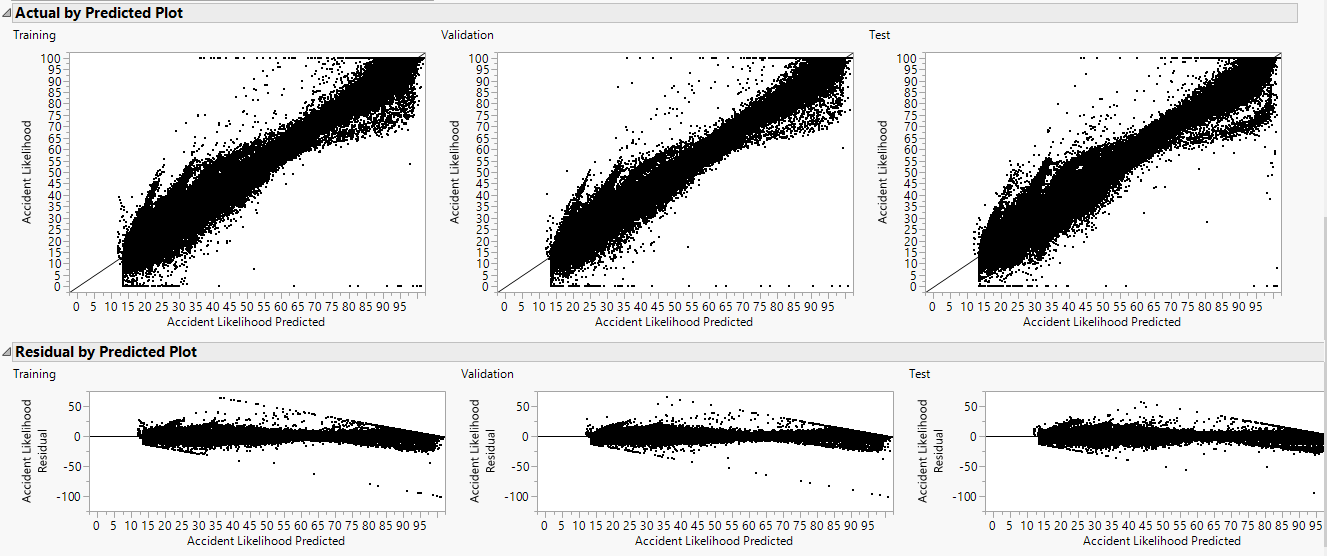
*Figure 3.14*

What we notice further is surprising, since although the residual plots and within threshold values look better, the Count of Max/Min errors exceed the rest, even than in the previous models. We note here that the RMSE value has drastically reduced and other improvements, to further improve this model can be done. We now must figure out a way to reduce the Count of Max/Min errors.

We run a new neural net NN2 with the parameter given below in the 2nd row, the results are documented. We notice that although the Count of Max/Min decreases the net revenue decreases as compared to the NN1.



*Figure 3.15*



*Figure 3.16*

## **Inferences**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NAME** | **Cross & Transformed Variables Included** | **Layers** | **Overfitting?** | **R Square (Train/Validation/Test)** | **Residuals Plot?** | **RMSE**  **(Test)** |
| NN1 | No | 111 111 | No | 0.9829 0.9827  0.9690 | PASS | 4.51 |
| NN2 | No | 111 222 | No | 0.9833 0.9834 0.9745 | PASS | 4.09 |

*Table 3.5*

*Table 3.6*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Cost of the model** | **Count of Out of threshold Errors** | **Cost of Out of threshold Errors** | **Count of within threshold predictions** | **Benefit of within threshold predictions** | **Count of Max/Min Errors** | **Cost of Max/Min Errors** | **Net Revenue** |
|  | **(A)** |  | **(B)** |  | **(C')** |  | **(D)** | **A+B+C+D** |
| NN1 | -$4,600.00 | 17201 | -$51,603.00 | 198881 | $1,392,167.00 | 9132 | -$9,132.00 | $1,326,832.00 |
| NN2 | -$4,600.00 | 30692 | -$92,076.00 | 185390 | $1,297,730.00 | 6741 | -$6,741.00 | $1,194,313.00 |

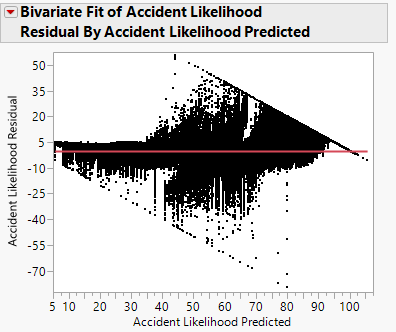
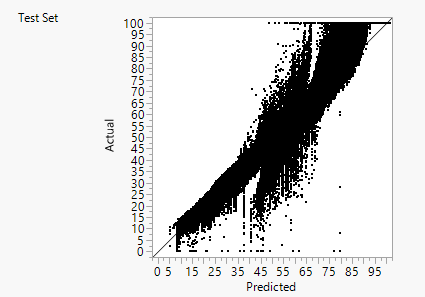
## **4.5 Boosted Tree**

We run the Boosted tree model generally and using the tweaks suggested in the model tweaking section. Also, as discussed in the pre-processing step, we are NOT considering variables E3, E6, E8, E10 and V5 as nominal/ordinal predictors and redoing the analysis as continuous because it hasn’t yielded any different results so far in all the models run. We would be changing the number trees and analyzing the results.

## **4.5.1 Observations**

We notice here in BT1 that the Count of Max/Min errors drastically reduces, although the count of out of threshold errors increase and there is a visible trend in the residual plot. The distribution looks good. The RMSE when compared to regression is low, but neural nets still have a better RMSE value. This looks like a potentially good model.

When BT2 is run, the RMSE reduces and the residual plot improves, although the Count of Max & Min Errors increases. This betters BT1 on performance over all criterions.



*Figure 3.17*

## **Inference**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **NAME** | **Cross & Transformed Variables Included** | **No. of Layers** | **Overfitting?** | **R Square (Train/Validation/Test)** | **Count of Max & Min Errors** | **In Range Predictions** | **Residuals Plot?** | **RMSE**  **(Test)** |
| BT1 | No | 50 | No | 0.954 0.954 0.944 | 75 | 216007 | Visible Trend | 6.06 |
| BT2 | No | 100 | No | 0.977 0.976 0.972 | 3859 | 212223 | PASS | 4.26 |

Table 3.7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Cost of the model** | **Count of Out of threshold Errors** | **Cost of Out of threshold Errors** | **Count of within threshold predictions** | **Benefit of within threshold predictions** | **Count of Max/Min Errors** | **Cost of Max/Min Errors** | **Net Revenue** |
|  | **(A)** |  | **(B)** |  | **(C')** |  | **(D)** | **A+B+C+D** |
| BT1 | -$5,000.00 | 55536 | -$166,608.00 | 160546 | $1,123,822.00 | 75 | -$75.00 | $952,139.00 |
| BT2 | -$5,000.00 | 30692 | -$92,076.00 | 185390 | $1,297,730.00 | 3859 | -$3,859.00 | $1,196,795.00 |

Table 3.8

## **4.6 Ensemble Model**

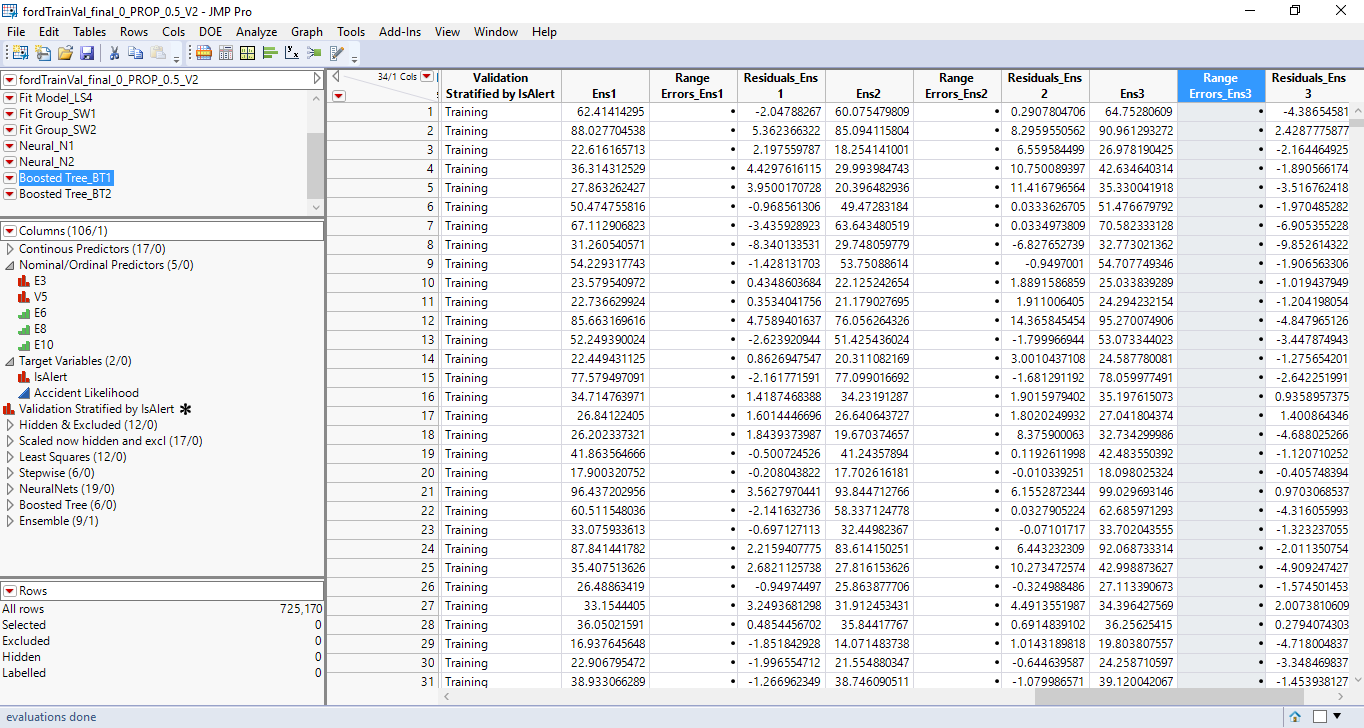
Based on our parameters we initially selected, these are the best settings of all the models that have passed all criterions. Now we ensemble a combination of these given below.

## **Observation**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NAME** | **Cross & Transformed Variables Included** | **Layers** | **Overfitting?** | **R Square (Train/Validation/Test)** | **Net Revenue** | **Residuals Plot?** | **RMSE**  **(Test)** |
| NN1 | No | 111 222 | No | 0.9833 0.9834 0.9745 | $1,326,832.00 | PASS | 4.51 |
| BT2 | No | 100 | No | 0.977 0.976 0.972 | $1,196,795.00 | PASS | 4.26 |

*Table 3.9*

From the above, only the results of the NN1 and the BT2 look encouraging. Thus, we go ahead and build and ensemble of these two and compare with the original NN1 and BT2.



*Figure 3.18*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NAME** | **Equation** | **Overfitting?** | **R Square** | **Residuals Plot?** | **RMSE**  **(Test)** |
| ENS1 | Average (NN2, BT2) | No | 0.9830 | Visible Trend | 3.51 |
| ENS2 | Max (NN2, BT2) | No | 0.9764 | Visible Trend | 4.14 |
| ENS3 | Min (NN2, BT2) | No | 0.9760 | Visible Trend | 4.17 |

## **Inference**

*Table 3.10*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Cost of the model** | **Count of Out of threshold Errors** | **Cost of Out of threshold Errors** | **Count of within threshold predictions** | **Benefit of within threshold predictions** | **Count of Max/Min Errors** | **Cost of Max/Min Errors** | **Net Revenue** |
|  | **(A)** |  | **(B)** |  | **(C')** |  | **(D)** | **A+B+C+D** |
| ENS1 | -$9,500.00 | 18110 | -$54,330.00 | 197972 | $1,385,804.00 | 4390 | -$4,390.00 | $1,317,584.00 |
| ENS2 | -$8,000.00 | 32802 | -$98,406.00 | 183280 | $1,282,960.00 | 2878 | -$2,878.00 | $1,173,676.00 |
| ENS3 | -$10,000.00 | 35858 | -$107,574.00 | 180224 | $1,261,568.00 | 7722 | -$7,722.00 | $1,136,272.00 |

*Table 3.11*

## **4.7 Final Evaluation**

We highlight the best features and go for the profit matrix. Assuming, that the cost of the model is given to us by the people who built them.

|  |  |
| --- | --- |
| Benefit of within threshold predictions | +$7 .00 |
| Cost of Out of threshold Errors | -$3.00 |
| Cost of Max/Min Errors | -$1.00 |

*Table 3.12*

|  |  |  |  |
| --- | --- | --- | --- |
| **MODEL** | **Net Revenue** | **RMSE** | **Residual Plot** |
| BASELINE | -$446,426.00 | 25.77 | Visible Trend |
| LS1 | -$46,188.00 | 12.56 | Visible Trend |
| SW2 | $789,487.00 | 12.68 | Visible Trend |
| NN1 | $1,326,832.00 | 4.51 | PASS |
| BT2 | $1,196,795.00 | 4.26 | PASS |
| ENS1 | $1,317,584.00 | 3.51 | Visible Trend |

*Table 3.13*

## **4.7.1 Recommendation – Continuous Response**

It is close between BT2 and ENS2, although the net revenue of ENS2 is better it fails our other criterions, the BT2 satisfies all our criterions. Although the problems remain that we were not able to find a model that fulfills all our criteria above. Especially a model that gives us maximum revenue while fulfilling all other criterions. For now, we will stick with BT2 since it is better than our baseline, although not in terms of net revenue, but it completes the objectives as stated in the aforementioned model selection criteria and beats our baseline with a better RMSE and Residual Plot.

## **Model Analysis of Categorical Response (Without Outliers)**

For modeling with categorical target variable, we begin with grouping all columns such that, running a model is easier and organized.

We have run a total of 8 categorical prediction models.

1. Decision Tree
2. Discriminant Analysis
3. Neural Networks
4. Nominal Logistic Regression
5. Ensemble – 1 to 4

## **Model Tweaking**

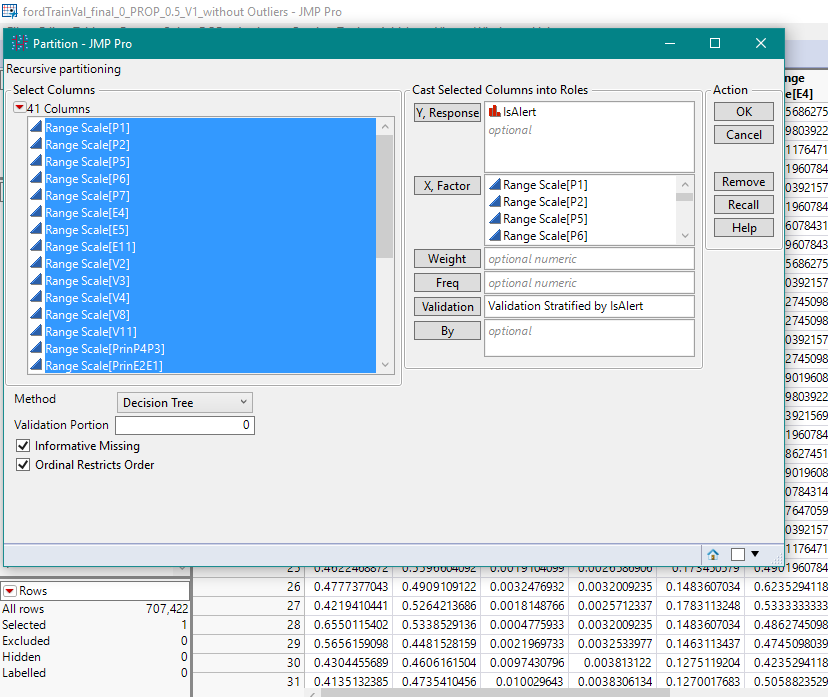
For every nominal model, we do the following tweaks to fine tune our model and run multiple iterations to get better results.

1. Remove the variable from the predictors that do not have any variation i.e. do not add any predictive value
2. Choose all the variables that are significant.

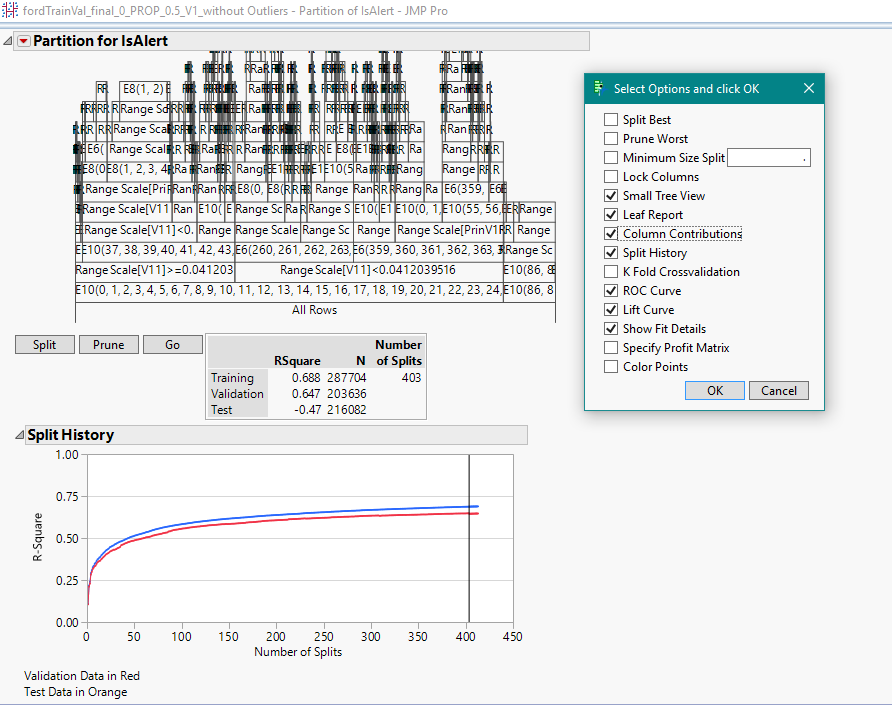
## **Decision Tree**

### **Observation**

For Decision Tree Model, we have taken the following variables in consideration:



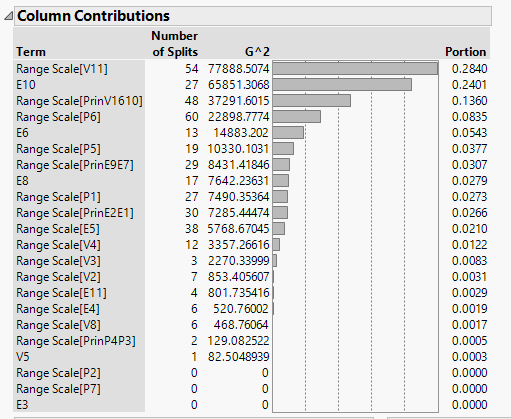
*Figure 4.1*



*Figure 4.2*

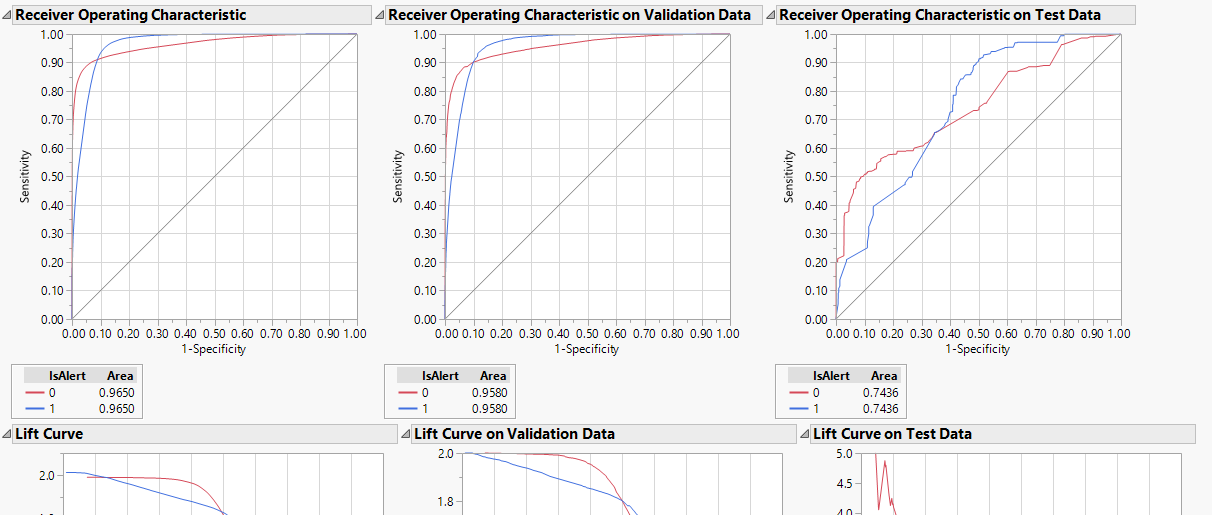
As we can see from the split history, the optimum split is found at 403 split.

From the below column contribution details, we can find out the information gain of each variable along with the number of splits. We see that, variable P2, P7 and E3 does not contribute much in the split. Thus, we can say that these 3 variables do not add much value to the prediction.



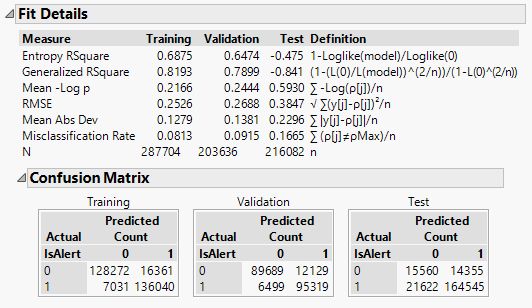
*Figure 4.3*

The ROC curve for Training and Validation displays almost the same results with a significantly good performance, but the Test dataset happens to result a much lower value than the Training and Validation. This also points towards a model overfit.



*Figure 4.4*

In the Fit Details, we can see the details of misclassification rate of all the 3 datasets. It is very evident that the Training and Validation have much lesser rate than that of the Test.



*Figure 4.5*

We have observed the confusion matrix and have come up with the following matrix for the TPR, SPC and ACC values:

Test Dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 95.09% | 93.62% | 88.39% |
| **True Negatives** | 88.69% | 88.09% | 52.01% |
| **False Positives** | 11.31% | 11.91% | 47.99% |
| **False Negatives** | 4.91% | 6.38% | 11.61% |
| **Model Accuracy** | 91.87% | 90.85% | 83.35% |

*Table 4.1*

### **Inference**

We can infer that the above model has very high misclassification rate on the Test data compared to Training and Validation. Also, the ROC curve doesn’t seem good compared to its Training and Validation dataset performance. If we consider the confusion matrix comparison, the true negative and the accuracy rate is very low for the Test dataset compared to the Training and Validation. We now calculate the cost of the model at various probability cut-offs for the test data:

**Cost Analysis on Test Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cut-off | Alert as Not Alert (1 as 0) | Not Alert as Alert (0 as 1) | Alert as Alert (1 as 1) | Not Alert as Not Alert (0 as 0) | Total Profit of the Model |
| 0.1 | 14024 | 15985 | 172143 | 13930 | **$ 1,478,240.00** |
| 0.2 | 15481 | 15517 | 170686 | 14398 | **$ 1,496,100.00** |
| 0.3 | 17536 | 15325 | 168631 | 14590 | **$ 1,474,200.00** |
| 0.4 | 19745 | 14965 | 166422 | 14950 | **$ 1,466,020.00** |
| 0.5 | 21622 | 14355 | 164545 | 15560 | **$ 1,489,480.00** |
| 0.6 | 26784 | 13465 | 159383 | 16450 | **$ 1,475,240.00** |
| 0.7 | 32106 | 12701 | 154061 | 17214 | **$ 1,445,300.00** |
| 0.8 | 42064 | 12188 | 144103 | 17727 | **$ 1,297,340.00** |
| 0.9 | 90809 | 7922 | 95358 | 21993 | **$ 749,040.00** |

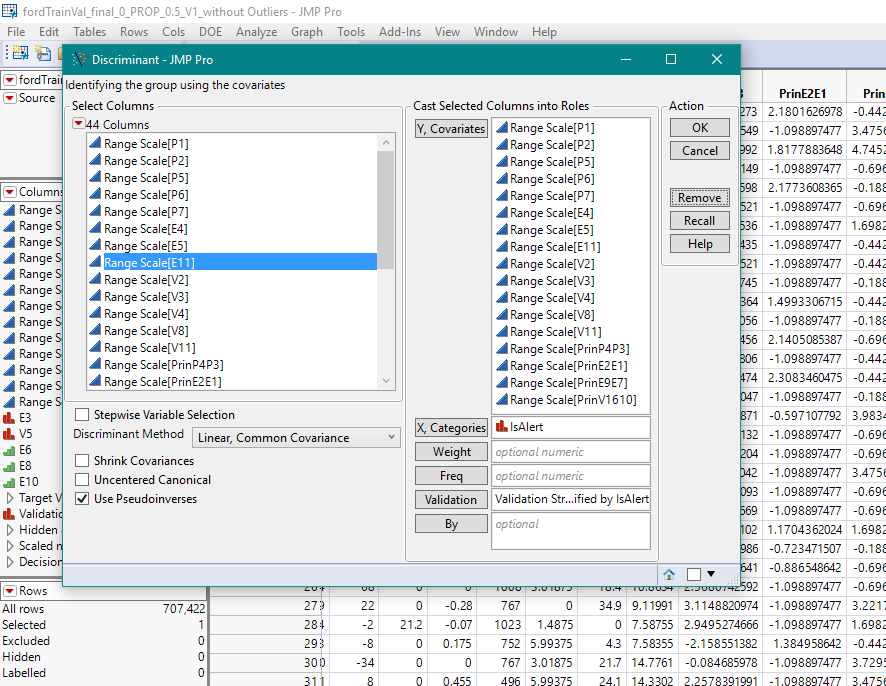
*Table 4.2*

The model is at its optimum at a probability cut-off of 0.2 with the total profit being $1,496,100.

## **Discriminant Analysis**

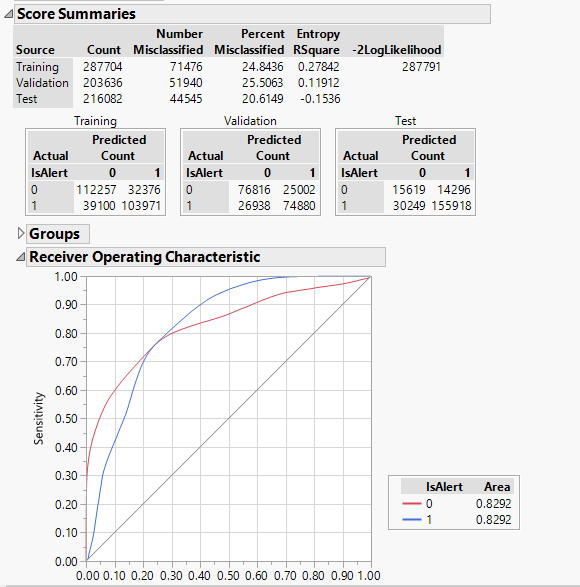
### **Observation**

For Discriminant Analysis, we have taken the following variables in consideration:



*Figure 4.6*

We have observed the confusion matrix and have come up with the following matrix for the TPR, SPC and ACC values:



*Figure 4.7*

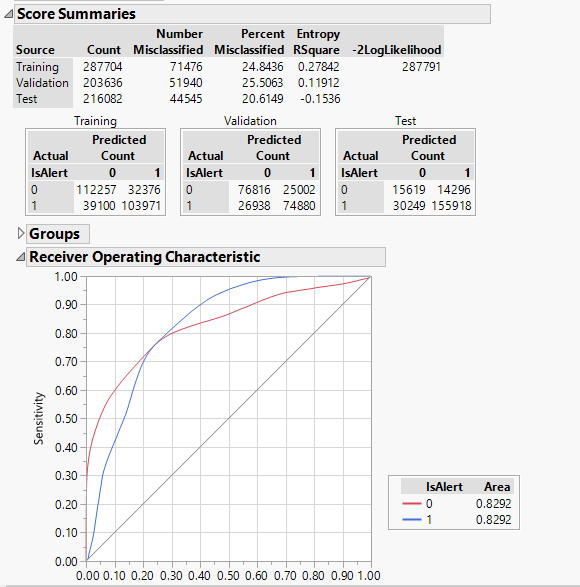
Test Dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 72.67% | 73.54% | 83.75% |
| **True Negatives** | 77.62% | 75.44% | 52.21% |
| **False Positives** | 22.38% | 24.56% | 47.79% |
| **False Negatives** | 27.33% | 26.46% | 16.25% |
| **Model Accuracy** | 75.16% | 74.49% | 79.39% |

*Table 4.3*

Although the values for TPR, SPC and ACC for the Test dataset in this model performed better than the Training and Validation dataset, but they are not better than our previous Decision Tree model.

The ROC curve seems to be better than our previous Decision Tree model.



*Figure 4.8*

### **Inference**

Though it is performing better than Decision Tree, we go ahead with calculating the cost of the model at various cut-off level.

**Cost Analysis on Test Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cut-off | Alert as Not Alert (1 as 0) | Not Alert as Alert (0 as 1) | Alert as Alert (1 as 1) | Not Alert as Not Alert (0 as 0) | Total Profit of the Model |
| 0.1 | 78 | 27939 | 186089 | 1976 | **$ 561,960.00** |
| 0.2 | 631 | 23243 | 185536 | 6672 | **$ 1,020,500.00** |
| 0.3 | 7125 | 18348 | 179042 | 11567 | **$ 1,380,120.00** |
| 0.4 | 17783 | 15415 | 168384 | 14500 | **$ 1,460,260.00** |
| 0.5 | 30249 | 14296 | 155918 | 15619 | **$ 1,322,840.00** |
| 0.6 | 46817 | 12308 | 139350 | 17607 | **$ 1,190,280.00** |
| 0.7 | 82702 | 8315 | 103465 | 21600 | **$ 871,880.00** |
| 0.8 | 122274 | 2993 | 63893 | 26922 | **$ 612,640.00** |
| 0.9 | 153039 | 2086 | 33128 | 27829 | **$ 88,040.00** |

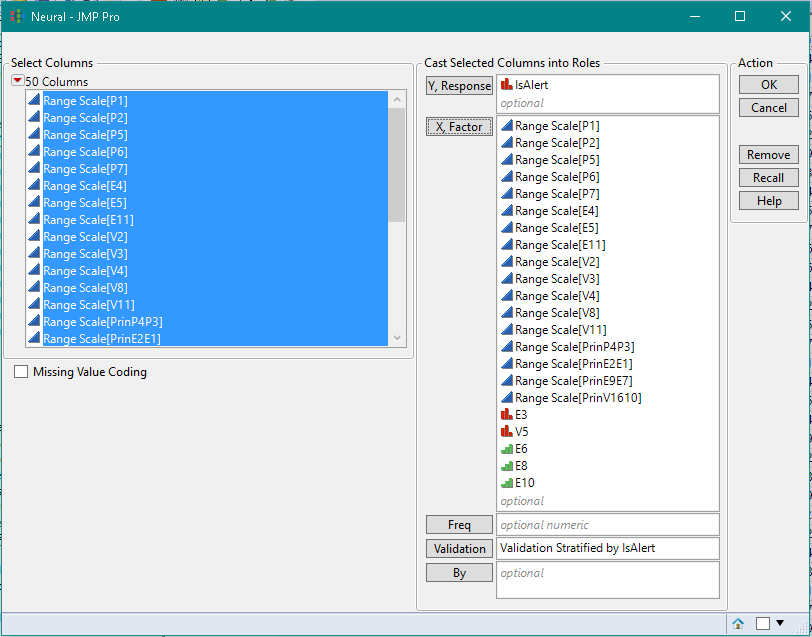
*Table 4.4*

The model is at its optimum at a probability cut-off of 0.4 with the total profit being $1,460,260

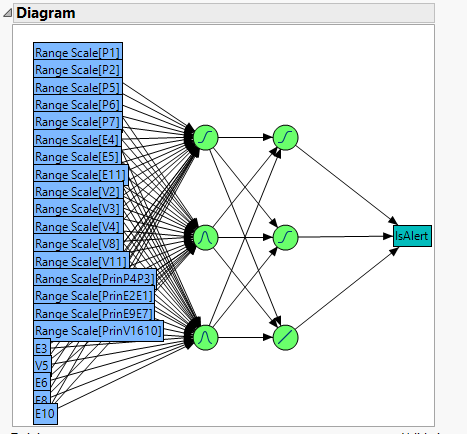
## **Neural Network**

### **Observation**

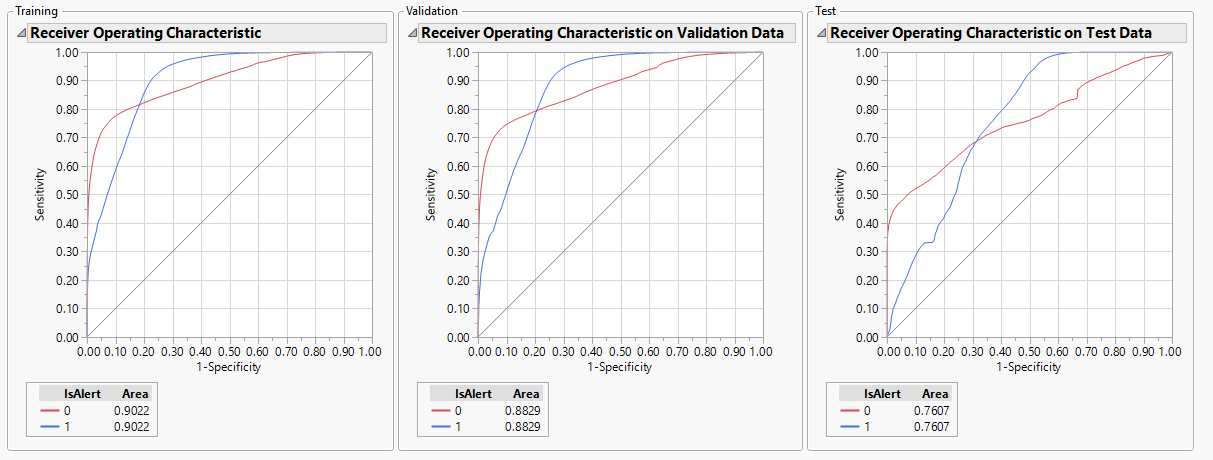
For Neural Network Model, we have taken the following variables in consideration:



*Figure 4.9*

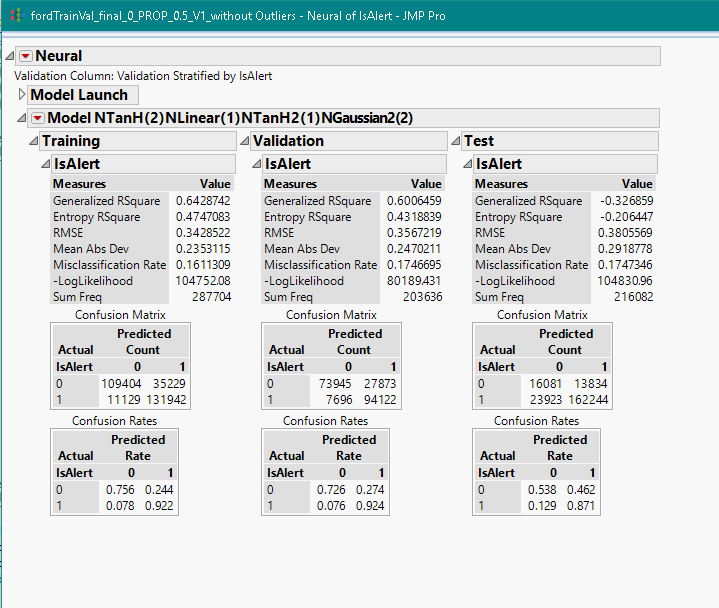


*Figure 4.10*



*Figure 4.11*

The ROC curve for Training and Validation displays a good result whereas the Test dataset happens to result a much lower value than the Training and Validation.



*Figure 4.12*

The misclassification rate for Training, Validation and Test dataset is more or less same.

We have observed the confusion matrix and have come up with the following matrix for the TPR, SPC and ACC values:

Test Dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 92.22% | 92.44% | 87.15% |
| **True Negatives** | 75.64% | 72.62% | 53.76% |
| **False Positives** | 24.36% | 27.38% | 46.24% |
| **False Negatives** | 7.78% | 7.56% | 12.85% |
| **Model Accuracy** | 83.89% | 82.53% | 82.53% |

*Table 4.5*

We note here that the misclassification rate of this model for Training, Validation and Test datasets are almost same and does not deviate much.

### **Inference**

The model accuracy of this model is same as that of Decision Tree Model. The accuracy for Training and Validation dataset is also not much different than Decision Tree Model. Hence, this can be considered as a valid model. Now, let us calculate the cost of the model for various cut-off level.

**Cost Analysis on Test Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cut-off | Alert as Not Alert (1 as 0) | Not Alert as Alert (0 as 1) | Alert as Alert (1 as 1) | Not Alert as Not Alert (0 as 0) | Total Profit of the Model |
| 0.1 | 4918 | 16449 | 181246 | 13466 | **$ 1,614,130.00** |
| 0.2 | 8408 | 15781 | 177759 | 14134 | **$ 1,611,160.00** |
| 0.3 | 12441 | 15147 | 173726 | 14768 | **$ 1,593,900.00** |
| 0.4 | 17125 | 14511 | 169042 | 15404 | **$ 1,563,820.00** |
| 0.5 | 23923 | 13834 | 162244 | 16081 | **$ 1,495,560.00** |
| 0.6 | 34060 | 12615 | 152107 | 17300 | **$ 1,414,720.00** |
| 0.7 | 54729 | 9725 | 131438 | 20190 | **$ 1,290,340.00** |
| 0.8 | 133933 | 2922 | 52234 | 26993 | **$ 386,560.00** |
| 0.9 | 148027 | 1971 | 38140 | 27944 | **$ 199,780.00** |

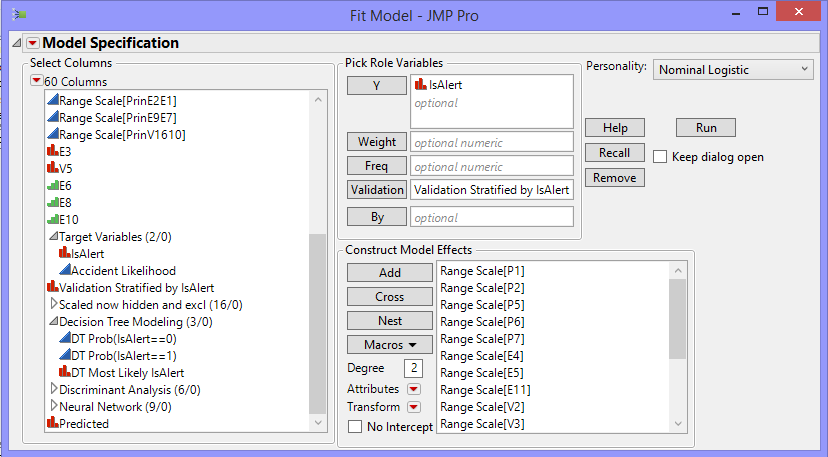
*Table 4.6*

The model is at its optimum at a probability cut-off at 0.1 with the total profit being $1,614,130.

## **Nominal Logistic Regression**

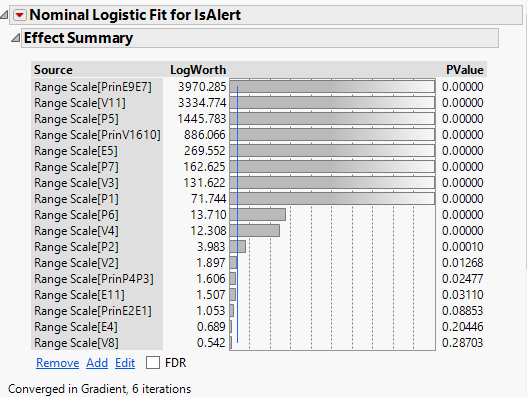
### **Observation**

For this model, we have taken the following variables in consideration:



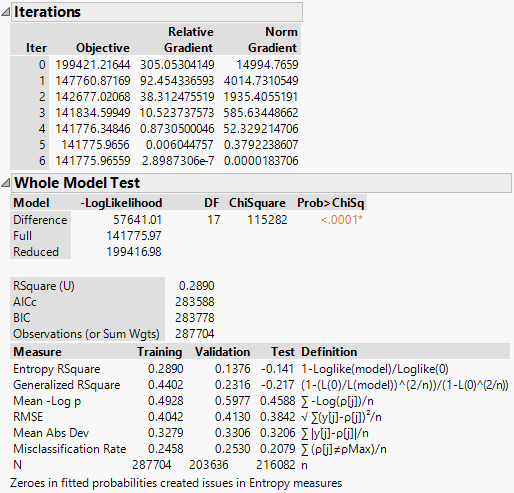
*Figure 4.13*

The information summary for the variables can be seen from the below effect summary



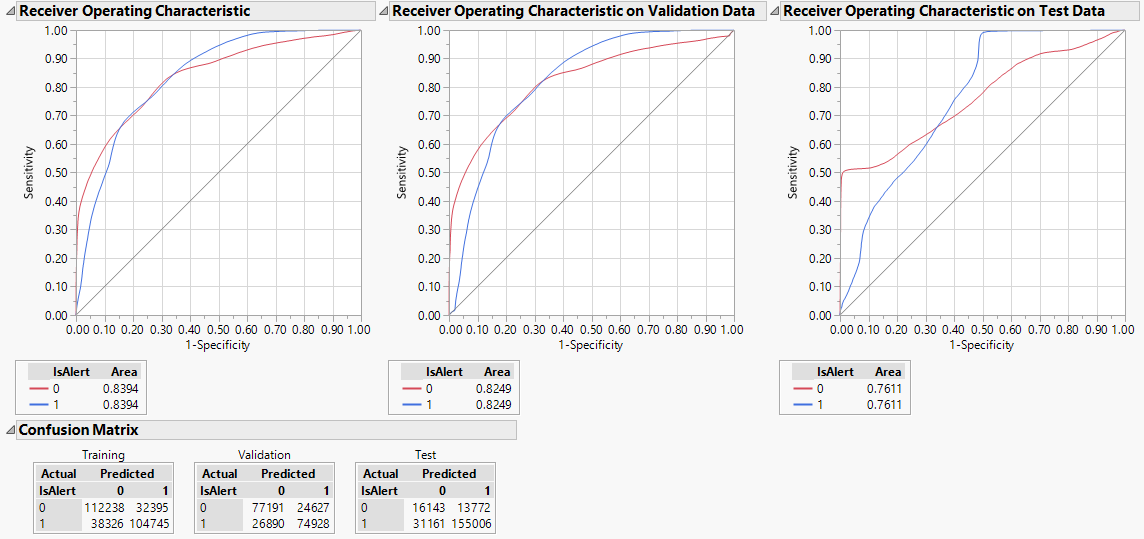
*Figure 4.14*

The misclassification rate of all the three Training, Validation and Test dataset are almost same and they seem to be a little higher than the previous models.

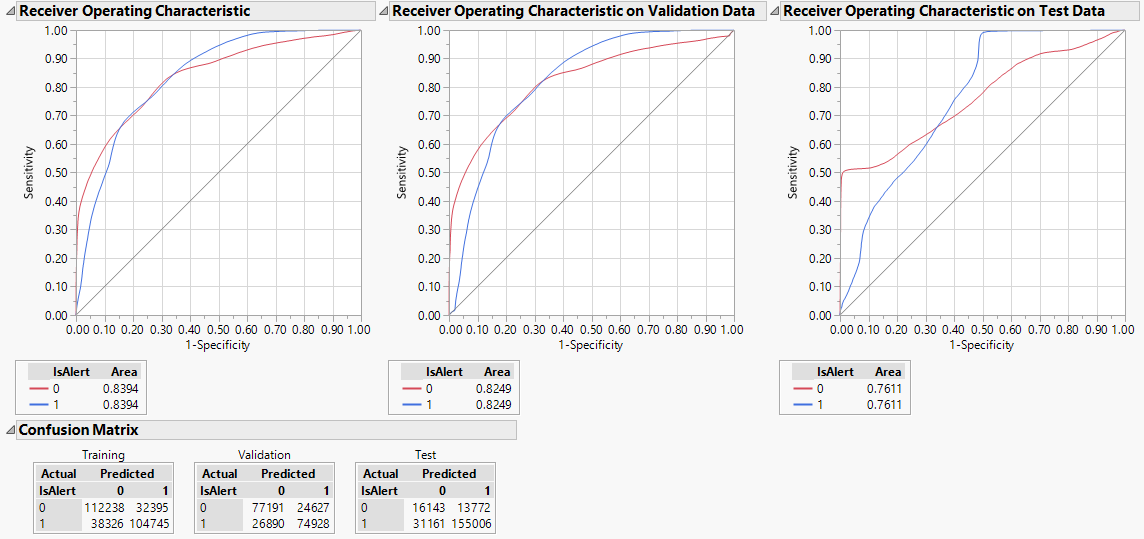


*Figure 4.15*

The ROC curve for Training and Validation displays almost same values, but the Test dataset happens to result a lower value than Training and Validation.



*Figure 4.16*



*Figure 4.17*

We have observed the confusion matrix and have come up with the following matrix for the TPR, SPC and ACC values:

Test Dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 73.21% | 73.59% | 83.26% |
| **True Negatives** | 77.60% | 75.81% | 53.96% |
| **False Positives** | 22.40% | 24.19% | 46.04% |
| **False Negatives** | 26.79% | 26.41% | 16.74% |
| **Model Accuracy** | 75.42% | 74.70% | 79.21% |

*Table 4.7*

We noted here that the TPR value in the Test dataset is higher than the Training and Validation dataset. In case of SPC also, the Test dataset has performed well than Training and Validation. The model accuracy for Test dataset is also high compared to the other two.

### **Inference**

As we have seen that the model accuracy, true negative and true positive rate all are on the higher side in the Test dataset compared to Training and Validation, we can say that the Test dataset is performing well. Along with this, the cost analysis of the model will give us a clearer picture on the model’s efficiency.

**Cost Analysis on Test Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cut-off | Alert as Not Alert (1 as 0) | Not Alert as Alert (0 as 1) | Alert as Alert (1 as 1) | Not Alert as Not Alert (0 as 0) | Total Profit of the Model |
| 0.1 | 322 | 21440 | 185845 | 8475 | **$ 1,206,980.00** |
| 0.2 | 768 | 15825 | 185399 | 14090 | **$ 1,759,560.00** |
| 0.3 | 5892 | 14650 | 180275 | 15265 | **$ 1,774,580.00** |
| 0.4 | 16984 | 14501 | 169183 | 15414 | **$ 1,567,640.00** |
| 0.5 | 31161 | 13772 | 155006 | 16143 | **$ 1,357,000.00** |
| 0.6 | 53089 | 11290 | 133078 | 18625 | **$ 1,166,640.00** |
| 0.7 | 88136 | 7324 | 98031 | 22591 | **$ 862,300.00** |
| 0.8 | 127922 | 2644 | 58245 | 27271 | **$ 534,580.00** |
| 0.9 | 156868 | 1741 | 29299 | 28174 | **$ 45,960.00** |

*Table 4.8*

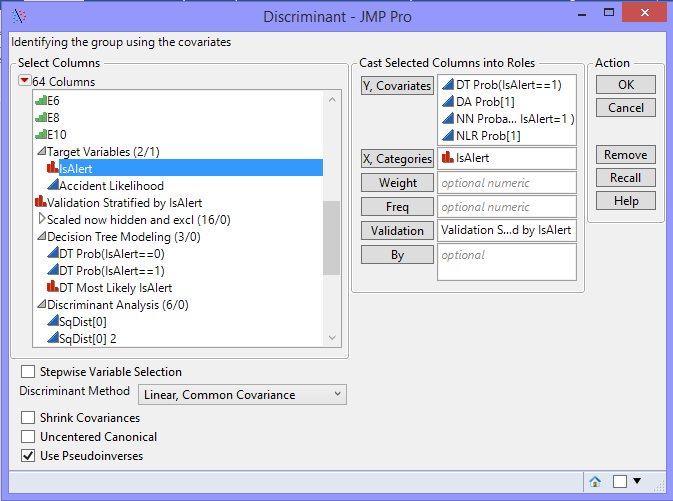
The model is at its optimum at a probability cut-off of 0.3 with the total profit being $1,774,580.

## **Ensemble Model 1**

So far, we have tried 4 different prediction models. Now, to get a better result we will try ensemble of all these 4 models following different methods. We begin with ensemble the predictor variables of all previous 4 models to form a Discriminant Analysis model.

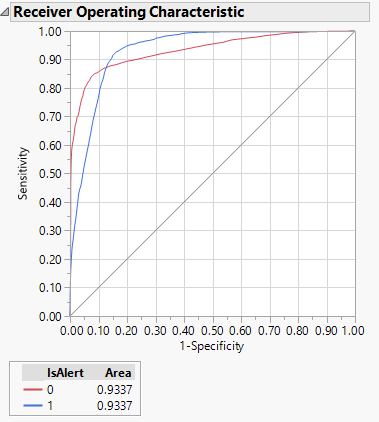
### **Observation**

We have taken the variables for this model as below:



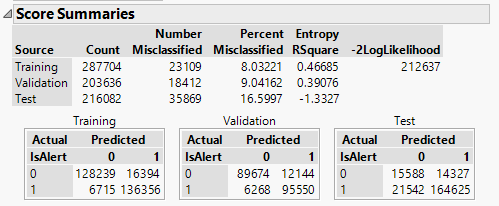
*Figure 4.18*

The ROC curve value shows a significantly higher value of 0.93.



*Figure 4.19*

The below score summaries display the confusion matrix of this model.



*Figure 4.20*

From the above confusion matrix, we can calculate the following:

Test Dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 95.31% | 93.84% | 88.43% |
| **True Negatives** | 88.67% | 88.07% | 52.11% |
| **False Positives** | 11.33% | 11.93% | 47.89% |
| **False Negatives** | 4.69% | 6.16% | 11.57% |
| **Model Accuracy** | 91.97% | 90.96% | 83.40% |

*Table 4.9*

### **Inference**

The model accuracy of the Training and Validation set is very high than the Test which shows a potential indicator of overfitting. In case of TPR and SPC also, the Training and Validation is performing well but the Test is not. Let us do the cost analysis of this model now.

**Cost Analysis on Test Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cut-off | Alert as Not Alert (1 as 0) | Not Alert as Alert (0 as 1) | Alert as Alert (1 as 1) | Not Alert as Not Alert (0 as 0) | Total Profit of the Model |
| 0.1 | 18971 | 15023 | 167196 | 14892 | **$ 1,475,700.00** |
| 0.2 | 19778 | 14894 | 166389 | 15021 | **$ 1,472,460.00** |
| 0.3 | 20319 | 14490 | 165848 | 15425 | **$ 1,502,040.00** |
| 0.4 | 20849 | 14383 | 165318 | 15532 | **$ 1,502,140.00** |
| 0.5 | 21542 | 14327 | 164625 | 15588 | **$ 1,493,880.00** |
| 0.6 | 22210 | 14220 | 163957 | 15695 | **$ 1,491,220.00** |
| 0.7 | 24278 | 13801 | 161889 | 16114 | **$ 1,491,760.00** |
| 0.8 | 26662 | 13425 | 159505 | 16490 | **$ 1,481,680.00** |
| 0.9 | 28596 | 13023 | 157571 | 16892 | **$ 1,483,200.00** |

*Table 4.10*

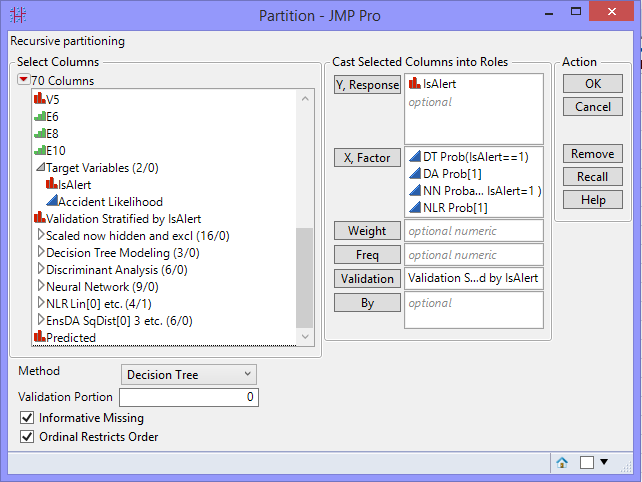
The model is at its optimum at a probability cut-off of 0.4 with the total profit being $1,502,140.

**5.7 Ensemble Model 2**

Now, we are going to use the probability values of our previous 4 models into a Decision Tree.

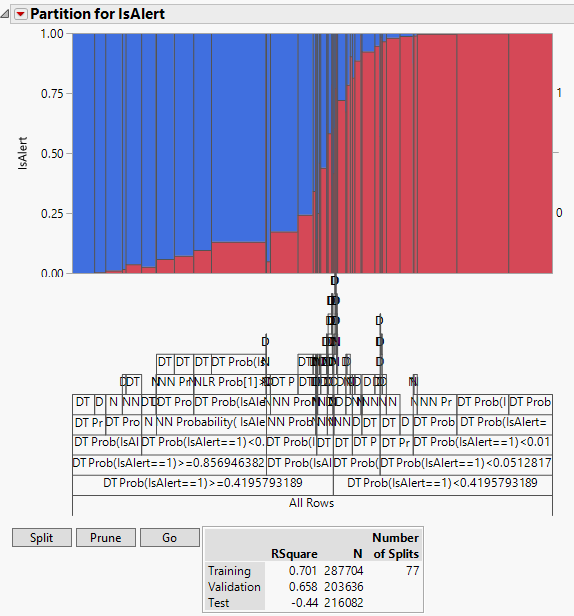
### **Observation**

We have taken the variables for this model as below:



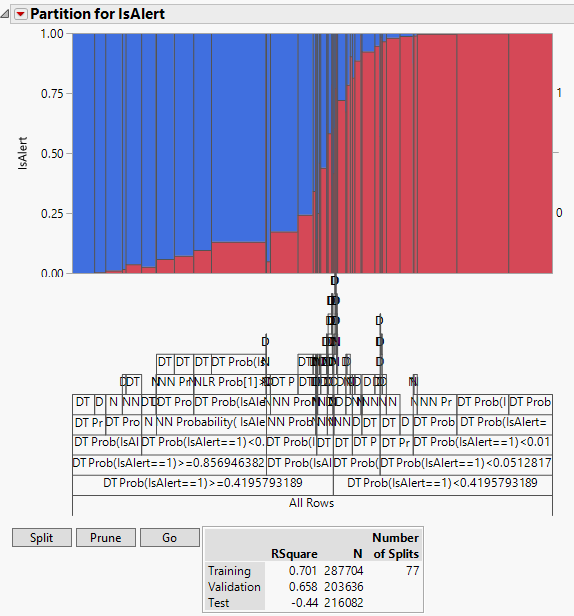
*Figure 4.21*

The Partition details is displayed as below.

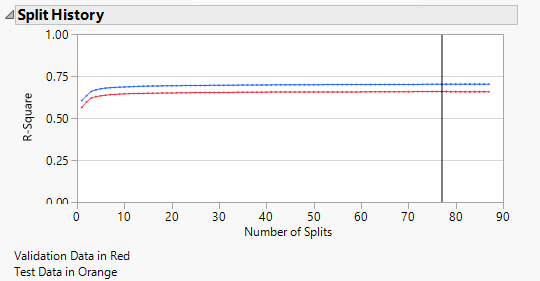


*Figure 4.22*

As it can be seen from Split History, the optimum split is found at the 77th split.

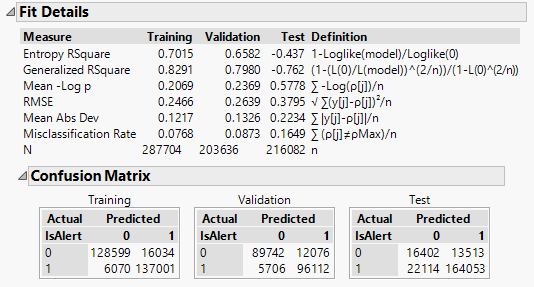


*Figure 4.23*



*Figure 4.24*

The misclassification rate for Training and Validation is less but is considerably high for the Test dataset.



*Figure 4.25*

From the above confusion matrix, we can calculate the following:

Test Dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 95.76% | 94.40% | 88.12% |
| **True Negatives** | 88.91% | 88.14% | 54.83% |
| **False Positives** | 11.09% | 11.86% | 45.17% |
| **False Negatives** | 4.24% | 5.60% | 11.88% |
| **Model Accuracy** | 92.32% | 91.27% | 83.51% |

*Table 4.11*

### **Inference**

Like the previous model, this model also behaves the same way. The model accuracy of the Training and Validation set is very high than the Test which shows a potential indicator of overfitting. In case of TPR and SPC also, the Training and Validation is performing well but the Test is not. Let us do the cost analysis of this model now.

**Cost Analysis on Test Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cut-off | Alert as Not Alert (1 as 0) | Not Alert as Alert (0 as 1) | Alert as Alert (1 as 1) | Not Alert as Not Alert (0 as 0) | Total Profit of the Model |
| 0.1 | 12812 | 15314 | 173355 | 14601 | **$ 1,569,780.00** |
| 0.2 | 15073 | 15012 | 171094 | 14903 | **$ 1,554,760.00** |
| 0.3 | 17820 | 14172 | 168347 | 15743 | **$ 1,583,820.00** |
| 0.4 | 19249 | 14142 | 166918 | 15773 | **$ 1,558,240.00** |
| 0.5 | 22114 | 13513 | 164053 | 16402 | **$ 1,563,840.00** |
| 0.6 | 26950 | 13318 | 159217 | 16597 | **$ 1,486,620.00** |
| 0.7 | 31575 | 12595 | 154592 | 17320 | **$ 1,466,420.00** |
| 0.8 | 40351 | 11944 | 145816 | 17971 | **$ 1,356,000.00** |
| 0.9 | 83479 | 8172 | 102688 | 21743 | **$ 870,640.00** |

*Table 4.12*

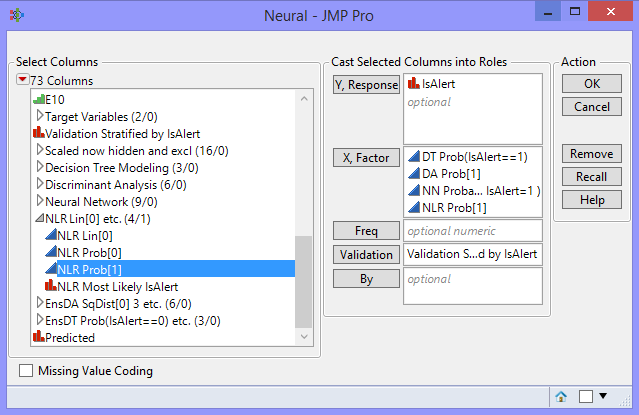
The model is at its optimum at a probability cut-off of 0.3 with the total profit being $1,583,820

## **Ensemble Model 3**

In this model, we are going to use the probability values of our previous 4 models into a Neural Network.

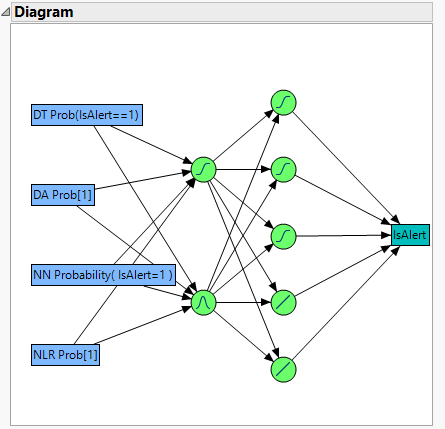
### **Observation**

We have taken the variables for this model as below:



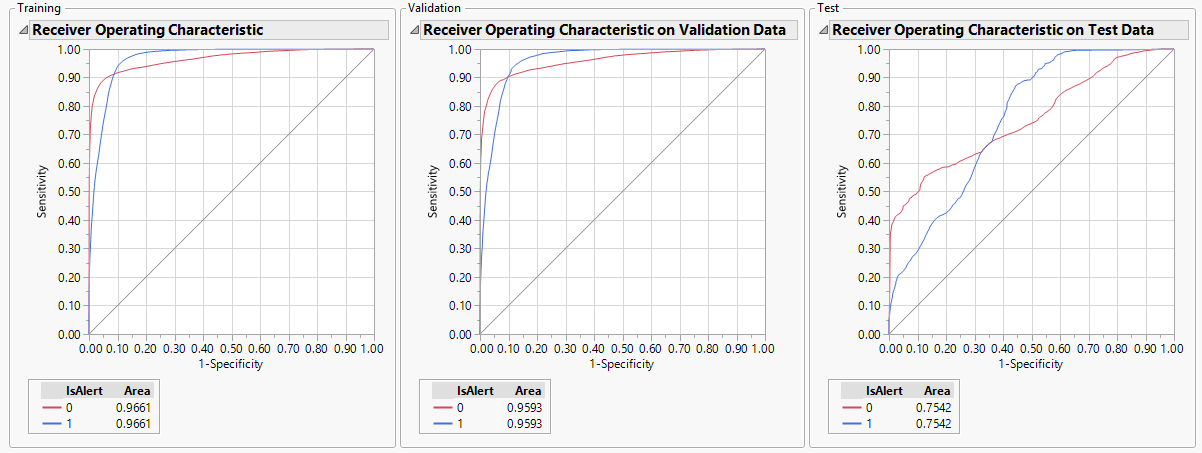
*Figure 4.26*

After running the model, we found the following diagram of the model:



*Figure 4.27*

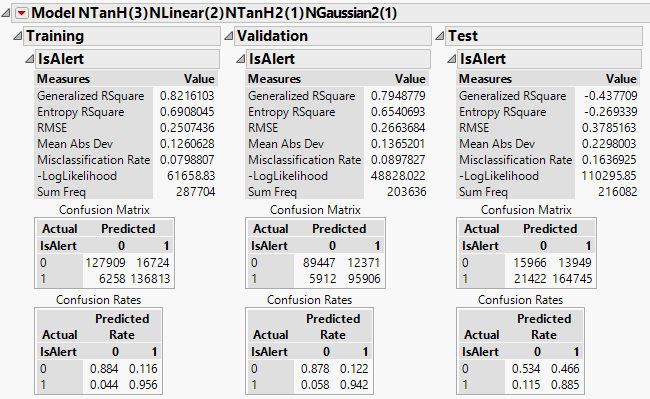
When we observed the ROC curve of all the three datasets, we found that Test dataset performed very badly compared to Training and Validation.



*Figure 4.28*

We can see from the below that although the misclassification rate for Training and

Validation is on a lower side but, the rate for Test dataset is on the higher side.



*Figure 4.29*

From the above confusion matrix, we can calculate the following:

Test Dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 95.63% | 94.19% | 88.49% |
| **True Negatives** | 88.44% | 87.85% | 53.37% |
| **False Positives** | 11.56% | 12.15% | 46.63% |
| **False Negatives** | 4.37% | 5.81% | 11.51% |
| **Model Accuracy** | 92.01% | 91.02% | 83.63% |

*Table 4.13*

### **Inference**

This model also behaves the pretty much same way like the previous two models. The model accuracy of the Training and Validation set is very high than the Test which shows a potential indicator of overfitting. In case of TPR and SPC also, the Training and Validation is performing well but the Test is not. Let us do the cost analysis of this model now.

**Cost Analysis on Test Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cut-off | Alert as Not Alert (1 as 0) | Not Alert as Alert (0 as 1) | Alert as Alert (1 as 1) | Not Alert as Not Alert (0 as 0) | Total Profit of the Model |
| 0.1 | 13064 | 15762 | 173103 | 14153 | **$ 1,519,940.00** |
| 0.2 | 15667 | 15381 | 170500 | 14534 | **$ 1,505,980.00** |
| 0.3 | 17881 | 15154 | 168286 | 14761 | **$ 1,484,400.00** |
| 0.4 | 19726 | 14860 | 166441 | 15055 | **$ 1,476,900.00** |
| 0.5 | 21422 | 13949 | 164745 | 15966 | **$ 1,534,080.00** |
| 0.6 | 25614 | 13188 | 160553 | 16727 | **$ 1,526,340.00** |
| 0.7 | 32023 | 12622 | 154144 | 17293 | **$ 1,454,760.00** |
| 0.8 | 44385 | 11985 | 141782 | 17930 | **$ 736,200.00** |
| 0.9 | 97887 | 7216 | 88280 | 22699 | **$ 678,080.00** |

*Table 4.14*

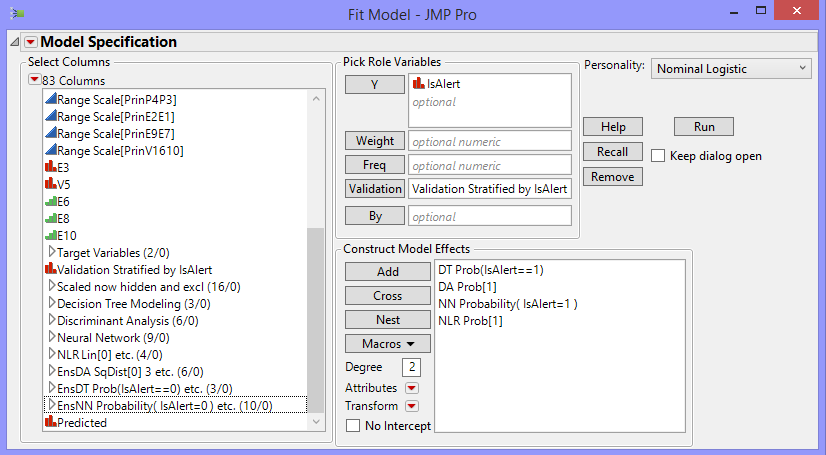
The model is at its optimum at a probability cut-off value of 0.5 with the total profit being $1,534,080

## **Ensemble Model 4**

Here, we are going to use the probability values of our previous 4 models into a Nominal Logistic Regression Model.

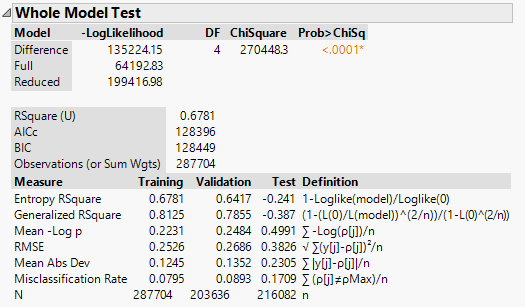
### **Observation**

We have taken the variables for this model as below:



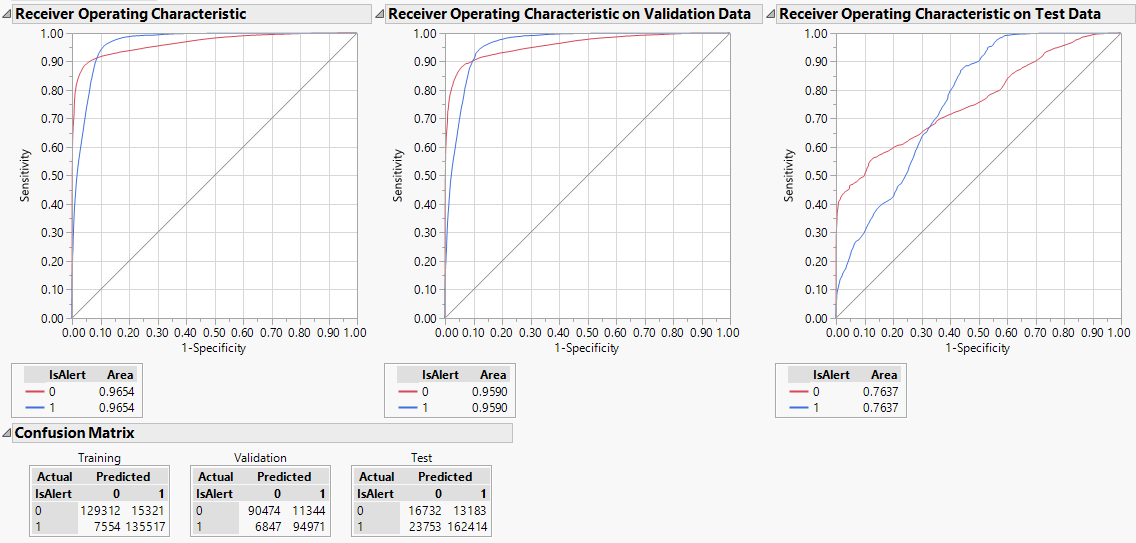
*Figure 4.30*

The misclassification rate for Test data shows a much higher value than the Training and Validation dataset.

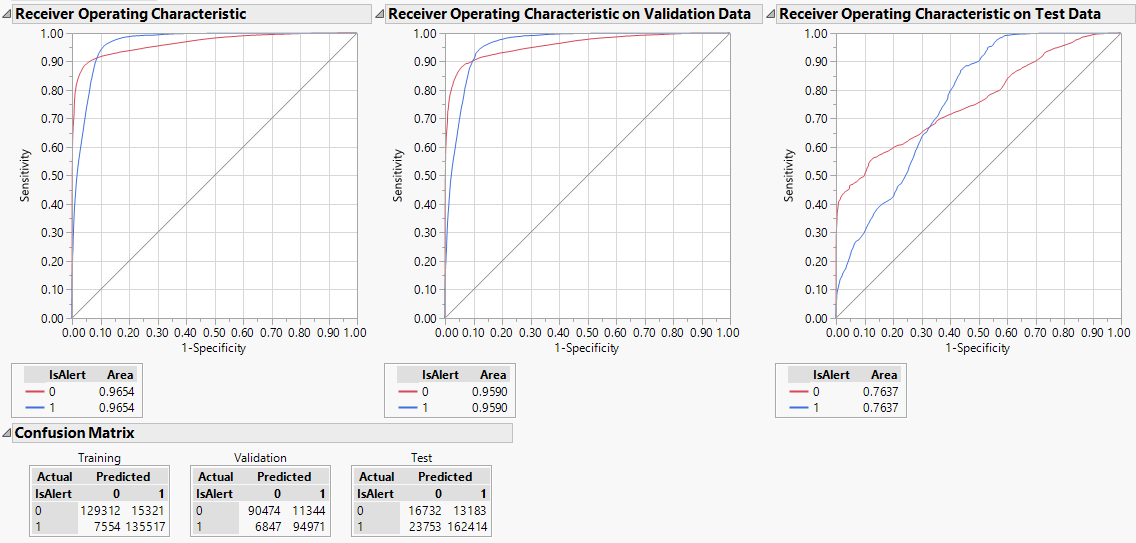


*Figure 4.31*

The ROC curve shows a significant decrease in the Test dataset compared to Training and Validation dataset which also indicates a probability of overfitting of the model.



*Figure 4.32*



*Figure 4.33*

From the above confusion matrix, we can calculate the following:

Test Dataset

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 94.72% | 93.28% | 87.24% |
| **True Negatives** | 89.40% | 88.86% | 55.93% |
| **False Positives** | 10.60% | 11.14% | 44.07% |
| **False Negatives** | 5.28% | 6.72% | 12.76% |
| **Model Accuracy** | 92.05% | 91.07% | 82.91% |

*Table 4.15*

### **Inference**

The model accuracy of the Training and Validation set is very high than the Test which shows a potential indicator of overfitting. In case of TPR and SPC also, the Training and Validation is performing well but the Test is not. Let us do the cost analysis of this model now.

**Cost Analysis on Test Data:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cut-off | Alert as Not Alert (1 as 0) | Not Alert as Alert (0 as 1) | Alert as Alert (1 as 1) | Not Alert as Not Alert (0 as 0) | Total Profit of the Model |
| 0.1 | 14437 | 15437 | 171730 | 14478 | **$ 1,524,980.00** |
| 0.2 | 17465 | 15096 | 168702 | 14819 | **$ 1,498,520.00** |
| 0.3 | 19975 | 14309 | 166192 | 15606 | **$ 1,527,020.00** |
| 0.4 | 21875 | 13525 | 164292 | 16390 | **$ 1,567,420.00** |
| 0.5 | 23753 | 13183 | 162414 | 16732 | **$ 1,564,060.00** |
| 0.6 | 27410 | 12885 | 158757 | 17030 | **$ 1,520,720.00** |
| 0.7 | 31985 | 12503 | 154182 | 17412 | **$ 1,467,420.00** |
| 0.8 | 41048 | 11751 | 145119 | 18164 | **$ 1,361,360.00** |
| 0.9 | 71189 | 8758 | 114978 | 21157 | **$ 1,057,840.00** |

*Table 4.16*

The model is at its optimum at a probability cut-off value of 0.4 with the total profit being $1,567,420.

## **Final Evaluation**

We highlight the best features and go for the profit matrix. Assuming, that the cost of the model is given to us by the people who built them.

Cost of predicting Alert as Not Alert (1 as 0) = -$10

Cost of predicting Not Alert as Alert (0 as 1) = -$50

Profit of predicting Alert as Alert (1 as 1) = +$10

Profit of predicting Not Alert as Not Alert (0 as 0) = +$50

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Name** | **Cost of the Model** | **Total Profit from the Model** | **Net Revenue from the Model (Profit – Cost)** |
| Baseline 1(All 0s) | 0 | $2,160,820.00 | $2,160,820.00 |
| Baseline 2(All 1s) | 0 | $10,804,100.00 | $10,804,100.00 |
| Decision Tree @ 0.2 cut-off | $5,000 | $1,496,100.00 | $1,491,100.00 |
| Discriminant Analysis @ 0.4 cut-off | $3,400 | $1,460,260.00 | $1,456,860.00 |
| Neural Network @ 0.1 cut-off | $4,600 | $1,614,130.00 | $1,609,530.00 |
| Nominal Logistic Regression @ 0.3 cut-off | $5,530 | $1,774,580.00 | $1,769,050.00 |
| Ensemble Model 1 @ 0.4 cut-off | $8,800 | $1,502,140.00 | $1,493,340.00 |
| Ensemble Model 2 @ 0.3 cut-off | $9,300 | $1,583,820.00 | $1,574,520.00 |
| Ensemble Model 3 @ 0.5 cut-off | $10,800 | $1,534,080.00 | $1,523,280.00 |
| Ensemble Model 4 @ 0.4 cut-off | $9,000 | $1,567,420.00 | $1,558,420.00 |

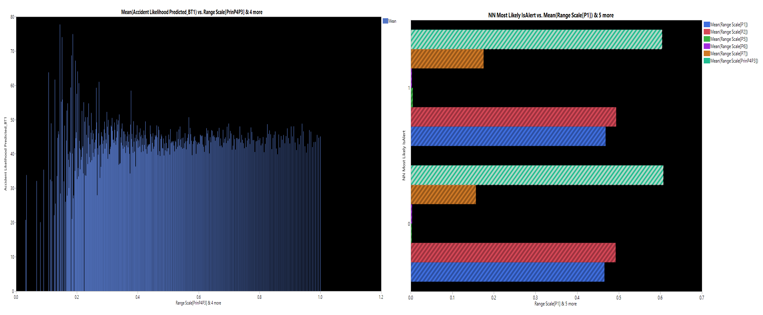
*Table 4.17*

### **Recommendation – Categorical Response**

If we consider the cost of the model, it is very evident that Nominal Logistic Regression Model gives the maximum profit among all at a probability cut-off value of 0.3. We have considered two base models, 1) Baseline 1 – Predicting All 0s, 2) Baseline 2 – Predicting All 1s. Comparing these two baseline models also, we found that Nominal Logistic Regression Model is the most profitable among all other with a highest net revenue of $1,769,050.

# **6.0 Data Visualization and Insights**

**Physiological Factors Vs Predicted Accident Likelihood, IsAlert**

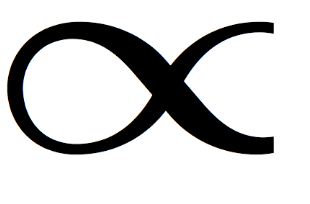


*Figure 5.1*

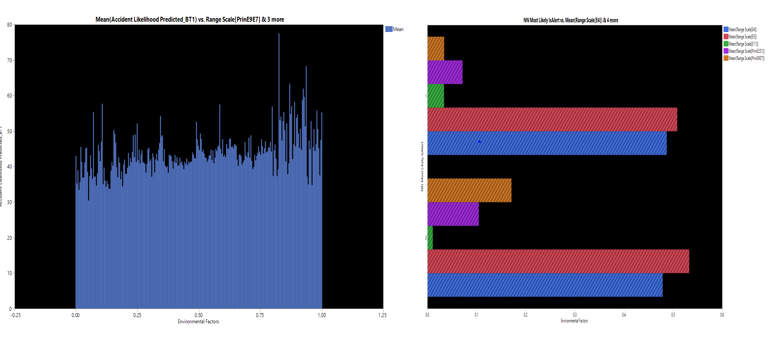
The following are the insights which can be gathered from the above graph which has been plotted for Predicted Accident Likelihood and the physiological parameters

* The Accident Likelihood is likely to be high when the physiological factors are low
* As the values of physiological parameters increases, the Accident likelihood becomes constant

It can be inferred that for the Accident Likelihood to be less, the physiological factor has to be more

**Accident Likelihood**  **1 / Physiological factors**

**Environmental Factors Vs Predicted Accident Likelihood, IsAlert**

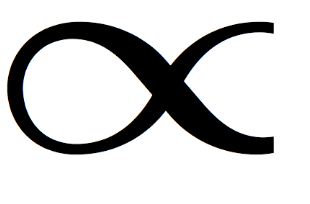


*Figure 5.2*

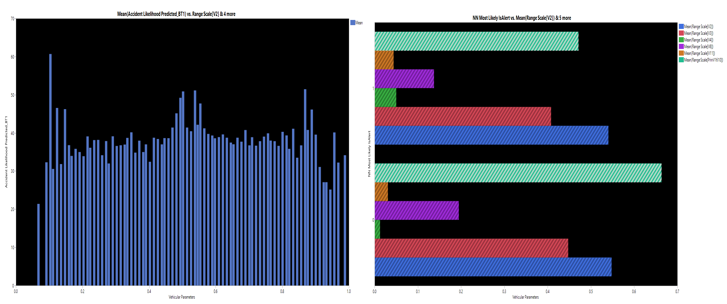
The following are the insights which can be gathered from the above graph which has been plotted for Predicted Accident Likelihood and the environmental parameters

* The Accident Likelihood is likely to be high when the environmental factors are high
* The trend in the graph indicates that the Accident Likelihood is high for environmental factors of value greater than 0.75

It can be inferred that for the Accident Likelihood to be less, the environmental factors have to be lower than 0.75

**Accident Likelihood**  **Environmental factors**

**Vehicular Factors Vs Predicted Accident Likelihood, IsAlert**



*Figure 5.3*

The following are the insights which can be gathered from the above graph which has been plotted for Predicted Accident Likelihood and the vehicular parameters:

* The Accident Likelihood is likely to be high for values of vehicular parameter less than 0.15
* There is a nonlinear trend and any specific trend can’t be estimated

It can be inferred that for the Accident Likelihood to be high, the values of vehicular parameters should be less than 0.15.

# **7.0 Project Conclusion & Recommendation**

The Report recommends Boosted Tree Model for the prediction of the continuous response “Accident Likelihood” and for prediction of the categorical response “IsAlert”, Nominal Logistic Regression model is recommended. Based on the gathered insights from the model, it is evident that, the physiological factors contribute the most to the alertness of a driver. For the driver to be alert, the physiological factors must have a lower value. The business can look at the factors contributing to the physiological parameters in order to bring in policies and standards which can reduce the values for physiological parameters. With a combination of different models, we were able to predict the alertness of the driver using the given parameters. The following enlisted are the business recommendations

* + Business should aim to introduce strict policies and implements standards which can contribute to a lower value for the physiological factors as it is inversely proportional to the alertness of the driver
  + The Environmental factors having values less than 0.75 yields an alert driver. Hence, the business should analyze the environmental factors to learn on the practical causes for the same
  + For a better and detailed understanding of the factors contributing to the alertness of a driver, we recommend the business to provide us with a data dictionary

# 

# **8.0 Appendix**

This section documents all the other models which were developed as part of this project

# **Data Models for Accident Likelihood - Dataset without Outliers**

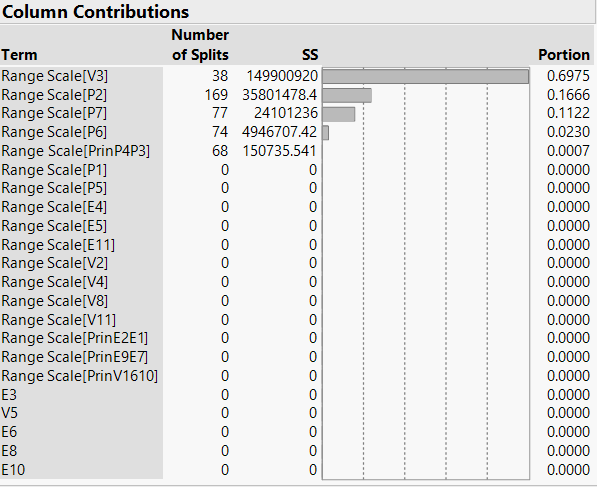
# **Overview**

This section provides models of the dataset without outliers. The following are the models which would be built and illustrated in this section

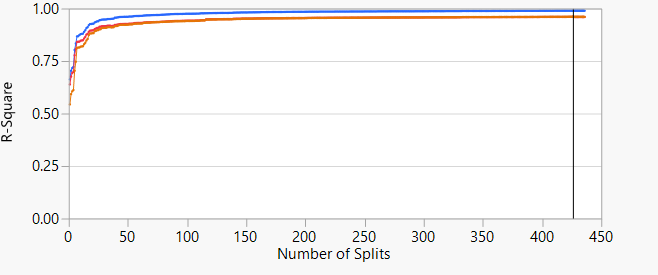
1. Decision Tree
2. Step wise
3. Standard Least Squares
4. Neural Nets
5. Ensemble Model

# **Decision Tree**

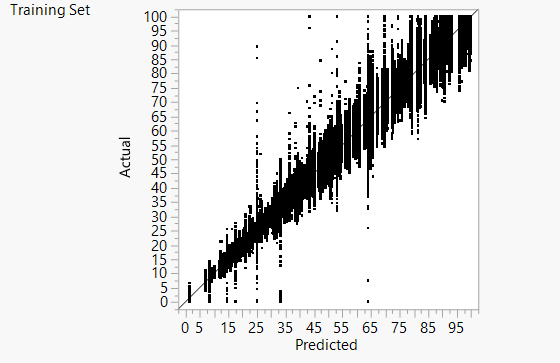
# **Column Contributions**

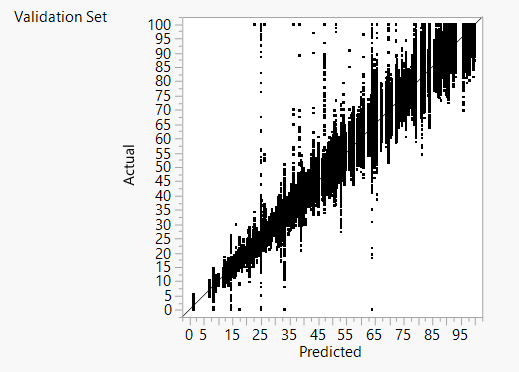


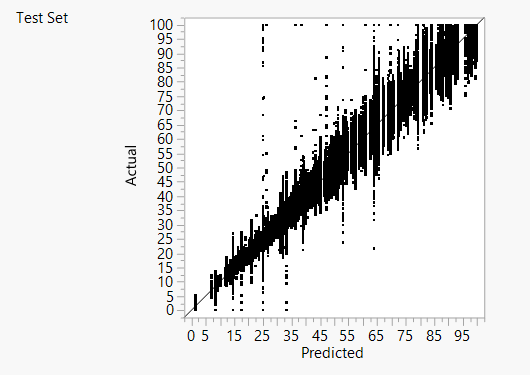
# **Split History**

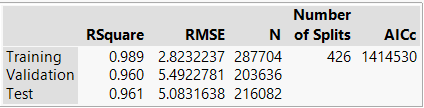


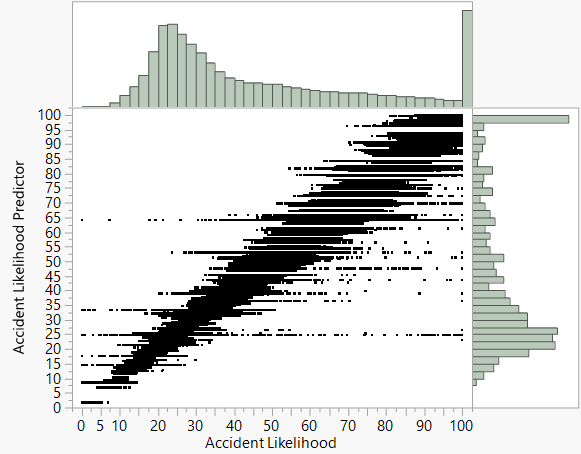
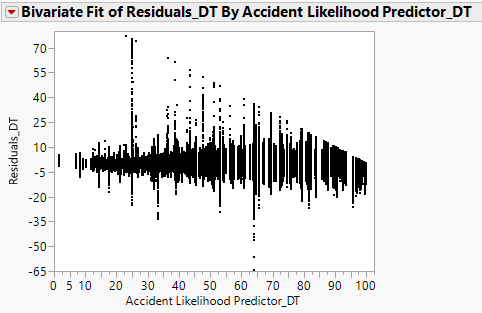
# **Actual by Predicted Plot**

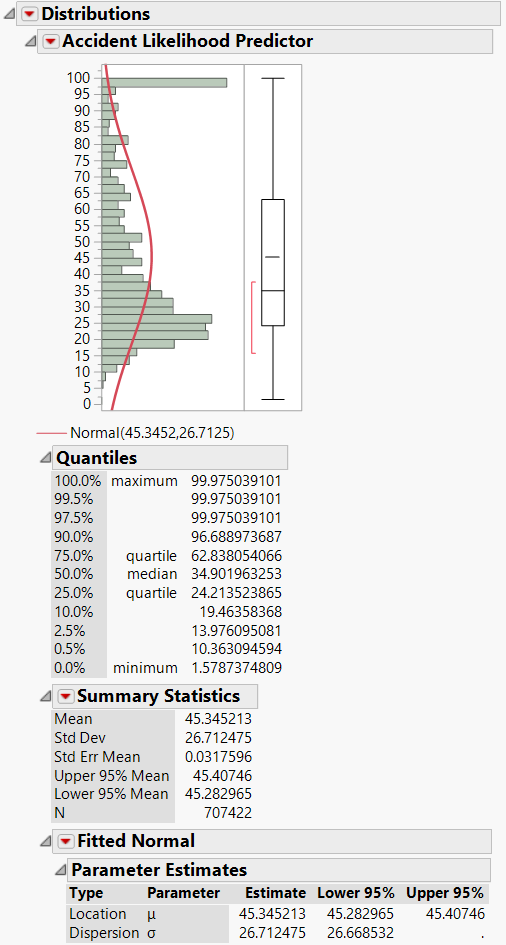








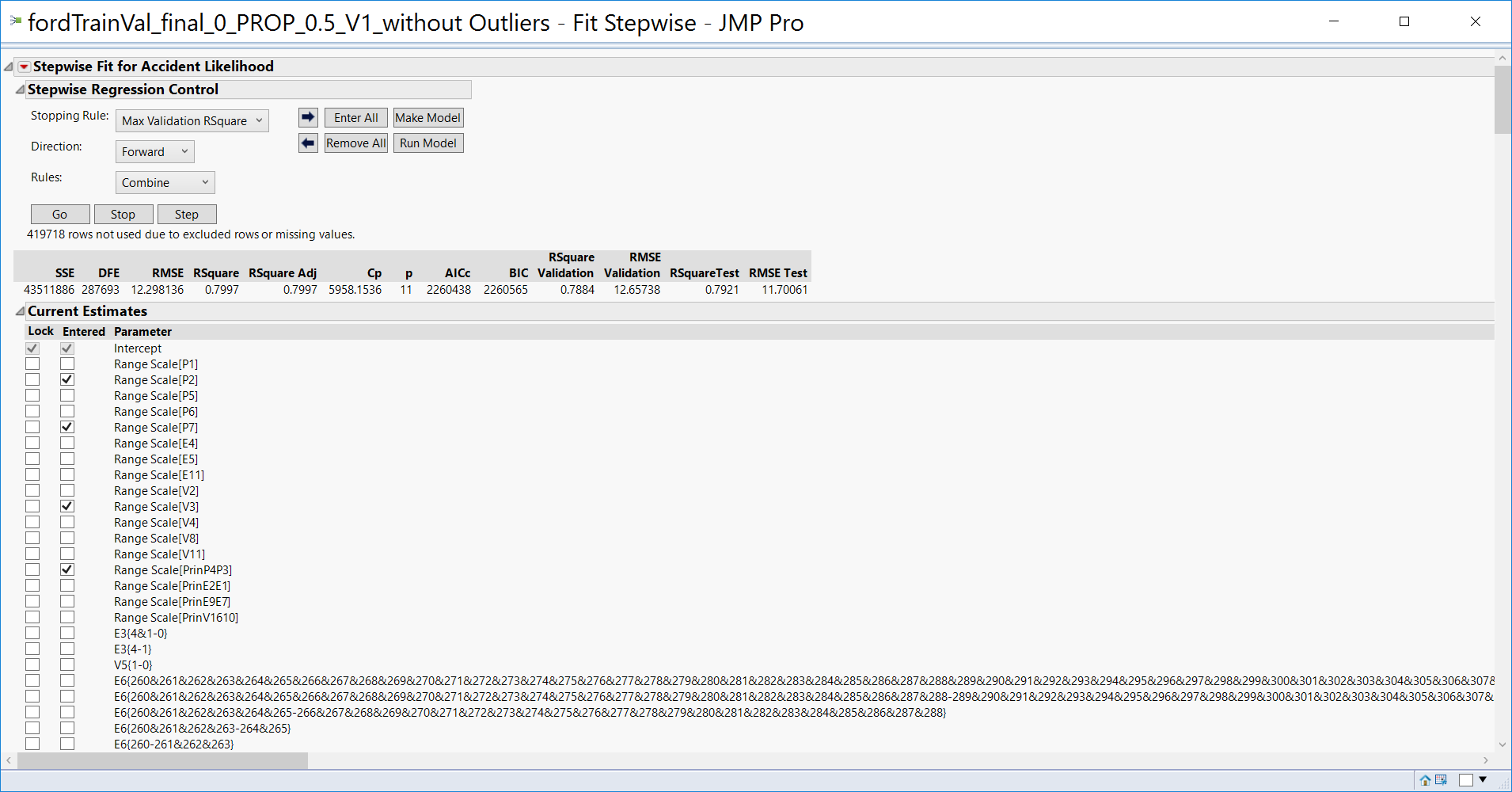


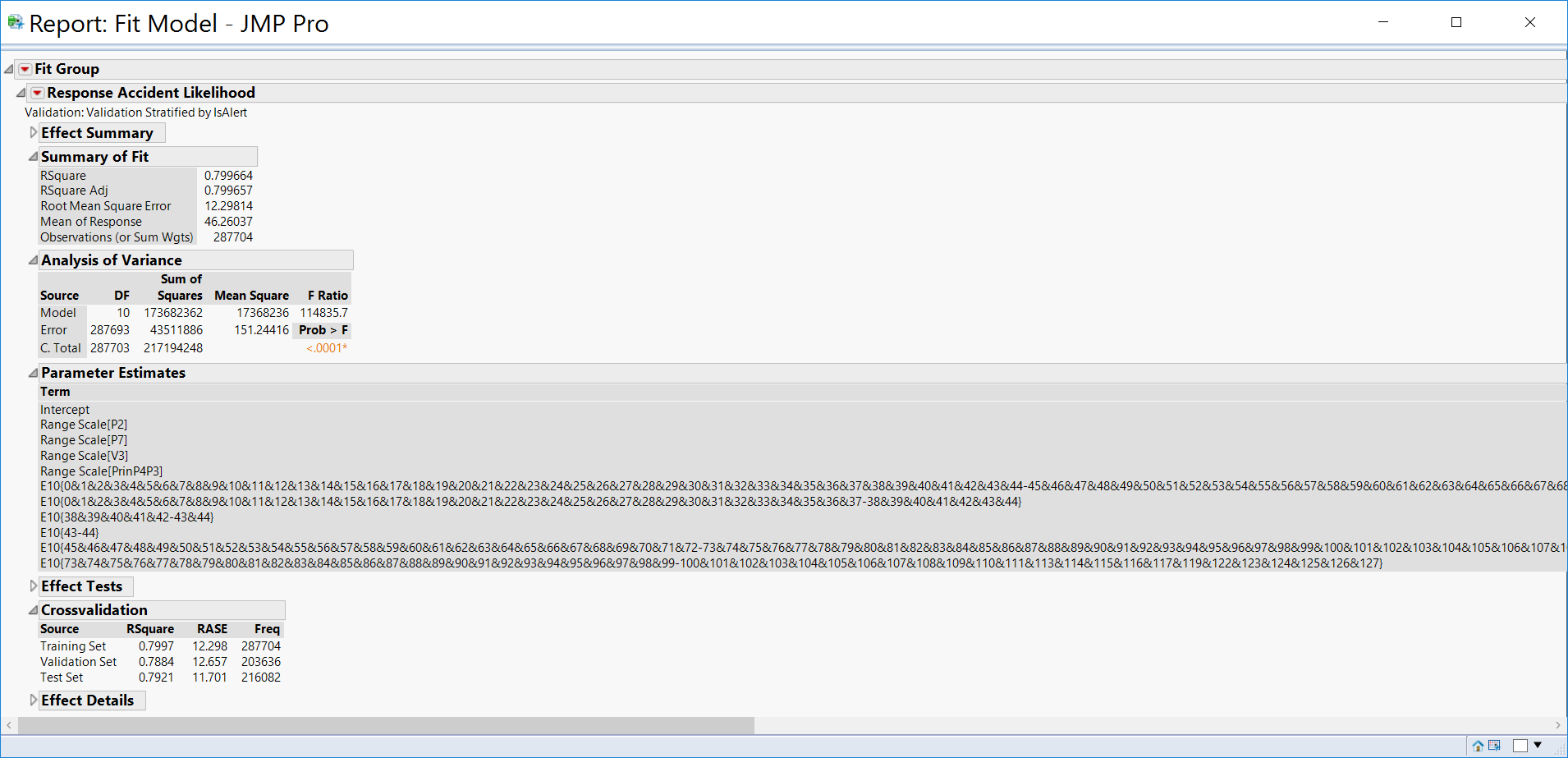


# **Inferences**

1. Accident Likelihood is most influenced by the parameter “Range Scale[V3]”
2. Other parameters which have marginal influence on Accident Likelihood are “Range Scale[P2]”, “Range Scale[P7]”, “Range Scale[P6]”, “Range Scale[PrinP4P3]”
3. The rest parameters do not influence the Accident Probability

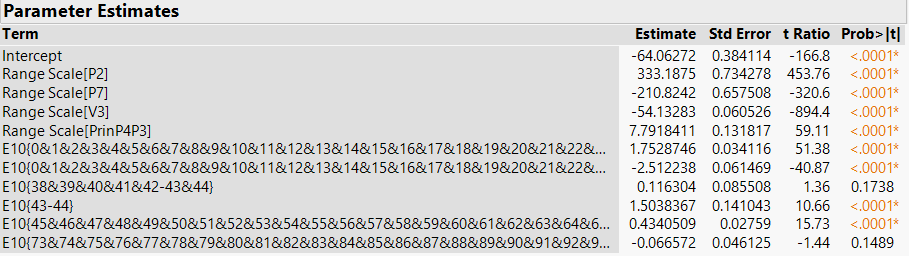
# **Stepwise Model**



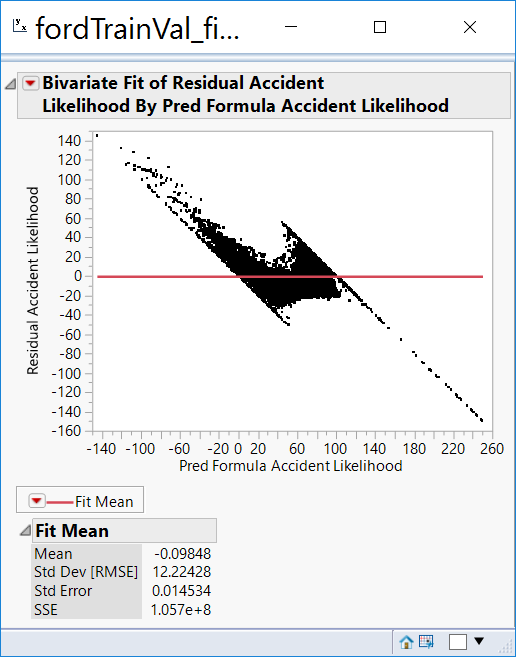


The parameters in the dataset, selected for this model contributes to 80% of the Accident Likelihood.

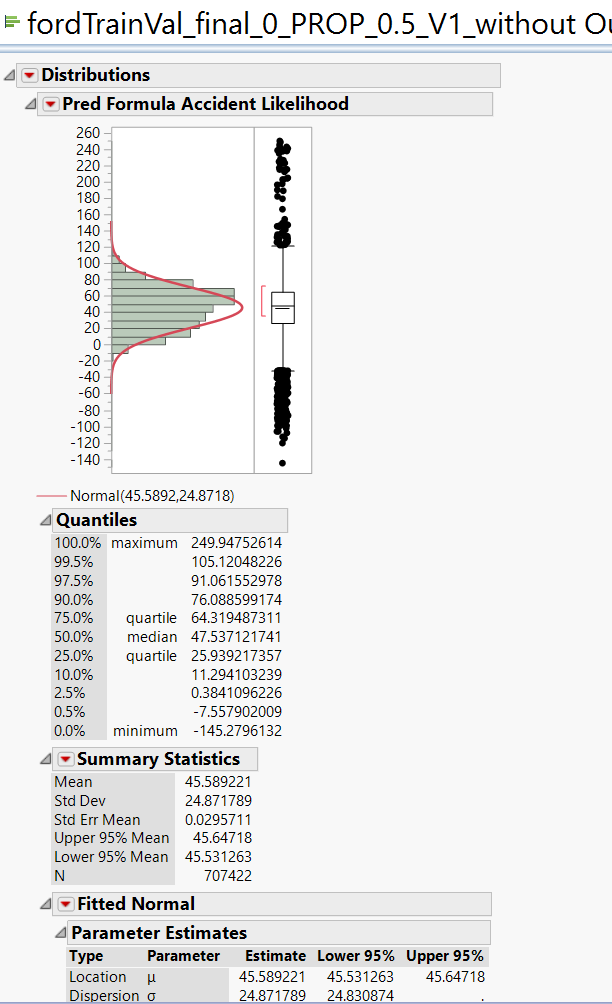
# **Parameter Estimates**



# **Residual Accident Likelihood vs Predicted Accident Likelihood**



# **Distribution Analysis**

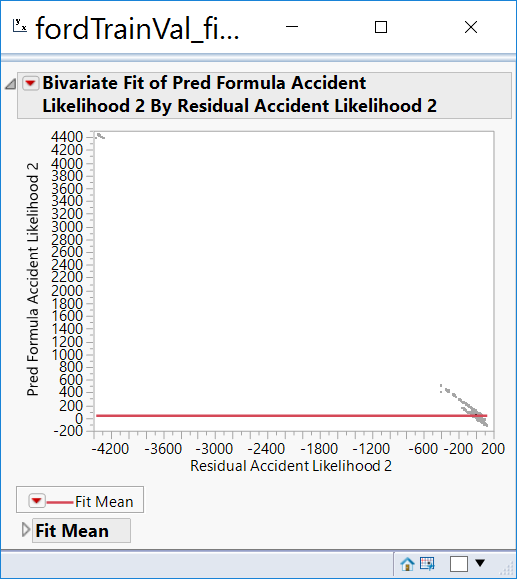


# **Standard Least Squares Model**

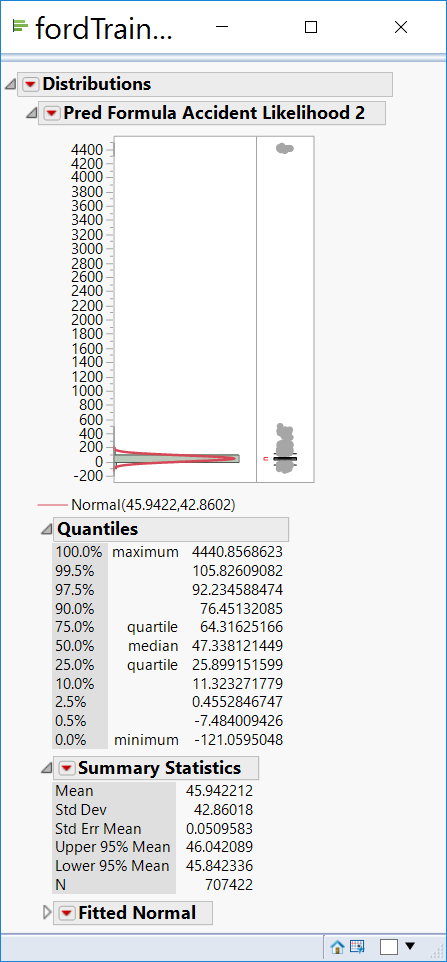


The parameters in the dataset, selected for this model contributes to 80% of the Accident Likelihood.

# **Residual Accident Likelihood vs Predicted Accident Likelihood**



# **Distribution Analysis**



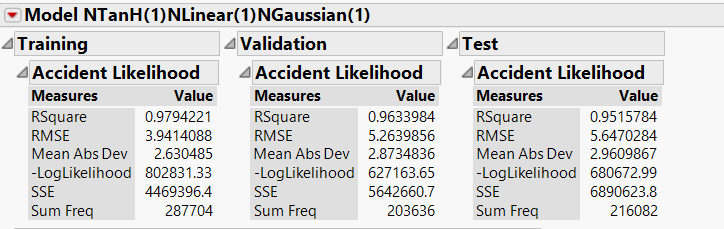
# **Inferences**

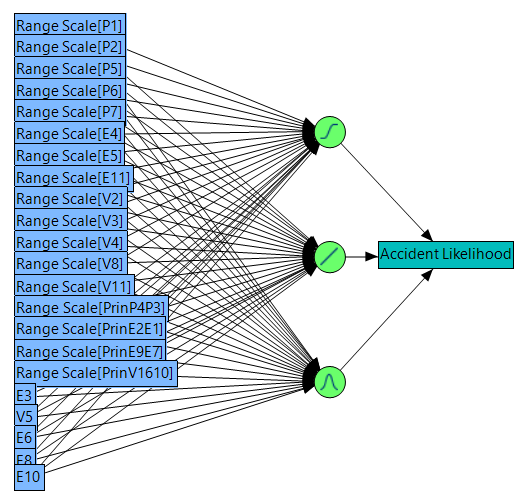
The following are the inferences from the model

1. Inconsistent variation in the Residual Plot
2. Presence of Outliers in the predicted values

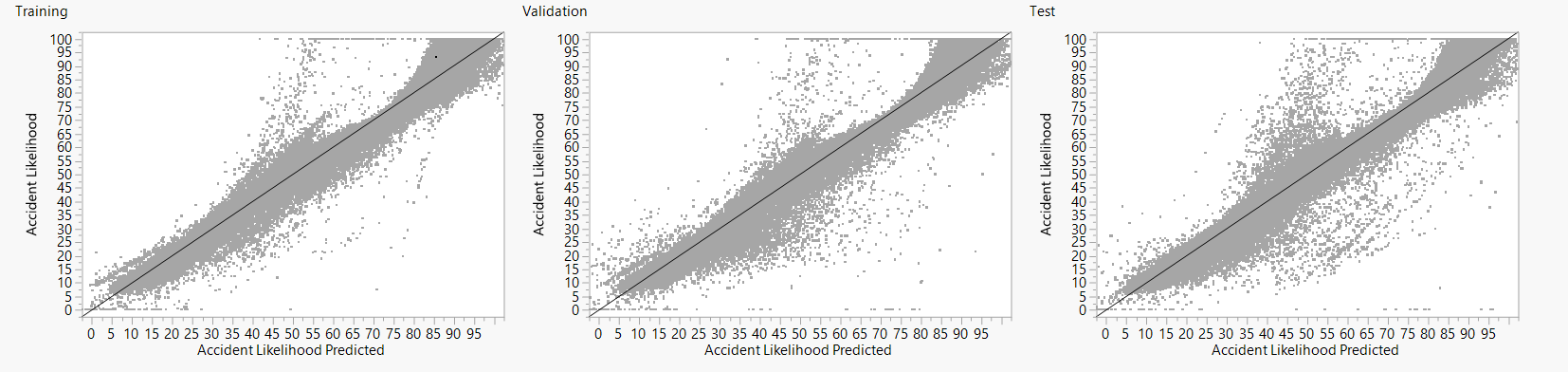
The parameters assumed to contribute to the Accident Likelihood may not be valid based on this model

# **Neural Nets Model**

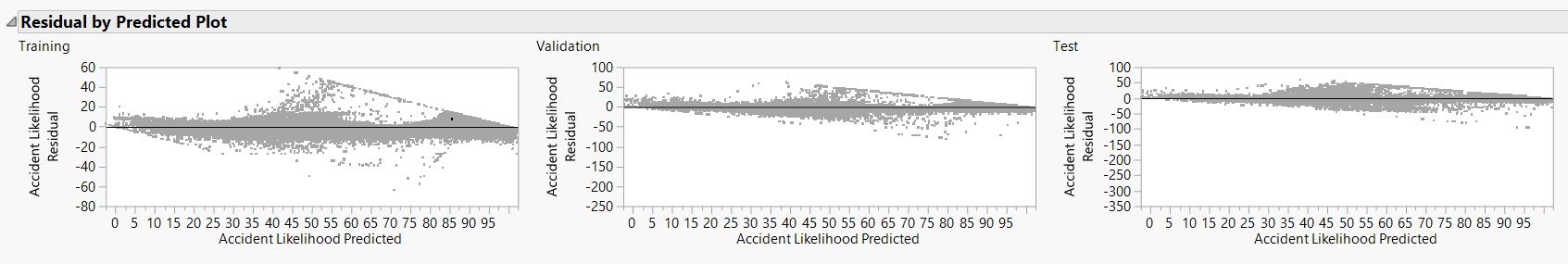


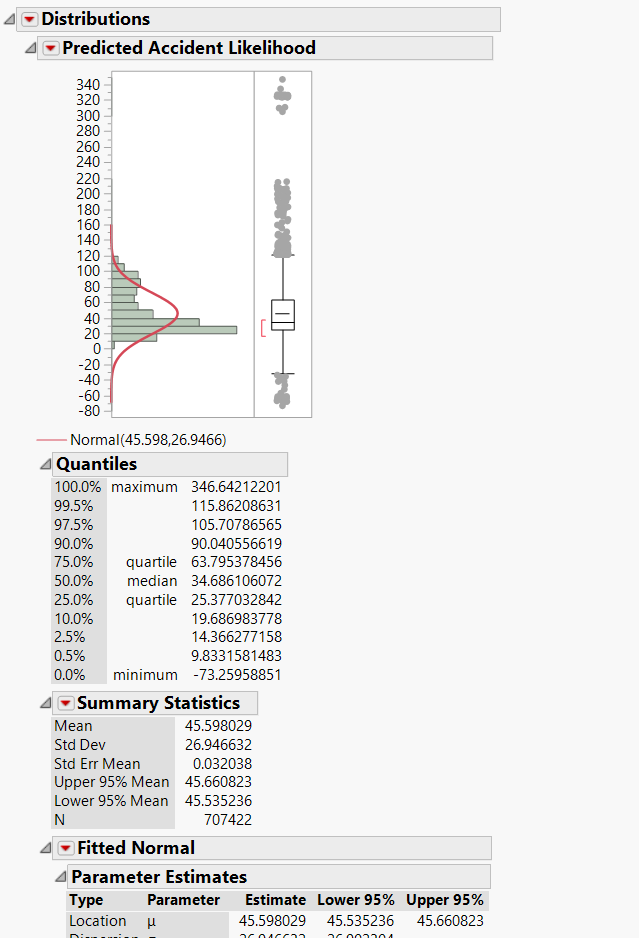


# **Actual by Predicted Plot**



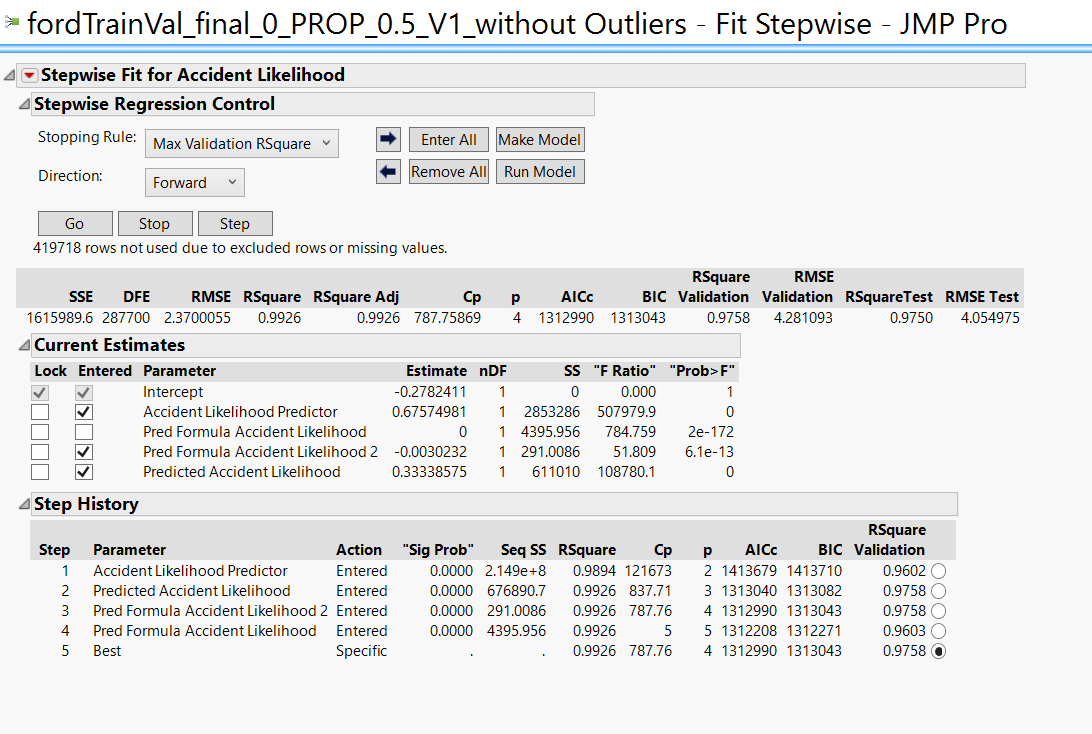
# **Residual by Predicted Plot**

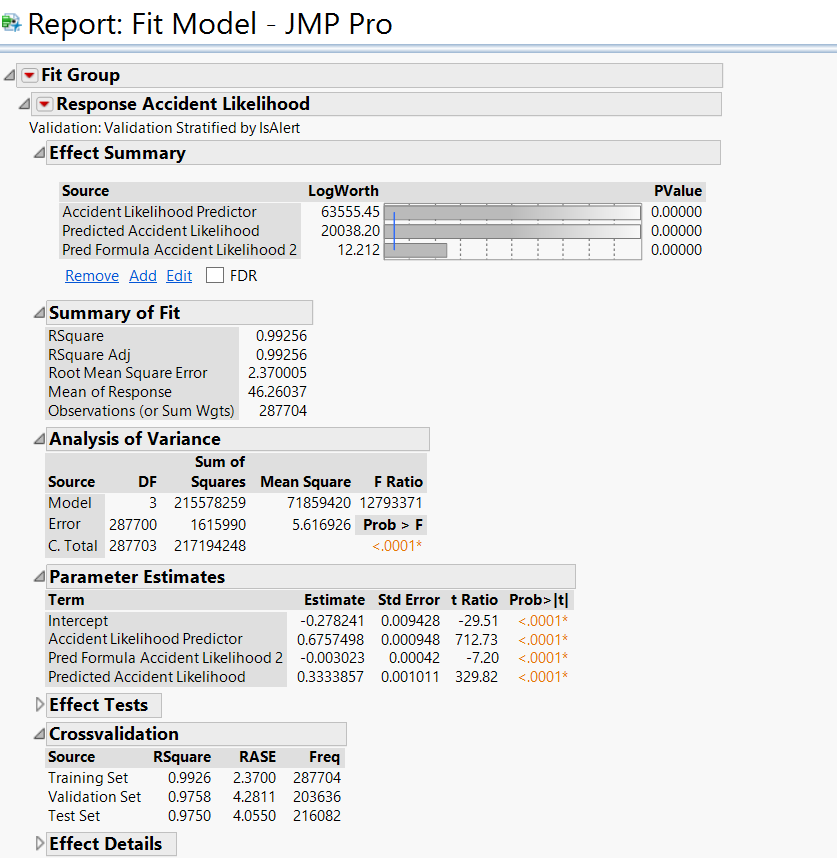
**Distribution Analysis**



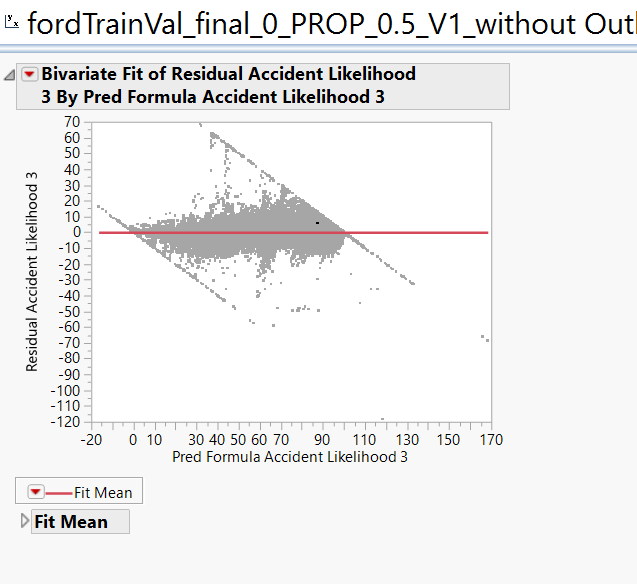
# **Ensemble Model**

(DT+SW+LS+NT on SW)

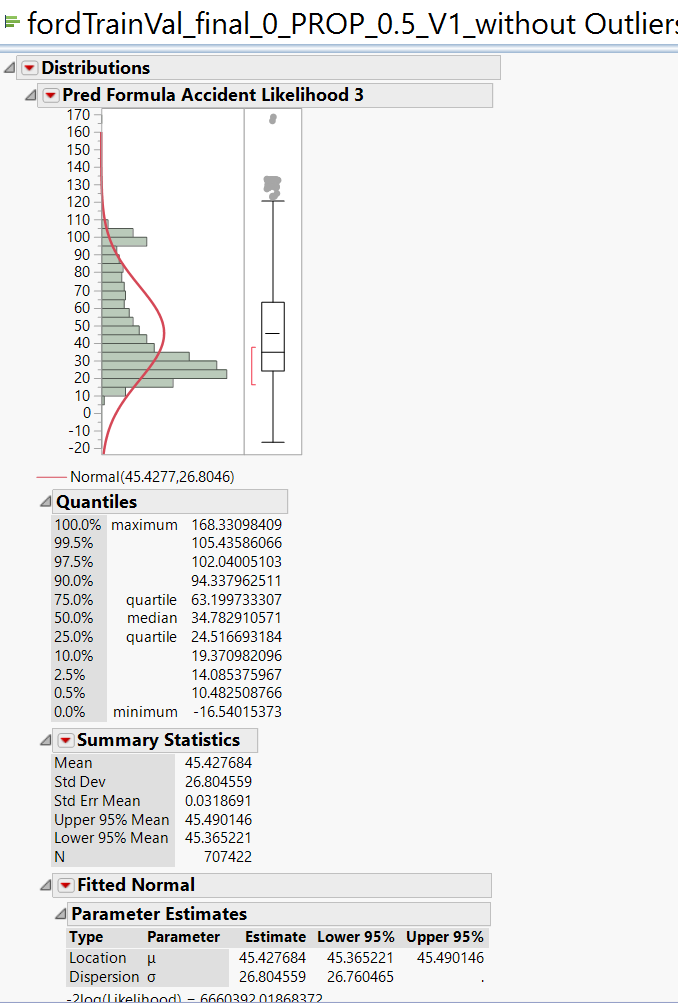




# **Residual Plot**



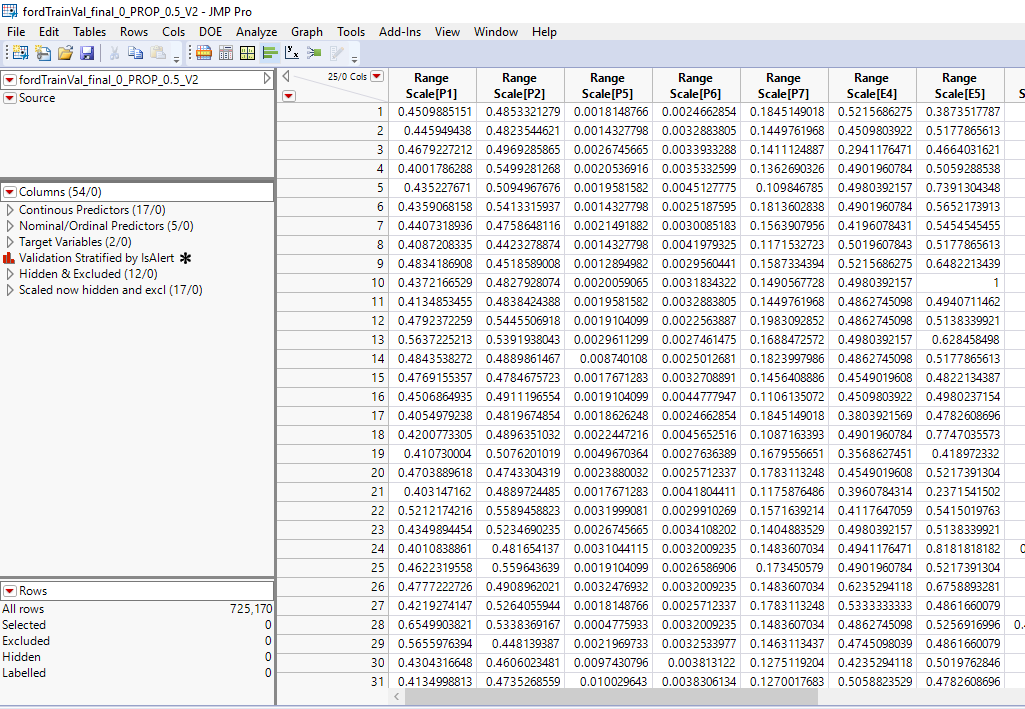
# **Distribution Analysis**



# **Conclusion**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Cost of the model** | **Count of Out of threshold Errors** | **Cost of Out of threshold Errors** | **Count of within threshold predictions** | **Benefit of within threshold predictions** | **Count of Max/Min Errors** | **Cost of Max/Min Errors** | **Net Revenue** | **RMSE** | **Residual Plot** |
|  | **(A)** |  | **(B)** |  | **(C')** |  | **(D)** | **A+B+C+D** |  |  |
| BASELINE | $0 | 195900 | -$587,700.00 | 20182 | $141,274.00 | 0 | $0.00 | -$446,426.00 | FAIL | FAIL |
| Decision Tree | $5,000 | 63926 | -$191,778.00 | 152156 | $1,065,092.00 | 0 | $0.00 | $878,314.00 | PASS | FAIL |
| Step wise | $7,500 | 63926 | -$191,778.00 | 152156 | $1,065,092.00 | 9906 | -$9,906.00 | $870,908.00 | FAIL | FAIL |
| Standard Least Squares | $8,000 | 63780 | -$191,340.00 | 152302 | $1,066,114.00 | 9800 | -$9,800.00 | $872,974.00 | FAIL | FAIL |
| Neural Nets | $10,000 | 19290 | -$54,870.00 | 197792 | $1,384,544.00 | 8678 | -$8,678.00 | $1,320,996.00 | PASS | PASS |
| Ensemble Model | $12,000 | 14353 | -$43,059.00 | 201729 | $1,412,103.00 | 8616 | -$8,616.00 | $1,372,428.00 | PASS | FAIL |

## **Model Analysis of Nominal Response (with Outliers)**



We begin with grouping all columns such that, running a model is easier and organized. We are going to run a total of 8 classification models.

* Decision Tree
* Neural Networks
* Discriminant Analysis
* Nominal Logistic Regression
* Ensemble – 1 to 4

### **Model Tweaking**

For every nominal model, we do the following tweaks to fine tune our model and run multiple iterations to get better results.

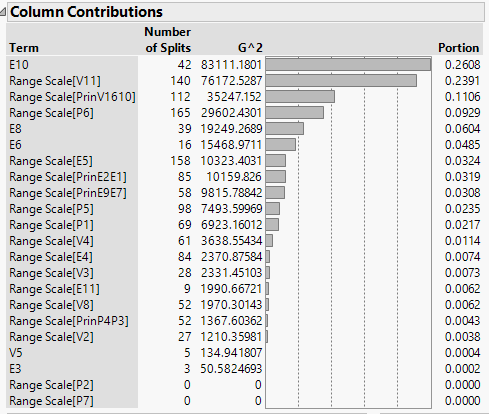
1. Remove the variable from the predictors that do not have any variation i.e. do not add any predictive value
2. Choose all the variables that are significant.

### **Decision Tree**

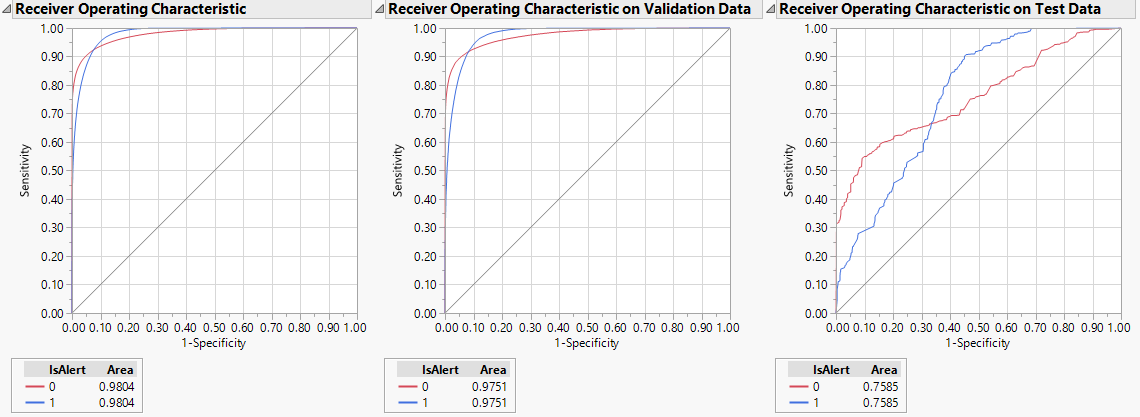
We run the Decision Tree model with the following parameters. Also, as discussed in the pre-processing step, we are considering variables E3, E6, E8, E10 and V5 as nominal/ordinal predictors and later compare model performance with the time when we use them as continuous predictors. We get the following huge small tree view:

## **Observations**

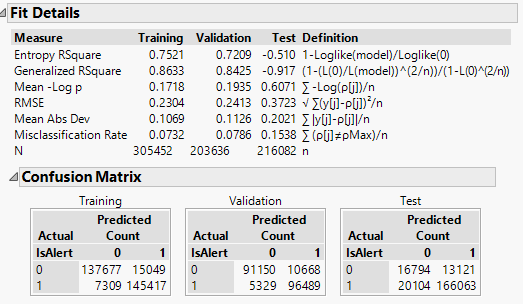
The column contributions are as follows:



We notice here that Physiological variables like P2 and P7 are not statistically significant enough to contribute to the column split. It’s safe to say that these variables do not add predictive power since there is no contribution from them even after 1,298 splits.



The above ROC curve shows that the model does significantly well in both validation and training. However, the curve is not the same level when it comes to test. The average is much lesser than training and validation. In order to verify the fit, we look at the confusion matrix of the model as shown below:



From above we can calculate the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 95.21% | 94.76% | 89.20% |
| **True Negatives** | 90.14% | 89.52% | 56.13% |
| **False Positives** | 9.85% | 10.47% | 43.86% |
| **False Negatives** | 4.78% | 5.23% | 10.79% |
| **Model Accuracy** | 92.68% | 92.14% | 84.62% |

## **Inference**

We notice that the model accuracy is not very significantly different from the training and validation accuracies. Hence, we can try and keep this as a valid model. We now calculate the cost of the model at various probability cut-offs for the test data:

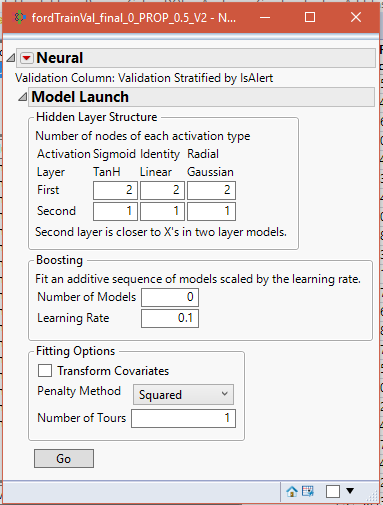
Cost Analysis on Test Data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cut-off** | **Alert as Not Alert (1 as 0)** | **Not Alert as Alert (0 as 1)** | **Alert as Alert (1 as 1)** | **Not Alert as Not Alert (0 as 0)** | **Total Profit of the Model** |
| 0.1 | 62395 | 9838 | 123772 | 20077 | **$ 1,125,720.00** |
| 0.2 | 46183 | 10677 | 139984 | 19238 | **$ 1,366,060.00** |
| 0.3 | 32688 | 11546 | 153479 | 18369 | **$ 1,549,060.00** |
| 0.4 | 25816 | 12447 | 160351 | 17468 | **$ 1,596,400.00** |
| 0.5 | 20101 | 13121 | 166066 | 16794 | **$ 1,643,300.00** |
| 0.6 | 17582 | 14091 | 168585 | 15824 | **$ 1,596,680.00** |
| 0.7 | 15604 | 15079 | 170563 | 14836 | **$ 1,537,440.00** |
| 0.8 | 14187 | 15526 | 171980 | 14389 | **$ 1,521,080.00** |
| 0.9 | 10748 | 16554 | 175419 | 13361 | **$ 1,487,060.00** |

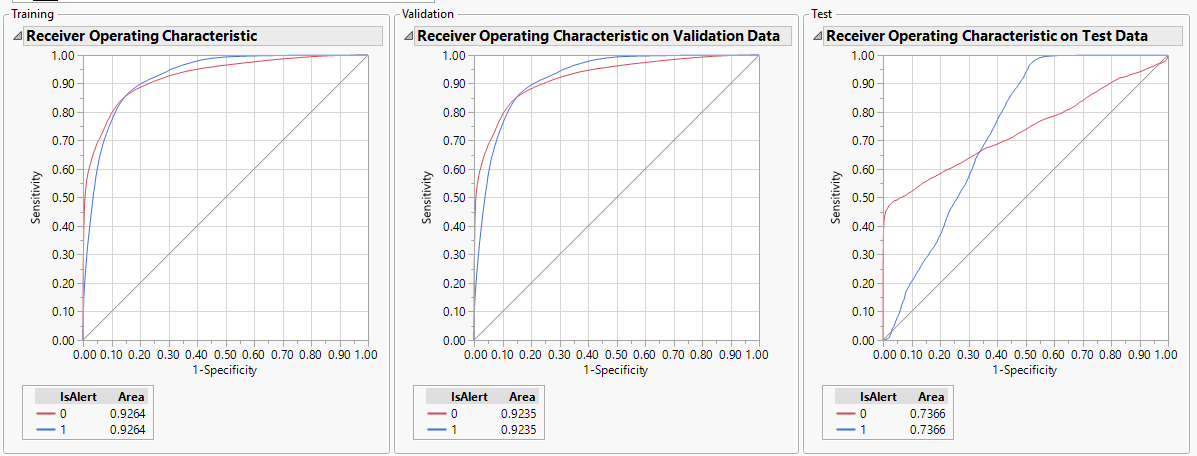
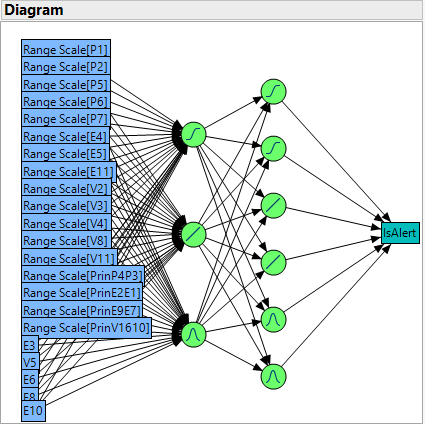
Conclusion: The model is at its optimum at a probability cut-off of 0.5 with the total profit being $1,643,300

### **Neural Networks**

We run the Neural Networks model with the following parameters. Since the Decision Tree model itself was very complex, we try to reduce the number of filters for the stages of model in order to reduce complexity a little bit.



## **Observations**



From the ROC curve it can be seen that this model’s results are almost same as that of Decision Tree but the curve in Test is much smoother than it was in the previous model. The confusion matrix of the model is as shown below:



From above we can calculate the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 87.01% | 86.87% | 83.90% |
| **True Negatives** | 83.83% | 83.54% | 56.36% |
| **False Positives** | 16.16% | 16.45% | 43.64% |
| **False Negatives** | 12.98% | 13.12% | 16.10% |
| **Model Accuracy** | 85.42% | 85.20% | 80.09% |

## **Inference**

We notice that the model accuracy is not very significantly different from the training and validation accuracies. However, the accuracy is more consistent than that of Decision Trees. Hence, we can try and keep this also as a valid model. We now calculate the cost of the model at various probability cut-offs for the test data:

Cost Analysis on Test Data:

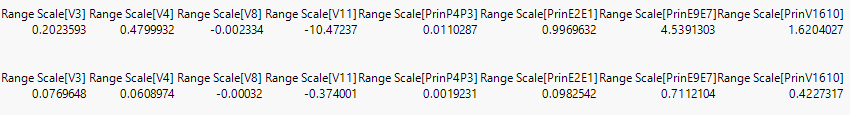
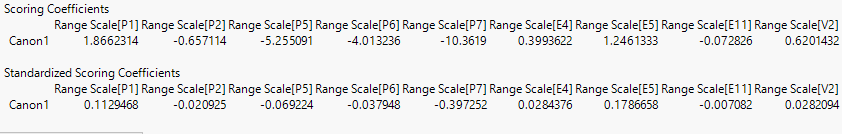
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cut-off** | **Alert as Not Alert (1 as 0)** | **Not Alert as Alert (0 as 1)** | **Alert as Alert (1 as 1)** | **Not Alert as Not Alert (0 as 0)** | **Total Profit of the Model** |
| 0.1 | 2289 | 16232 | 183878 | 13683 | **$ 1,688,440.00** |
| 0.2 | 5139 | 15595 | 181028 | 14320 | **$ 1,695,140.00** |
| 0.3 | 13723 | 14802 | 172444 | 15113 | **$ 1,602,760.00** |
| 0.4 | 21566 | 13947 | 164601 | 15968 | **$ 1,531,400.00** |
| 0.5 | 29975 | 13054 | 156192 | 16861 | **$ 1,452,520.00** |
| 0.6 | 39778 | 12184 | 146389 | 17731 | **$ 1,343,460.00** |
| 0.7 | 52906 | 11109 | 133261 | 18806 | **$ 1,188,400.00** |
| 0.8 | 76336 | 9197 | 109831 | 20718 | **$ 911,000.00** |
| 0.9 | 119233 | 5836 | 66934 | 24079 | **$ 389,160.00** |

Conclusion: The model is at its optimum at a probability cut-off of 0.2 with the total profit being $1,695,140

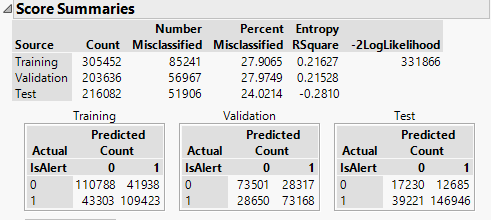
### **Discriminant Analysis**

We run the Discriminant model with the same parameters. Here, we do not include the predictor variables which are not continuous since the model does not allow them. So, we have lesser number of predictor variables for this model.

## **Observations**



The score summaries and confusion matrix of the model is as shown below:



From above we can calculate the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 71.65% | 71.86% | 78.93% |
| **True Negatives** | 72.54% | 72.19% | 57.60% |
| **False Positives** | 27.46% | 27.81% | 42.40% |
| **False Negatives** | 28.35% | 28.14% | 21.07% |
| **Model Accuracy** | 72.09% | 72.03% | 75.98% |

## **Inference**

We notice that the model accuracy is actually better for test than training and validation. However, the accuracy is lesser than that of Decision Trees and Neural Networks. We can try and decide whether this is a good model or not by calculating the cost of the model at various probability cut-offs for the test data:

Cost Analysis on Test Data:

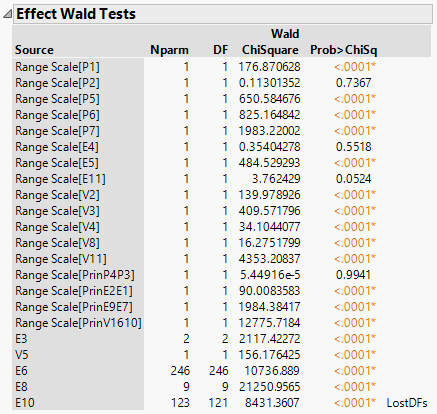
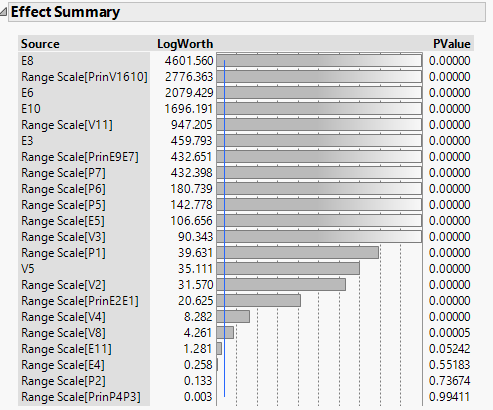
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cut-off** | **Alert as Not Alert (1 as 0)** | **Not Alert as Alert (0 as 1)** | **Alert as Alert (1 as 1)** | **Not Alert as Not Alert (0 as 0)** | **Total Profit of the Model** |
| 0.1 | 181968 | 209 | 4199 | 29706 | **$ -302,840.00** |
| 0.2 | 150742 | 2023 | 35425 | 27892 | **$ 140,280.00** |
| 0.3 | 116164 | 6941 | 70003 | 22974 | **$ 340,040.00** |
| 0.4 | 74936 | 10011 | 111231 | 19904 | **$ 857,600.00** |
| 0.5 | 39221 | 12685 | 146946 | 17230 | **$ 1,304,500.00** |
| 0.6 | 13501 | 15249 | 172666 | 14666 | **$ 1,562,500.00** |
| 0.7 | 2082 | 17712 | 184085 | 12203 | **$ 1,544,580.00** |
| 0.8 | 451 | 24749 | 185716 | 5166 | **$ 873,500.00** |
| 0.9 | 144 | 28553 | 186023 | 1362 | **$ 499,240.00** |

Conclusion: The model is at its optimum at a probability cut-off of 0.6 with the total profit being $1,562,500.

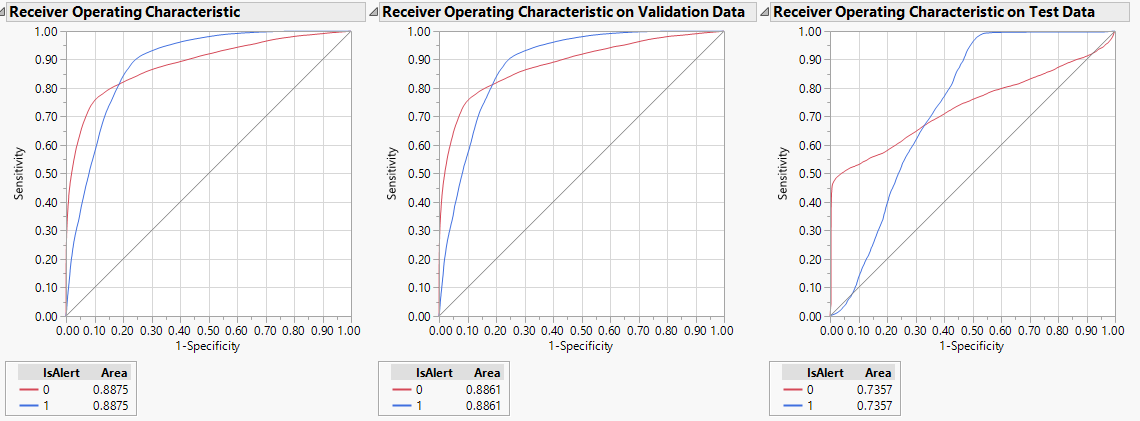
### **Nominal Logistic Regression**

We run the Nominal Logistic Regression model here since our target variable ‘IsAlert’ is nominal. Once again, we include all the predictor variables and use them as an add.

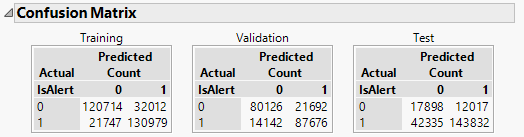
## **Observations**



We notice here that some columns which were not significant in earlier models are significant here and vice versa.



The ROC curve suggests that the test performance is slightly less than training and validation. Following is the confusion matrix for this model:



From above we can calculate the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 85.76% | 86.11% | 77.26% |
| **True Negatives** | 79.04% | 78.70% | 59.83% |
| **False Positives** | 20.96% | 21.30% | 40.17% |
| **False Negatives** | 14.24% | 13.89% | 22.74% |
| **Model Accuracy** | 82.40% | 82.40% | 74.85% |

## **Inference**

Although the model accuracy in test is almost consistent with the training and validation, it has a huge false negative percentage. So, the best way to decide the goodness of this model would be through cost analysis:

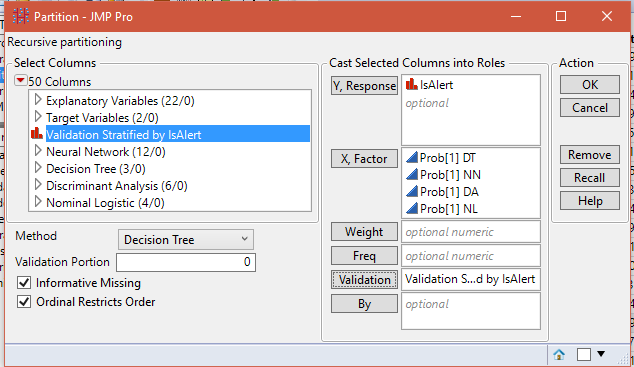
Cost Analysis on Test Data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cut-off** | **Alert as Not Alert (1 as 0)** | **Not Alert as Alert (0 as 1)** | **Alert as Alert (1 as 1)** | **Not Alert as Not Alert (0 as 0)** | **Total Profit of the Model** |
| 0.1 | 1616 | 16058 | 184551 | 13857 | **$ 1,719,300.00** |
| 0.2 | 5834 | 15141 | 180333 | 14774 | **$ 1,726,640.00** |
| 0.3 | 13198 | 14353 | 172969 | 15562 | **$ 1,658,160.00** |
| 0.4 | 25745 | 13351 | 160422 | 16564 | **$ 1,507,420.00** |
| 0.5 | 42335 | 12017 | 143832 | 17898 | **$ 1,309,020.00** |
| 0.6 | 59194 | 10154 | 126973 | 19761 | **$ 1,158,140.00** |
| 0.7 | 80704 | 8084 | 105463 | 21831 | **$ 934,940.00** |
| 0.8 | 110520 | 6101 | 75647 | 23814 | **$ 536,920.00** |
| 0.9 | 145709 | 4087 | 40458 | 25825 | **$ 34,390.00** |

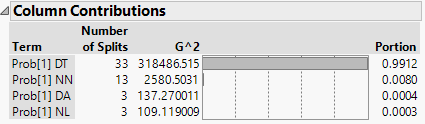
Conclusion: The model is at its optimum at a probability cut-off of 0.2 with the total profit being $1,726,640

### **Ensemble Model 1**

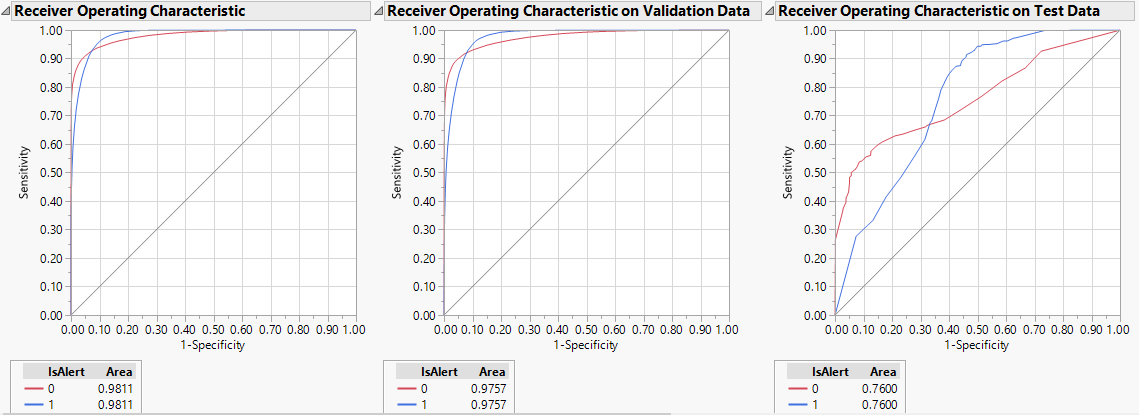
We have seen 4 different models. Now we will try an ensemble of all this models through different iterations and methods. We first begin by using the probability values from the previous 4 models as predictor variables and use it in a Decision Tree



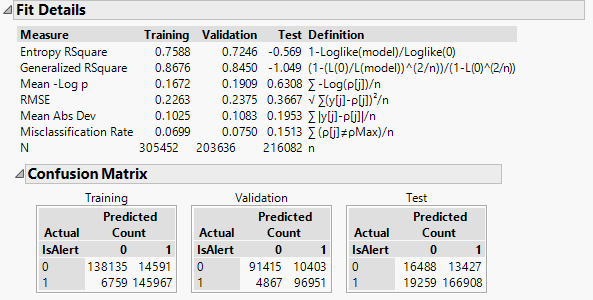
## **Observation**



The probability values of Decision Tree contribute to the maximum number of splits in this model.



The ROC curves are not significantly different from the original Decision Tree model. As seen, the test data is slightly underperforming when compared to training and validation. The confusion matrix is shown below:



From above we can calculate the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 95.57% | 95.22% | 89.65% |
| **True Negatives** | 90.45% | 89.78% | 55.12% |
| **False Positives** | 9.55% | 10.22% | 44.88% |
| **False Negatives** | 4.43% | 4.78% | 10.35% |
| **Model Accuracy** | 93.01% | 92.50% | 84.87% |

## **Inference**

Model accuracy is almost consistent through training, validation and test. However, the rate of true negatives is on the higher side. So, the best way to decide the goodness of this model would be through cost analysis:

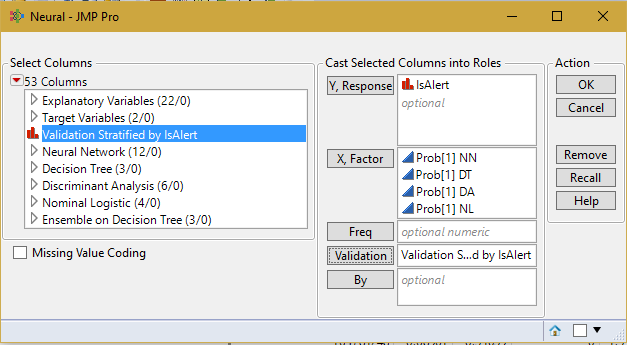
Cost Analysis on Test Data:

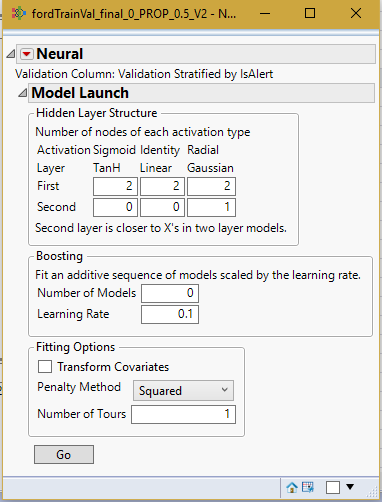
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cut-off** | **Alert as Not Alert (1 as 0)** | **Not Alert as Alert (0 as 1)** | **Alert as Alert (1 as 1)** | **Not Alert as Not Alert (0 as 0)** | **Total Profit of the Model** |
| 0.1 | 59749 | 10042 | 126418 | 19873 | **$ 1,158,240.00** |
| 0.2 | 44529 | 10882 | 141638 | 19033 | **$ 1,378,640.00** |
| 0.3 | 28222 | 12016 | 157945 | 17899 | **$ 1,591,380.00** |
| 0.4 | 23403 | 12710 | 162764 | 17205 | **$ 1,618,360.00** |
| 0.5 | 19259 | 13427 | 166908 | 16488 | **$ 1,629,540.00** |
| 0.6 | 17217 | 13762 | 168950 | 16153 | **$ 1,636,880.00** |
| 0.7 | 15771 | 13848 | 170396 | 16067 | **$ 1,657,200.00** |
| 0.8 | 13354 | 14613 | 172813 | 15302 | **$ 1,629,040.00** |
| 0.9 | 10553 | 15303 | 175614 | 14612 | **$ 1,616,060.00** |

Conclusion: The model is at its optimum at a probability cut-off of 0.7 with the total profit being $1,657,200

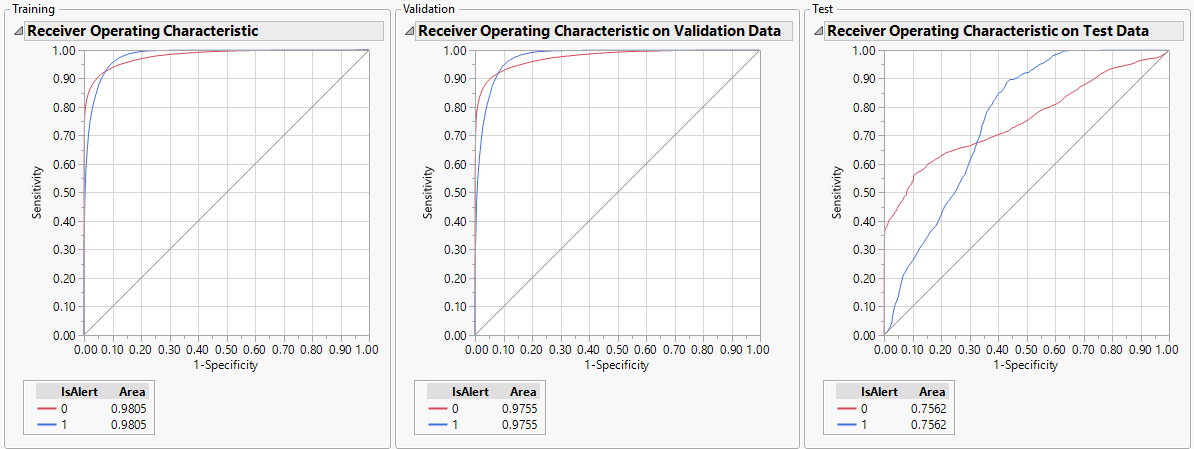
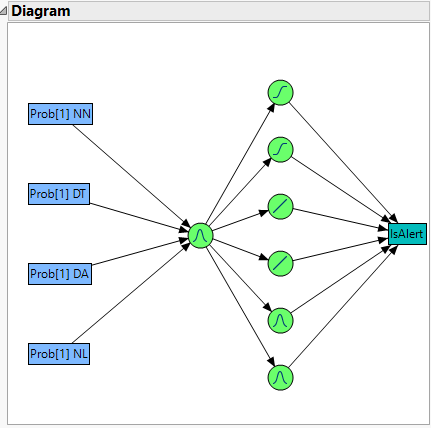
### **Ensemble Model 2**

We have seen 1 ensemble models Next, we use the probability values from the previous 4 models as predictor variables and use it in a Neural Network.

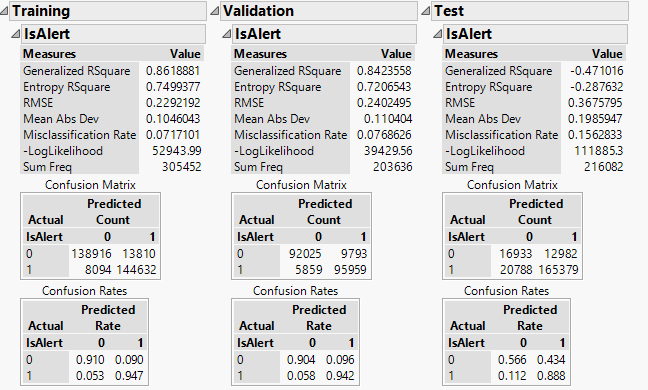




## **Observation**



The ROC curves are not significantly different from the original models. The confusion matrix is shown below:



From above we can calculate the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 94.70% | 94.25% | 88.83% |
| **True Negatives** | 90.96% | 90.38% | 56.60% |
| **False Positives** | 9.04% | 9.62% | 43.40% |
| **False Negatives** | 5.30% | 5.75% | 11.17% |
| **Model Accuracy** | 92.83% | 92.31% | 84.37% |

## **Inference**

Model accuracy is almost consistent through training, validation and test. However, the rate of true negatives and false positives is on the higher side. So, the best way to decide the goodness of this model would be through cost analysis:

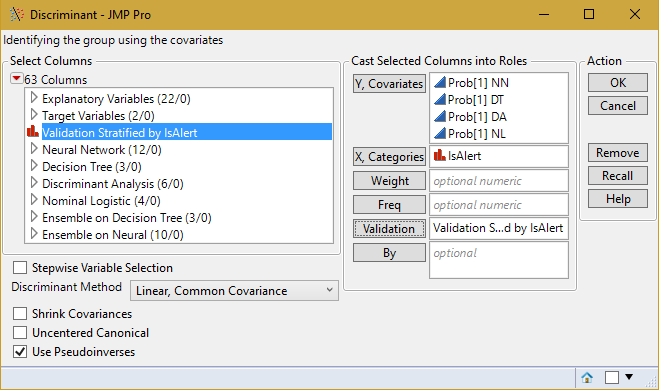
Cost Analysis on Test Data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cut-off** | **Alert as Not Alert (1 as 0)** | **Not Alert as Alert (0 as 1)** | **Alert as Alert (1 as 1)** | **Not Alert as Not Alert (0 as 0)** | **Total Profit of the Model** |
| 0.1 | 10272 | 16147 | 175895 | 13768 | **$ 1,537,280.00** |
| 0.2 | 12813 | 15500 | 173354 | 14415 | **$ 1,551,160.00** |
| 0.3 | 14598 | 15176 | 171569 | 14739 | **$ 1,547,860.00** |
| 0.4 | 16770 | 14244 | 169397 | 15671 | **$ 1,597,620.00** |
| 0.5 | 20788 | 12982 | 165379 | 16933 | **$ 1,643,460.00** |
| 0.6 | 27451 | 12205 | 158716 | 17710 | **$ 1,587,900.00** |
| 0.7 | 34204 | 11400 | 151963 | 18515 | **$ 1,533,340.00** |
| 0.8 | 43762 | 10600 | 142405 | 19315 | **$ 1,422,180.00** |
| 0.9 | 10553 | 15303 | 175614 | 14612 | **$ 1,616,060.00** |

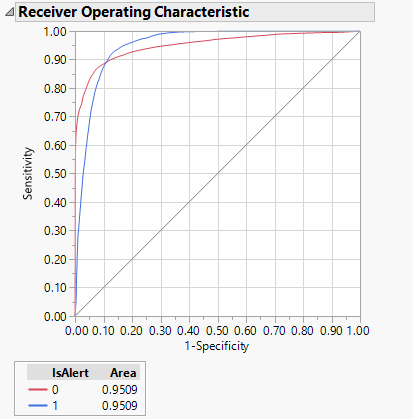
Conclusion: The model is at its optimum at a probability cut-off of 0.5 with the total profit being $1,643,460

### **Ensemble Model 3**

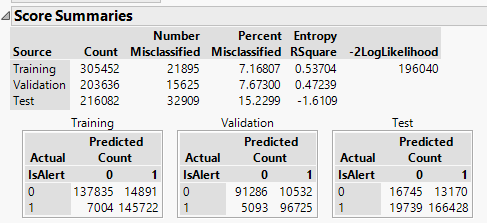
We have seen two ensemble models Next, we use the probability values from the previous 4 models as predictor variables and use it in Discriminant Analysis.



## **Observation**



The confusion matrix is shown below:



From above we can calculate the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 95.41% | 95.00% | 89.40% |
| **True Negatives** | 90.25% | 89.66% | 55.98% |
| **False Positives** | 9.75% | 10.34% | 44.02% |
| **False Negatives** | 4.59% | 5.00% | 10.60% |
| **Model Accuracy** | 92.83% | 92.33% | 84.77% |

## **Inference**

Model accuracy is almost consistent through training, validation and test. However, the rate of false negatives is on the higher side and true negatives is on the lower side. So, the best way to decide the goodness of this model would be through cost analysis:

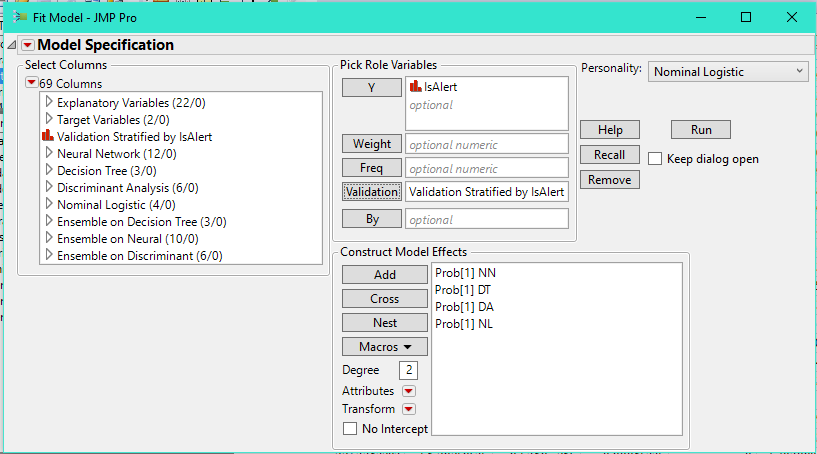
Cost Analysis on Test Data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cut-off** | **Alert as Not Alert (1 as 0)** | **Not Alert as Alert (0 as 1)** | **Alert as Alert (1 as 1)** | **Not Alert as Not Alert (0 as 0)** | **Total Profit of the Model** |
| 0.1 | 17129 | 14197 | 169038 | 15718 | **$ 1,595,140.00** |
| 0.2 | 18019 | 14085 | 168148 | 15830 | **$ 1,588,540.00** |
| 0.3 | 18664 | 13774 | 167503 | 16141 | **$ 1,606,740.00** |
| 0.4 | 19205 | 13270 | 166962 | 16645 | **$ 1,646,320.00** |
| 0.5 | 19739 | 13170 | 166428 | 16745 | **$ 1,645,640.00** |
| 0.6 | 20414 | 13054 | 165753 | 16861 | **$ 1,643,740.00** |
| 0.7 | 21520 | 12946 | 164647 | 16969 | **$ 1,632,420.00** |
| 0.8 | 22998 | 12761 | 163169 | 17154 | **$ 1,621,360.00** |
| 0.9 | 25704 | 12533 | 160463 | 17382 | **$ 1,590,040.00** |

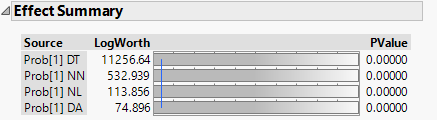
Conclusion: The model is at its optimum at a probability cut-off of 0.4 with the total profit being $1,646,320

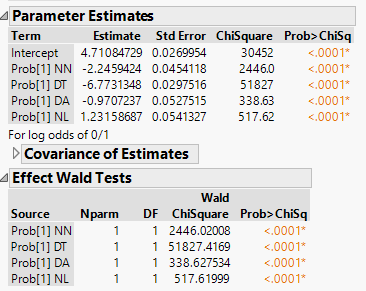
### **Ensemble Model 4**

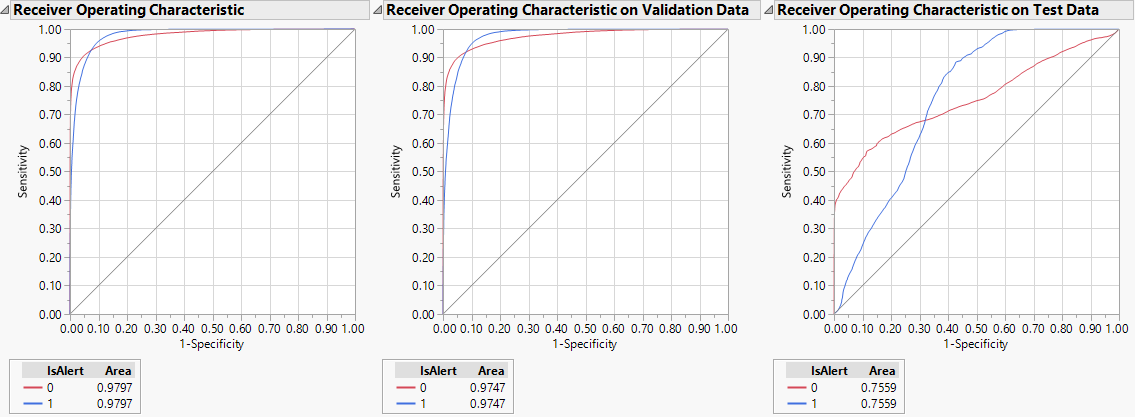
We have seen two ensemble models. Next, we use the probability values from the previous 4 models as predictor variables and use it in Nominal Logistic Regression.



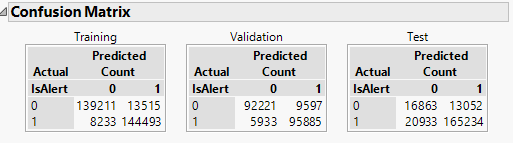
## **Observation**







The confusion matrix is shown below:



From above we can calculate the following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Training** | **Validation** | **Test** |
| **True Positives** | 94.61% | 94.17% | 88.76% |
| **True Negatives** | 91.15% | 90.57% | 56.37% |
| **False Positives** | 8.85% | 9.43% | 43.63% |
| **False Negatives** | 5.39% | 5.83% | 11.24% |
| **Model Accuracy** | 92.88% | 92.37% | 84.27% |

## **Inference**

Model accuracy is almost consistent through training, validation and test. However, the rate of false negatives is on the higher side. So, the best way to decide the goodness of this model would be through cost analysis:

Cost Analysis on Test Data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Cut-off** | **Alert as Not Alert (1 as 0)** | **Not Alert as Alert (0 as 1)** | **Alert as Alert (1 as 1)** | **Not Alert as Not Alert (0 as 0)** | **Total Profit of the Model** |
| 0.1 | 12723 | 15057 | 173444 | 14858 | **$ 1,597,260.00** |
| 0.2 | 16097 | 14403 | 170070 | 15512 | **$ 1,595,180.00** |
| 0.3 | 17819 | 13778 | 168348 | 16137 | **$ 1,623,240.00** |
| 0.4 | 19073 | 13472 | 167094 | 16443 | **$ 1,628,760.00** |
| 0.5 | 20933 | 13052 | 165234 | 16863 | **$ 1,633,560.00** |
| 0.6 | 23384 | 12648 | 162783 | 17267 | **$ 1,624,940.00** |
| 0.7 | 26952 | 12294 | 159215 | 17621 | **$ 1,588,980.00** |
| 0.8 | 32528 | 11443 | 153639 | 18472 | **$ 1,562,560.00** |
| 0.9 | 49702 | 10091 | 136465 | 19824 | **$ 1,354,280.00** |

Conclusion: The model is at its optimum at a probability cut-off value of 0.5 with the total profit being $1,633,560.

### **Model Evaluation**

We highlight the best features and go for the profit matrix. Assuming, that the cost of the model is given to us by the people who built them.

Cost of predicting Alert as Not Alert (1 as 0) = -$10

Cost of predicting Not Alert as Alert (0 as 1) = -$50

Profit of predicting Alert as Alert (1 as 1) = +$10

Profit of predicting Not Alert as Not Alert (0 as 0) = +$50

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Name** | **Cost of the Model** | **Total Profit from the Model** | **Net Revenue from the Model (Profit – Cost)** |
| Decision Tree @ 0.5 cut-off | $5,000 | $16,43,300 | $16,38,300 |
| Neural Network @ 0.2 cut-off | $7,500 | $16,95,140 | $16,87,640 |
| Discriminant Analysis @ 0.6 cut-off | $6,000 | $15,62,500 | $15,56,500 |
| Nominal Logistic Regression @ 0.2 cut-off | $5,000 | $17,26,640 | $17,21,640 |
| Ensemble Model 1 @ 0.7 cut-off | $8,000 | $16,57,200 | $16,49,200 |
| Ensemble Model 2 @ 0.5 cut-off | $10,000 | $16,43,460 | $16,33,460 |
| Ensemble Model 3 @ 0.4 cut-off | $9,000 | $16,46,320 | $16,37,320 |
| Ensemble Model 4 @ 0.5 cut-off | $8,000 | $16,33,560 | $16,25,560 |
| Baseline Model (All values predicted as 0) | $0 | $10,804,100 | $10,804,100 |
| Baseline Model (All values predicted as 1) | $0 | $2,160,820 | $2,160,820 |

## **Recommendation – Classification Model**

Looking at just the cost matrix, it is safe to say that the Nominal Logistic Regression model at a cut off of 0.2 is the best model among the lot, because of its highest profit value and net revenue. If the company had a base model, we would compare this model against it, and if it beats the base model, we would recommend this model to the company. However, in a case such as this, you might want to look at minimizing the cost of false positives. The mistake of a false positive can be pretty high and the company may want to lessen the rate of false positives rather than the overall profit of the model. In such a case, we would recommend the model which has the least false positive rate. In this case, the original Nominal Logistic Regression Model itself has the lowest false positive rate as well, with a false positive rate of 40.17%. So overall, we would recommend the Nominal Logistic Regression model for this classification problem.