

|  |
| --- |
| **ISSS602 DATA ANALYTICS LAB** |
| **Assignment 3:**  **Predicting Automative Insurance Fraud** |
|  |

**Prepared By**

***Raveena Chakrapani***

***(raveenac.2021@smu.mitb.edu.sg)***

**20th March 2022**

Contents

[1. OVERVIEW 4](#_Toc98697837)

[2. OBJECTIVE 4](#_Toc98697838)

[3. DATA 4](#_Toc98697839)

[3.1 Raw Data Description 4](#_Toc98697840)

[3.2 Loading Data into JMP Pro 4](#_Toc98697841)

[3.3 Data Quality check 4](#_Toc98697842)

[3.3.1 Completeness and Accuracy 4](#_Toc98697843)

[3.3.2 Missing Value 5](#_Toc98697844)

[3.3.3 Excluding the unnecessary variables 5](#_Toc98697845)

[3.3.4 Recoding the classes 6](#_Toc98697846)

[3.4 Exploring the Data 7](#_Toc98697847)

[3.4.1 Univariate Analysis 7](#_Toc98697848)

[3.4.2 Complete Separation & Quasi-Complete Separation 7](#_Toc98697849)

[3.4.3 Bivariate Analysis 8](#_Toc98697850)

[3.4.4 Multicollinearity 8](#_Toc98697851)

[3.4.5 Predictor Variables 8](#_Toc98697852)

[3.4.6 Data Sampling 9](#_Toc98697853)

[4. Calibrating Predictive Models usiing Logistic Regression 12](#_Toc98697854)

[4.1 Calibrating Base Model 12](#_Toc98697855)

[4.1.1 Model Performance 13](#_Toc98697856)

[4.1.2 Saving Prediction Formula 14](#_Toc98697857)

[4.2 Calibrating Stepwise Regression 14](#_Toc98697858)

[4.2.1 Stepwise Min AIC 15](#_Toc98697859)

[4.2.2 Stepwise Max Validation Square 16](#_Toc98697860)

[5. Calibrating Predictive Models using Recursive Partioning 16](#_Toc98697861)

[5.1 Decision Tree 16](#_Toc98697862)

[5.1.1 Model Fitting 16](#_Toc98697863)

[5.1.2 Model Performance 16](#_Toc98697864)

[5.1.3 Saving Prediction Formula 18](#_Toc98697865)

[5.2 Random Forest 18](#_Toc98697866)

[5.2.1 Model Fitting 18](#_Toc98697867)

[5.2.2 Model Performance 18](#_Toc98697868)

[5.2.3 Saving Prediction Formula 19](#_Toc98697869)

[5.3 Random Tree 19](#_Toc98697870)

[5.3.1 Model Fitting 19](#_Toc98697871)

[5.3.2 Model Performance 19](#_Toc98697872)

[5.3.3 Saving Prediction Formula 20](#_Toc98697873)

[6. MODEL COMPARISON 21](#_Toc98697874)

[7. INTERPRETATION & RECOMMENDATION 23](#_Toc98697875)

[8. REFERENCES 23](#_Toc98697876)

# OVERVIEW

Genrally, Insurance fraud is a deliberate deception perpetrated against or by an insurance company or agent for the purpose of financial gain.Especially,automotive insurance fraud entails someone attempting to deceive an insurance company about a claim involving their personal or commercial motor vehicle. It can involve giving out misleading information or providing false documentation to support the claim. Automotive insurance fraud can be brought into many different forms such as staging an accident, the policyholder not involved in the claimed accident, duplicate claims for the same injury, a fake injury and many other misrepresentations (Derrig, R.A., 2002). In order to detect these fraudulent claims, insurance companies and fraud investigators need to know what characteristics lead to a fraudulent claim.

# OBJECTIVE

The objective of this analysis is to build a model to predict fraud automobile insurance claims and to identify the features in the database which contributes to predict the fraud cases. This prediction can be done by building multiple predictive models by using logistic regression (including stepwise) and decision tree (including random forest and boosting trees). Further, a single performance criterion will be used for model assessment.

# DATA

## Raw Data Description

For the purpose of this assignment, automobile claim data called *claimdata.csv* will be used. The data file is in csv format. It consists of 34 fields and 15,420 claim records. *FraudFound* is the response variable.

## Loading Data into JMP Pro

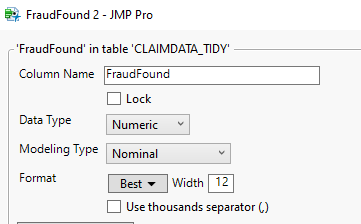
Import the *claimdata.csv* file into JMP Pro 16 and save it as *CLAIMDATA.jmp*

## Data Quality check

### Completeness and Accuracy

Once the data is loaded and integrated, let us check for accuracy and completeness. By accuracy, let us check whether datatypes in the imported dataset matches the original dataset. By completeness, we check whether no. of rows and columns, which are imported, matches the original dataset. This data contains 15,420 records and 34 columns. Some of the variables datatype is incorrectly assumed by JMP Pro. Let’s modify it.

For. Eg**. Right click on** *FraudFound* **🡪 Column Info🡪 Nominal in Modelling Type 🡪 OK**



Similarly change for all the below mentioned variables.

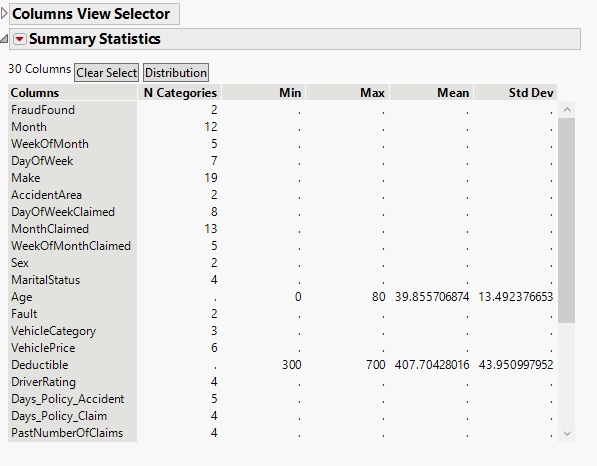
|  |  |  |
| --- | --- | --- |
|  | Modeling Data Type | |
| **Variables** | **Existing** | **Modified** |
| WeekOfMonth | Continuous | Nominal |
| WeekOfMonthClaimed | Continuous | Nominal |
| FraudFound | Continuous | Nominal |
| DriverRating | Continuous | Ordinal |
| DaysPolicyAccident | Nominal | Ordinal |
| DaysPolicyClaim | Nominal | Ordinal |
| PastNumberOfClaims | Nominal | Ordinal |
| AgeOfPolicyHolder | Nominal | Ordinal |
| NumberOfSuppliments | Nominal | Ordinal |
| AddressChangeClaim | Nominal | Ordinal |
| Year | Continuous | Nominal |

### Missing Value

Let us check for missing values in the dataset

* From menu bar, select **Cols** -> **Columns Viewer**.
* Columns Viewer dialog window appears.
* At the **Select Columns** pane, select all the fields on the list.
* Click on **Show Summary** button.

Minimise the dialog window by clicking on the black triangle in front of **Columns View Selector**.



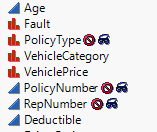
The report revelas that there are no missing values in the dataset.

### Excluding the unnecessary variables

While doing analysis, to avoid selecting the unwanted columns accidentally, let us hide and exclude the unnecessary columns. Here let us hide these columns shown in below fig. Select these columns on Columns Pane 🡪Right click 🡪 Hide/ Unhide and select Exclude/Unexclude options.

Reasons for Exclusion:

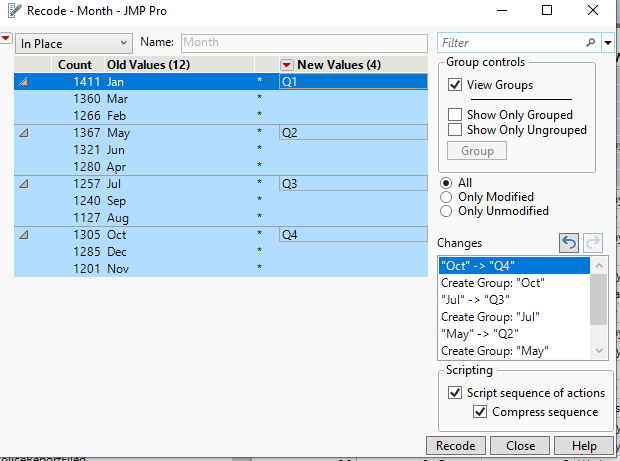
* *PolicyNumber*. & *RepNumber* 🡪 Unique number for each policy holder and each Insurance Representative.
* *PolicyType* 🡪 Combined column of *VehicleCategory* and *BasePolicy*



### Recoding the classes

There are multiple variables for which the categories are mre than 5. This may impact our model. Let’stry to reduce the number of categories in each variable as much as possible.

**Right Click on *Month* 🡪 Recode 🡪 Group into required classes 🡪 Select Recode Button.**



Similarly group all the following variables as shown in the table below.

|  |  |
| --- | --- |
| **Variable** | **Recoded Classes** |
| Month | Quarters Q1,Q2,Q3 &Q4 |
| MonthClaimed | Quarters Q1,Q2,Q3 &Q4 |
| DaysPolicyAccident | Less than 30 |
|  | More than 30 |
| MaritalStatus | Married |
|  | Unmarried |
| Vehicle Price | < 29000 |
|  | 30000 to 69000 |
|  | > 69000 |
| DaysPolicyClaim | Less than 30 |
|  | More than 30 |
| No.ofSuppliments | 1 to 2 |
|  | 3 to 5 |
|  | more than 5 |
| AddressChangeClaim | Changed |
|  | No change |
| NumberOfCars | 1 Vehicle |
|  | 2 Vehicles |
|  | > 3 vehicles |

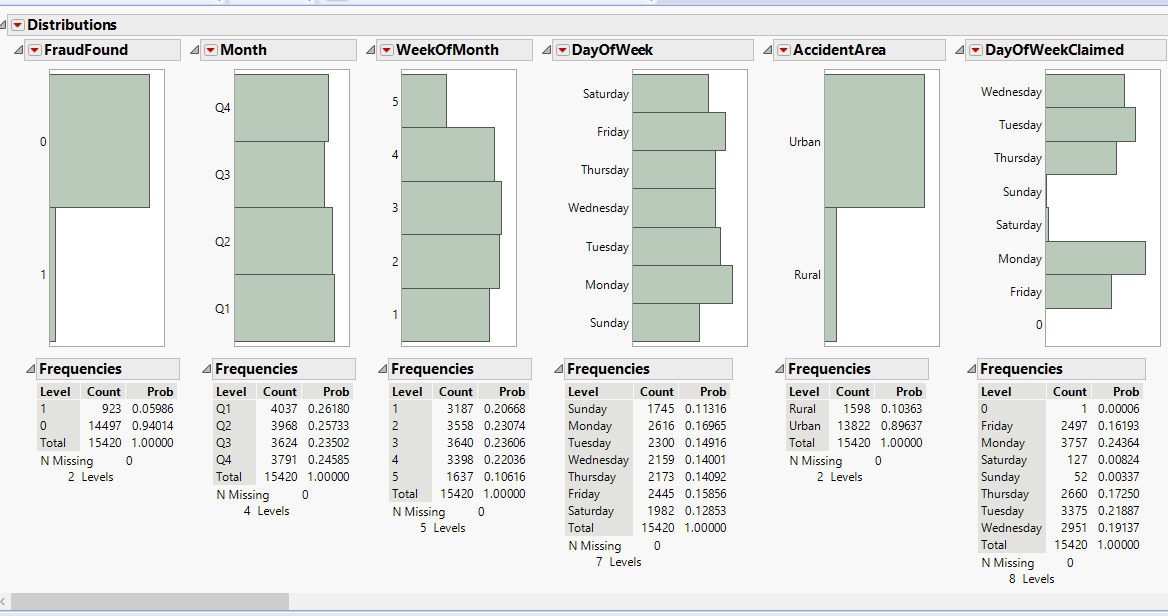
And for all the nominal variables, value orders are checked and to be changed if required.After performing all the above quality checks, save the dataset as *CLAIMDATA\_TIDY*.jmp

## Exploring the Data

Let us explore the *CLAIMDATA\_TIDY*.jmp dataset by using appropriate exploratory univariate and bivariate data analysis techniques. The purpose of this analysis is to discover the distributions of the variables and their inter-relationships.

### Univariate Analysis

Select Analyze 🡪 Distribution. Select all variables 🡪 click on Y, columns 🡪 OK

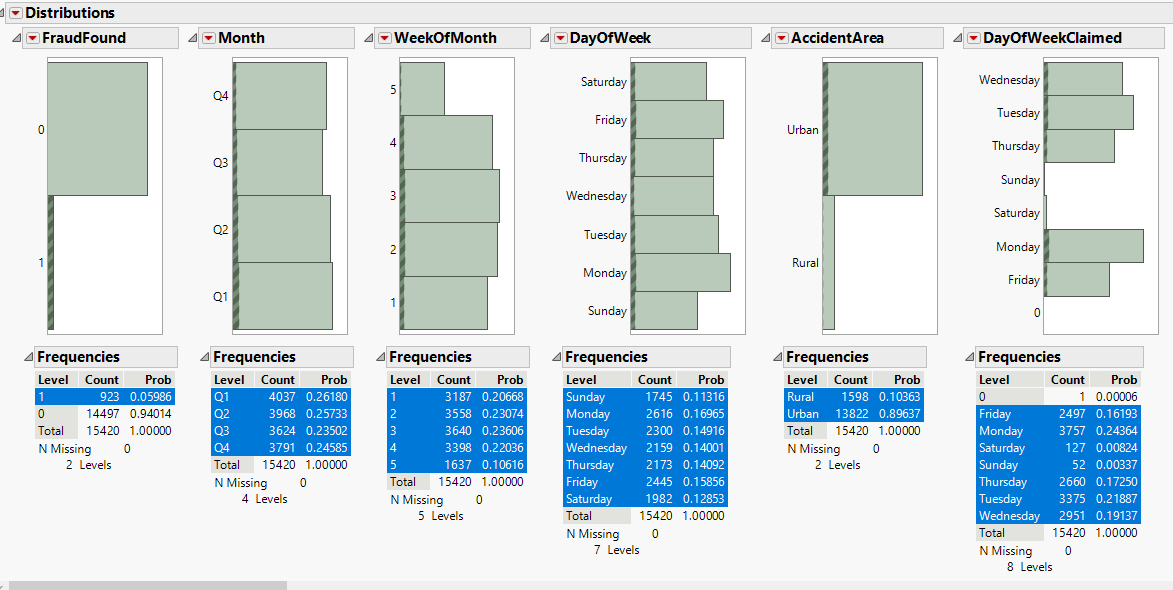


The above report hepls us to look at the distribution of each variables and frequency of each value.

### Complete Separation & Quasi-Complete Separation

A complete separation happens when the outcome variable separates a predictor variable or a combination of predictor variables completely.Quasi-complete separation happens when the outcome variable separates a predictor variable or a combination of predictor variables to certain degree. Lets check for these variables in our dataset.

Select FraudFound as 1 in the Distribution chart and its portions are highlighted.



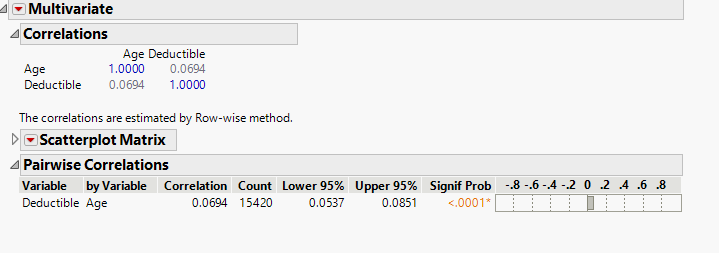
Scroll right and observe all the variables. There is no sign of Complete separtion or quasi complete separation.

### Bivariate Analysis

Select Analyze 🡪 Multivariate Methods 🡪 Multivariate. Select continuous variables *Age* & *Deductible* 🡪 click on Y, columns 🡪 OK

### Multicollinearity

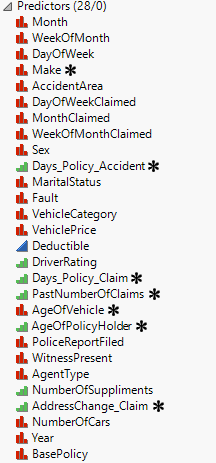
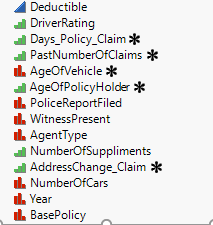
Let us examine the inter-relationship between the predictors. It is important not to include predictors that are highly correlated in building the predictive model. One way to examine the correlation of multiple variables at one go is by using correlation matrix.



The analysis results reveal that the selected variables are not strongly correlated. A highly correlated pair will have a correlation value greater than +/-0.80. In view of this, we will use these variables in building our model.

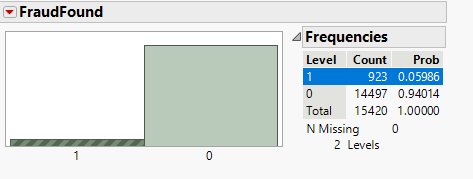
### Predictor Variables

After preparing our data and performing quality checks, following variables can be concluded as predictors. It can be grouped into a column called Predictors by right clicking on the highlighted variables 🡪Group Columns

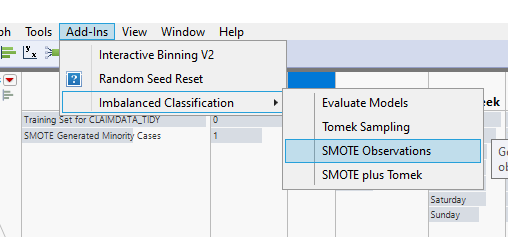
### Data Sampling

One of the most common characteristic noticed in the distribution analysis is that this dataset is highly imbalanced. Imbalanced data refers to those types of datasets where the target class has an uneven distribution of observations, i.e one class label has a very high number of observations and the other has a very low number of observations.



The above graph shows that Fraud Found cases are extremely low when compared to Fraud not found. This is an classic example of imbalanced dataset. It explicity shows that this dataset is biased towards a Fraud Not found class. If the dataset is biased towards one class, an algorithm trained on the same data will be biased towards the same class. Therefore, it is crucial to balance the dataset.

In JMP Pro, let us insatll and use the **Imbalanced Classification** add in which has different types as shown in the figure.



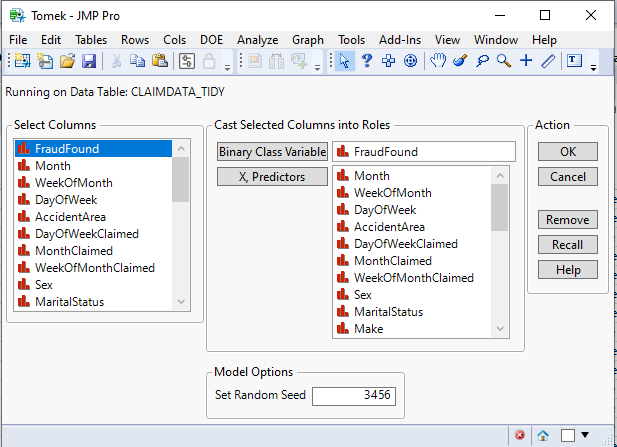
There are 2 types of Sampling. Oversampling and Undersampling.

*Oversampling* the minority classes to increase the number of minority observations until we've reached a balanced dataset. We can use our existing dataset to synthetically generate new data points for the minority classes. Synthetic Minority Over-sampling Technique (SMOTE) is a technique that generates new observations by interpolating between observations in the original dataset.

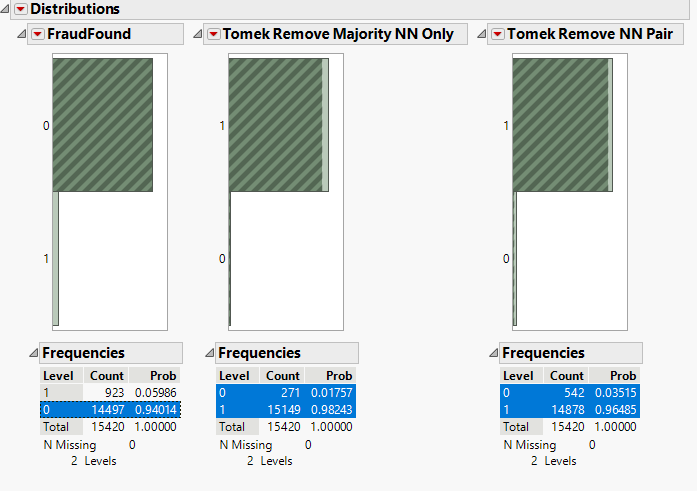
*Undersampling* the majority class - essentially throwing away data to make it easier to learn characteristics about the minority classes.Tomek Sampling helps to remove data points from the majority classes. Tomek Links refers to a method for identifying pairs of nearest neighbors in a dataset that have different classes. Removing data points such in the majority class has the effect of making the decision boundary in the training dataset less noisy or ambiguous.

SMOTE plus Tomek is the combination of both the above mentioned methods. It not only increases the number of minority observations but also decreases the number of majority observations to create a balanced dataset.

* First, let us try using the Tomek Sampling.
* Select Add-Ins 🡪 Imbalanced Classification 🡪 Tomek Sampling.
* FraudFound as Binary Class Variable , All other predictors as X, Predictors.
* Set Random Seed as 3456

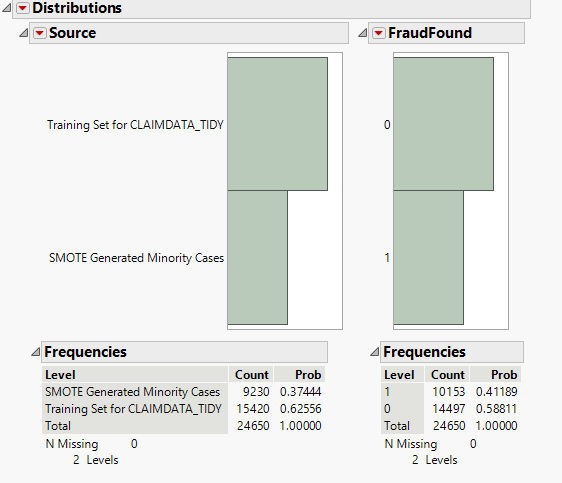


Let us look at the distribution of Tomek Columns (Remove Majority NN only and NN Pair) and compare with the original FraudFound column



The report shows that even after performing sampling it is unable to create a balanced set. There is no significant difference in the majority class. Hence, this technique can be rejected.

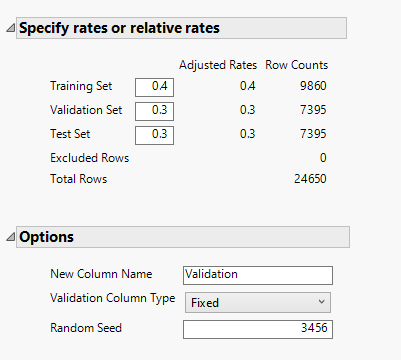
Similary perform SMOTE technique and analyse the distribution. Save the newly created dataset as ***CLAIMDATA\_TIDY with SMOTE.jmp***. This data table will be used for building all our upcoming predictive models.



This clearly shows that significant no. of minority cases are increased and dataset is balanced now. Lets go ahead to perform data sampling and build predictive model.

At this point, before constructing the predictive model, let us divide the dataset into three distinct sets as training, validationa and test data. This is to avoid the problem of overfitting and underfitting in our data mining efforts. Let us sample the data set into 40%, 30% and 30% for training, validation and test data sets respectively.

* From menu bar, **select Analyze -> Predictive Modeling -> Make Validation Column.**
* The Make Validation Column dialog window appears.
* At the Select Columns panel, click on *FraudFound*.
* Next, click on Stratification Columns button.
* Click Go



# Calibrating Predictive Models usiing Logistic Regression

## Calibrating Base Model

Now, we are going to fit a base model with 28 predictors.

• From menu bar, click **Analyze** -> **Fit Model**.

The **Fit Model** dialog window appears.

• From **Select Columns** pane, click on *FraudFound*.

• At **Pick Role Variables** pane, click on **Y** button.

**Personality** change to **Nominal Logistic**, Set Target Level as 1

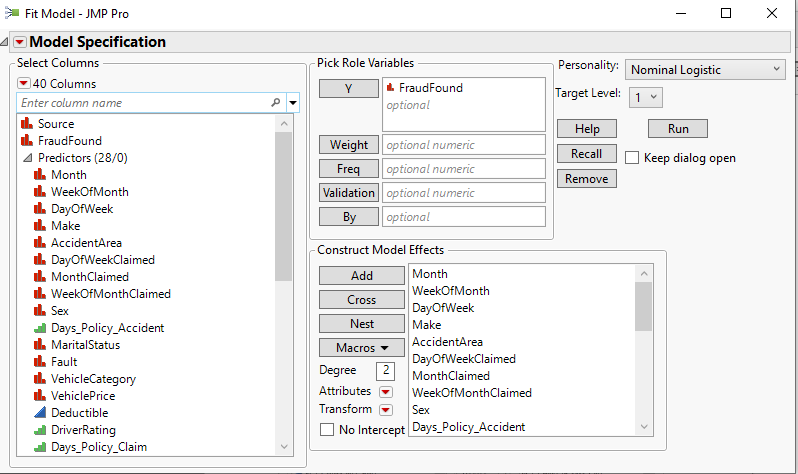
Next, we are going to include the predictors into the model.

• From **Select Columns** pane, click on all Predictor Variables (grouped already)

Next, we will include the data sampling field.

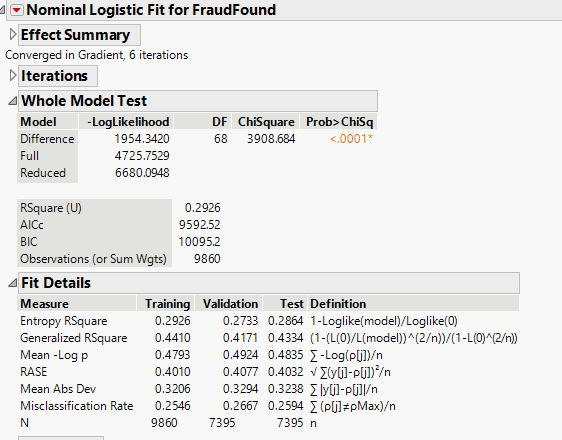
• From **Select Columns** pane, click on *DATA SAMPLING*.

• At **Pick Role Variables** pane, click on **Validation** button and click on **Run** button

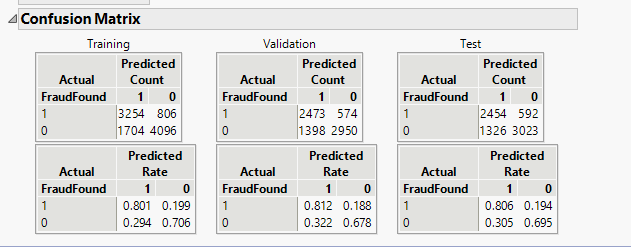


### 4.1.1 Model Performance

The report window appears.



It is reported at **Fit Details** report. With reference to the screenshot above, the misclassification rate for the validation data is 0.2667 and the difference between the misclassification rate of training and validation data is less than 5%. Hence, the model seems to be good.However, it is not sufficient to just refer to the misclassification rate. From the **Logistic Regression Report** window, click on the red triangle next to **Nominal Logistic Fit for FraudFound**. Select **Confusion Matrix** from the drop-down list. The **Confusion Matrix** report appears.



The confusion matrix provides details of what type of misclassification is more frequent. From the confusion matrix above, it can be seen that the model does better in predicting fraud found cases correctly and is less accurate in classifying fraud not found cases. The true positive rate of training,validation and test are 0.801, 0.812 and 0.806 respectively. On the other, the true negative rate for training, validation and test are 0.706, 0.678 and 0.695 respectively.

This model is complex, and some variable levels are not significant. Hence, let us reduce the numbers of predictors but without compromise the performance of the model significantly.

### 4.1.2 Saving Prediction Formula

The last step in model building using JMP Pro is to save the prediction formula for future use such as model comparison or model deployment.

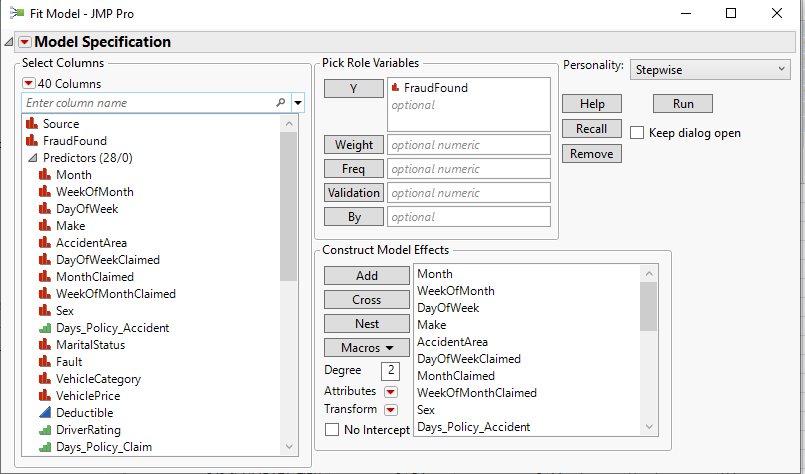
From Fit Normal Logistic - Full Model dialog window, click on the at the top of the report.

Select **Save Column** -> **Save Prediction Formula** from the context menu.

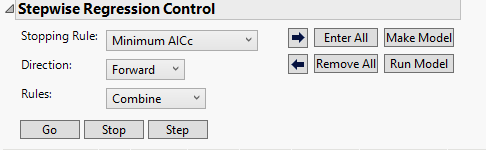
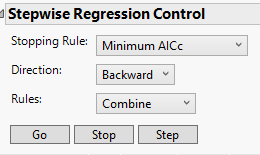
4 new columns have added to the data table, they are: Lin[1], Prob(Target FraudFound==FraudFound), Prob(FraudFound ==No FraudFound) and Most Likely FraudFound.

## Calibrating Stepwise Regression

This model calibration is similar to Logistic Regression in selecting variables with the only difference of selecting **Stepwise** in Personality.

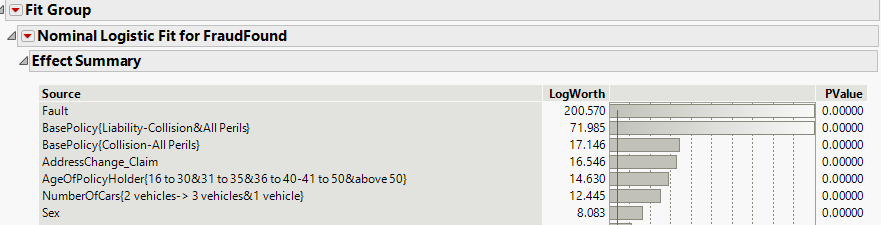


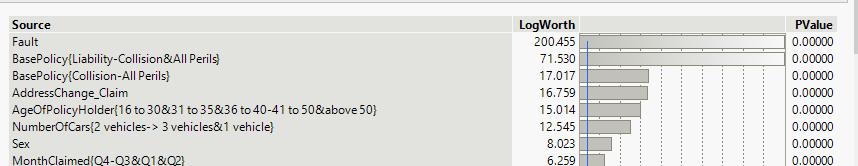
### Stepwise Min AIC

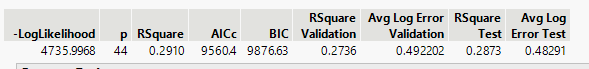
In Forward, the system starts from 0 variables. So, select Go

In Backward, the system starts will all predictors and reduce it gradually. So, select Enter All and then selet Go.

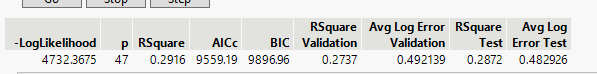




In both Forward and Backward methods the important variables contributing to the model are Fault, Base Policy, address change claim.

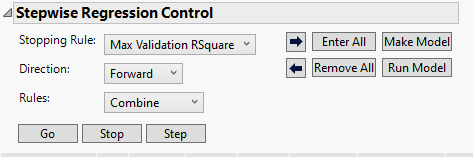


In Forward Method, Min AICc 🡪 9560. This is the minumum of all.



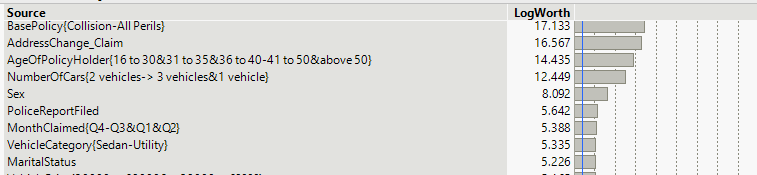
In Backward Method Min AICc 🡪 9559. There is no much difference.

### 4.2.2 Stepwise Max Validation Square

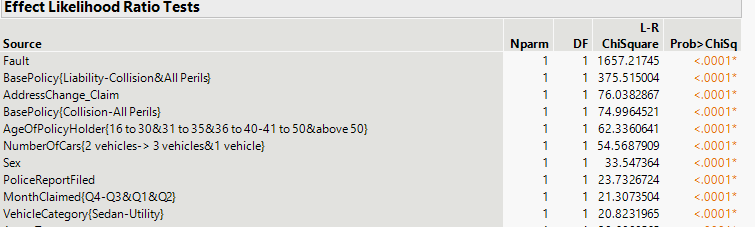


Select GO and the report appears.

The top contributing variables are as follows:



The below chart shows that variables with larger ChiSquare value such as Fault, Base Policy, AddressChange\_Claim are all significant.



# Calibrating Predictive Models using Recursive Partioning

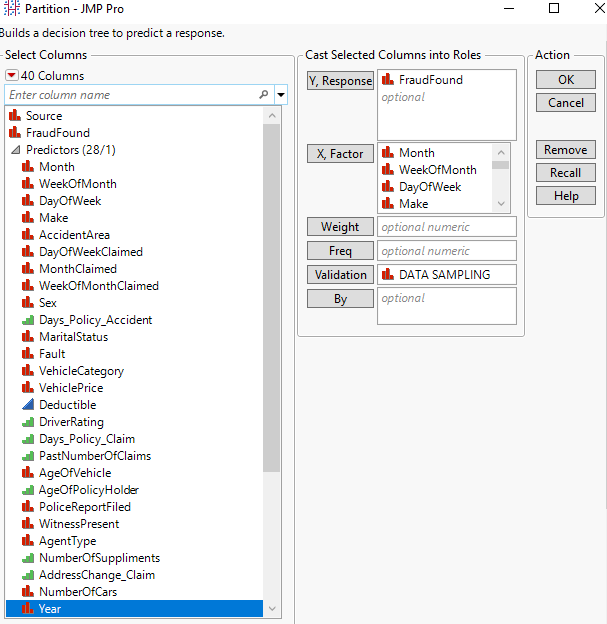
Recursive partitioning is useful for both exploring relationships and for predictive modeling: It is very flexible and allows user to find not only splits that are optimal in a global sense, but also node-specific splits that satisfy various criteria. This is its advantage as it allows flexibility at the discretion of the user.

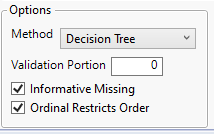
## Decision Tree

### 5.1.1 Model Fitting

Select Analyze 🡪 Predictive Modelling 🡪 Partition. The Partition window appears.

Similar to Regression, choose desired variables as shown in the figure and Click OK.

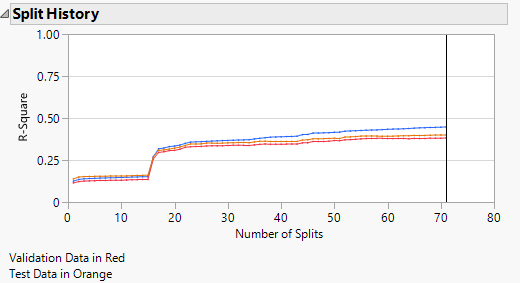
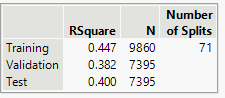




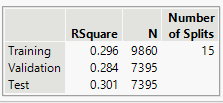
### 5.1.2 Model Performance

Now let us ask JMP to automatically check all the independent variables and all the possible splits for them, and choose the variable and split that maximize the Log Worth statistic. This can be done by selecting Go button to the left of Split History.

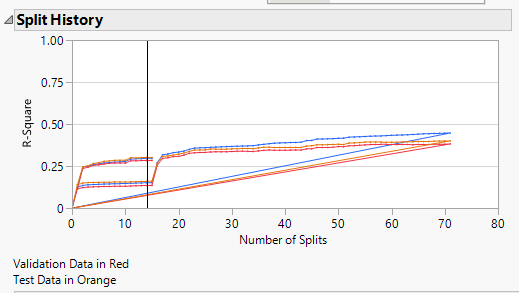
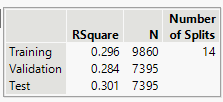
The tree has 71 splits in total. This tree is overly complex. It’s the initial tree.

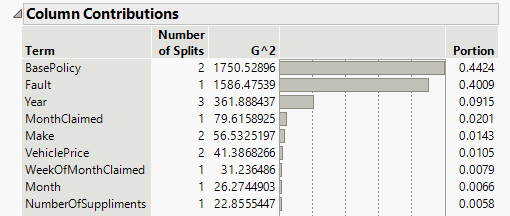
Let us not consider the initial tree since it is very complex. Lets select minimum size split and insert 0.05 to limit the leaf size to 5% of data. Then we can press Go button to obtain the automatically generated tree. The resulting model has 15 splits.



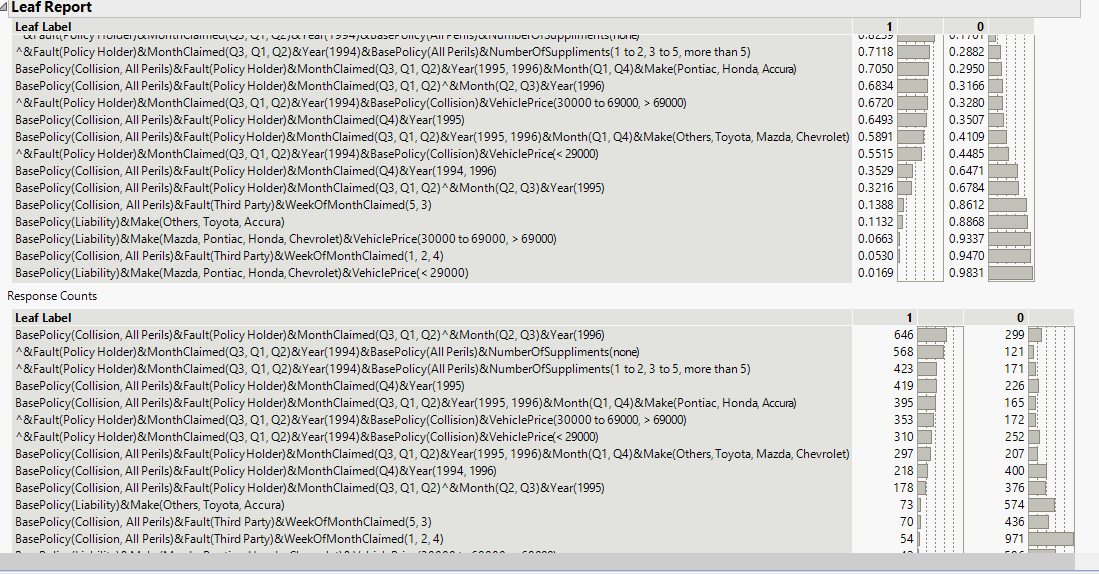
Let’s further prune the worst leaf of the tree by using Prune Worst option.

The column contributions reveal the top variables contributing to the are Base Policy, Fault, Deductible



The leaf report after pruning is sorted and shown below.



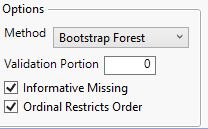
### 5.1.3 Saving Prediction Formula

As explained in the Regression section, save the prediction formula and rename it by the model name.

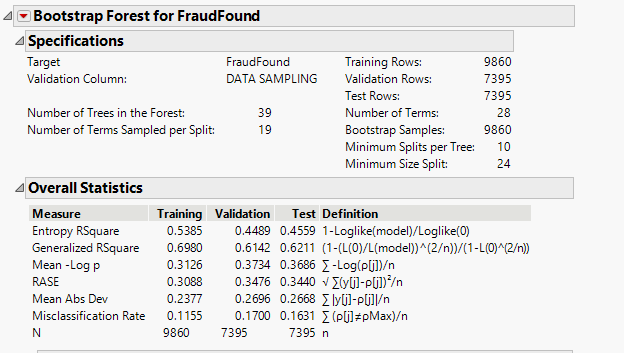
## Random Forest

### 5.2.1 Model Fitting

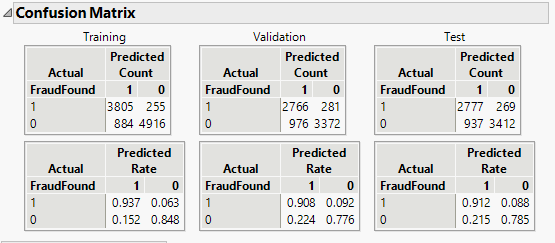
Since descision tree model is lazy and does not do repetitive splitting based on residuals, let us perform Random Forest which is an advanced recursive partitioning technique. **Select Analyze 🡪 Predictive Modelling 🡪 Bootstrap Forest**. Select the desired variables similar to Decision Tree model with the only difference of selecting **Bootstrap Forest** in the Method dropdown box and click **OK**.



### 5.2.2 Model Performance

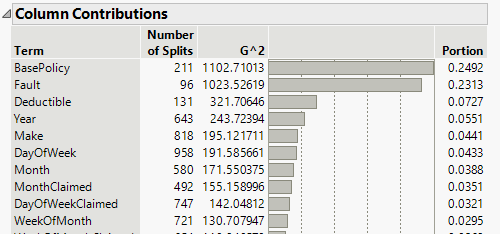
Let us analyse the report window.

The misclassification rate difference is close to 5% between Training and test as well as Training and Validation.



The confusion Matrix report provides us information that the model performs well for True Positives for Fraud found and its Predicted rates are 93%, 92% and 91% for Training, Validation and Test data respectively.

The top contributing variables are as follows



### 5.2.3 Saving Prediction Formula

As explained in the Regression section, save the prediction formula and rename it by the model name.

## Random Tree

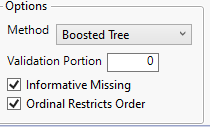
### 5.3.1 Model Fitting

**Boosted Tree** or **Gradient boosting**is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an **ensemble** of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest.

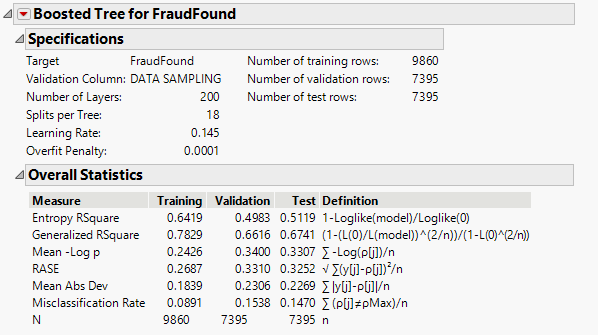
### 5.3.2 Model Performance

A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

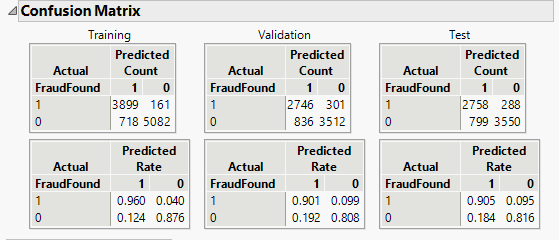
Let us perform Bossted Tree which is an advanced recursive partitioning technique. **Select Analyze 🡪 Predictive Modelling 🡪 Boosted Tree**. Select the desired variables similar to Decision Tree model with the only difference of selecting **Bootstrap Tree** in the Method dropdown box and click **OK**.



Let us analyse the report window.

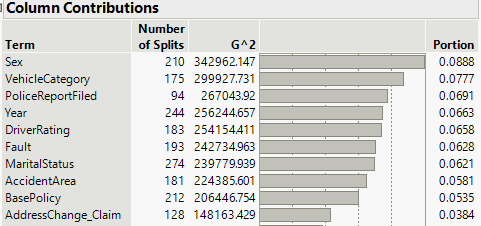


The misclassification rate difference is little above 5% between Training and test (5.7%) as well as Training and Validation (6.4%)



The confusion Matrix report provides us information that the model performs well for True Positives for Fraud found and its Predicted rates are 96%, 91% and 90% whereas Predicted Rates for True negatives are 87%, 81%, 82% for Training, Validation and Test data respectively.

The top contributing variables are as follows



### 5.3.3 Saving Prediction Formula

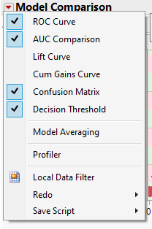
As explained in the Regression section, save the prediction formula and rename it by the model name.

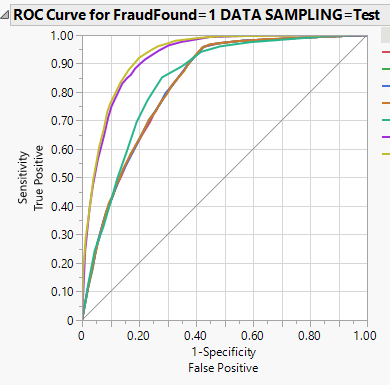
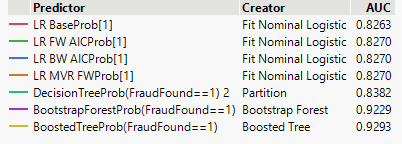
# MODEL COMPARISON

Now let us compare the predicted models based on metrics.

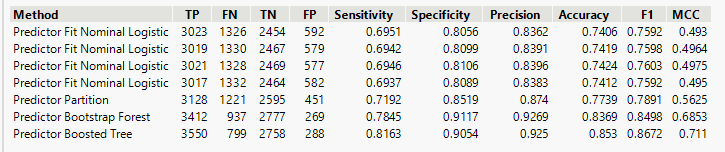
**Select Analyze 🡪 Predictive Modelling 🡪 Model Comparison**

From the red triangle to the left of model comparisons. Select all these options shown below

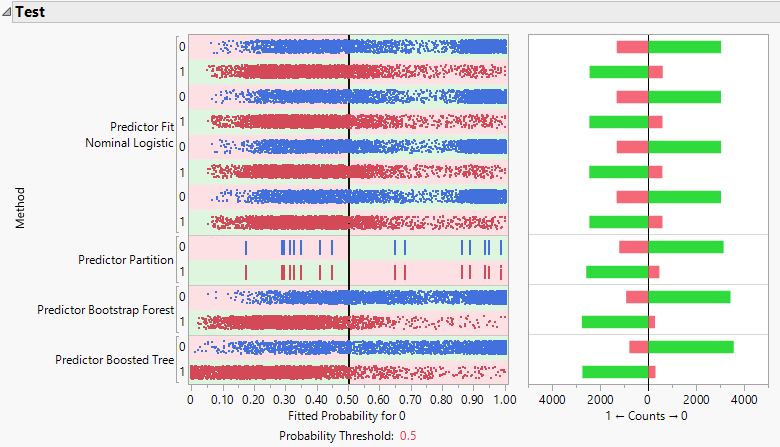


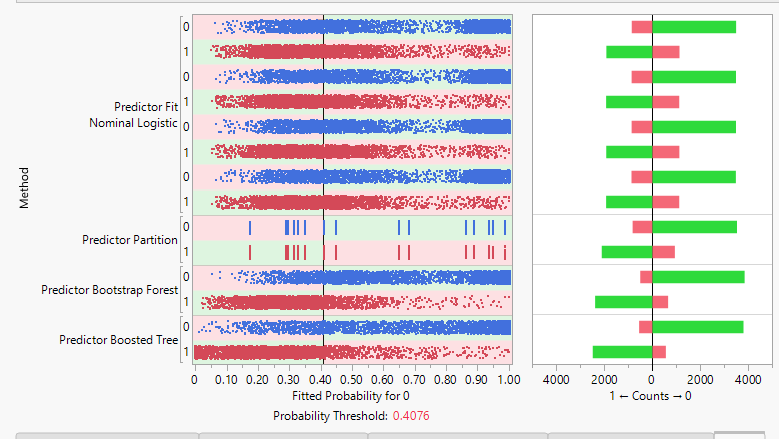
 

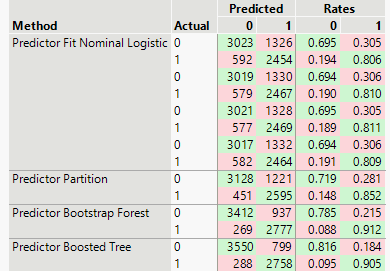
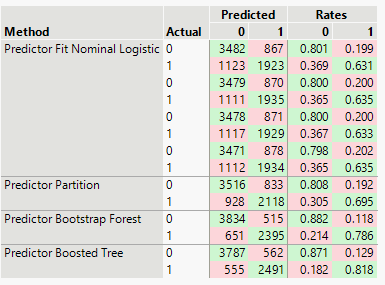
By ROC Curve, it is understood that Boosted Tree is found to be good model because The closer an ROC curve is to the **upper left corner**, the more efficient is the test is.



Also, by considering F1 score, Boosted Tree proves to be a good model because the higher the F1 score is, the more efficient the model is.Now for our objective of detecting Fraud, its better to keep the probability at 0.4 so even the False positives are higher, it may help the insurance company to keep its customers cautious.





With 0.5 Prob Threshold With 0.4 Prob Threshold

It is concluded that on basis of F1 metric, Boosted Tree model at 0.4 threshold is the best.

# INTERPRETATION & RECOMMENDATION

The following Interpretations and recommendations are made:

1. Vehicle Category – 72 % of people owning Sedan are found to be in Fraud found.
2. Driver Rating – People with low driver rating are having higher fraud found probability
3. Fault- Policy Holder is majorly found fraud (81%) than Third Party
4. Marital Status- Married and Unmarried status ratio of fraud found is 7:3
5. Base Policy – Predominantly Collision (42%) and All Perils categories (36%) are found to be deceived.
6. Address Change Claim - People who change address are probably more found to be fraud than people who don’t change address much.

To further, substantiate the analysis:

1. Important variables such as vehicle price could have been more helpful while building Predictive Model.
2. Educational Qualification, Profession could have been again helpful in building the predictive model.

# REFERENCES

* <https://www.jair.org/index.php/jair/article/view/11192>
* <https://repositorio.ucp.pt/bitstream/10400.14/15529/1/MScBA%20-%20Jo%C3%A3o%20Vale%20-%20Using%20Data%20Mining%20to%20Predict%20Automobile%20Insurance%20Fraud.pdf>
* <https://www.jeremyjordan.me/imbalanced-data/>