Team INFINITY

Travelers Case Competition

**Team Members:**

Abishek Ramachandraraju Krishnaraj

Animesh Choudhary

Arjun Periyasamy Sathasivam

Dinesh Shriram

Subhashree Srinivasan

Contents

[1. Tool Used for Model Creation 2](#_Toc497508893)

[2. Data Pre-Processing 3](#_Toc497508894)

[2.1 Outlier Detection 4](#_Toc497508895)

[2.2 Processing Missing Data 4](#_Toc497508896)

[2.3 Column Transformations 8](#_Toc497508897)

[3. Variable Selection 10](#_Toc497508898)

[4. Final Performance Training 13](#_Toc497508899)

# Tool Used for Model Creation



* JMP has an in-memory architecture, we can interact immediately with data without having to submit code and wait for output or graphs.
* The user interface makes it easier to explore and perform analysis. Pre-processing techniques like imputing the missing values, cleaning the data and creating validation column is extremely simple because they are readily available embedded options.
* It is faster to create multiple models with/without multiple techniques, compare them across various parameters like ROC, AUC, misclassification rate, R-square, adj. R square etc. to finalize on a model. Visualization of required parameters and results is easier.

# Data Pre-Processing

In this stage, we’ve detected outliers and removed them from our analysis. Following this, we identified the variables that contain missing data and recoded them. Created a validation set in to create models.

### Outlier Detection

Outlier detection using Mahalanobis Distances

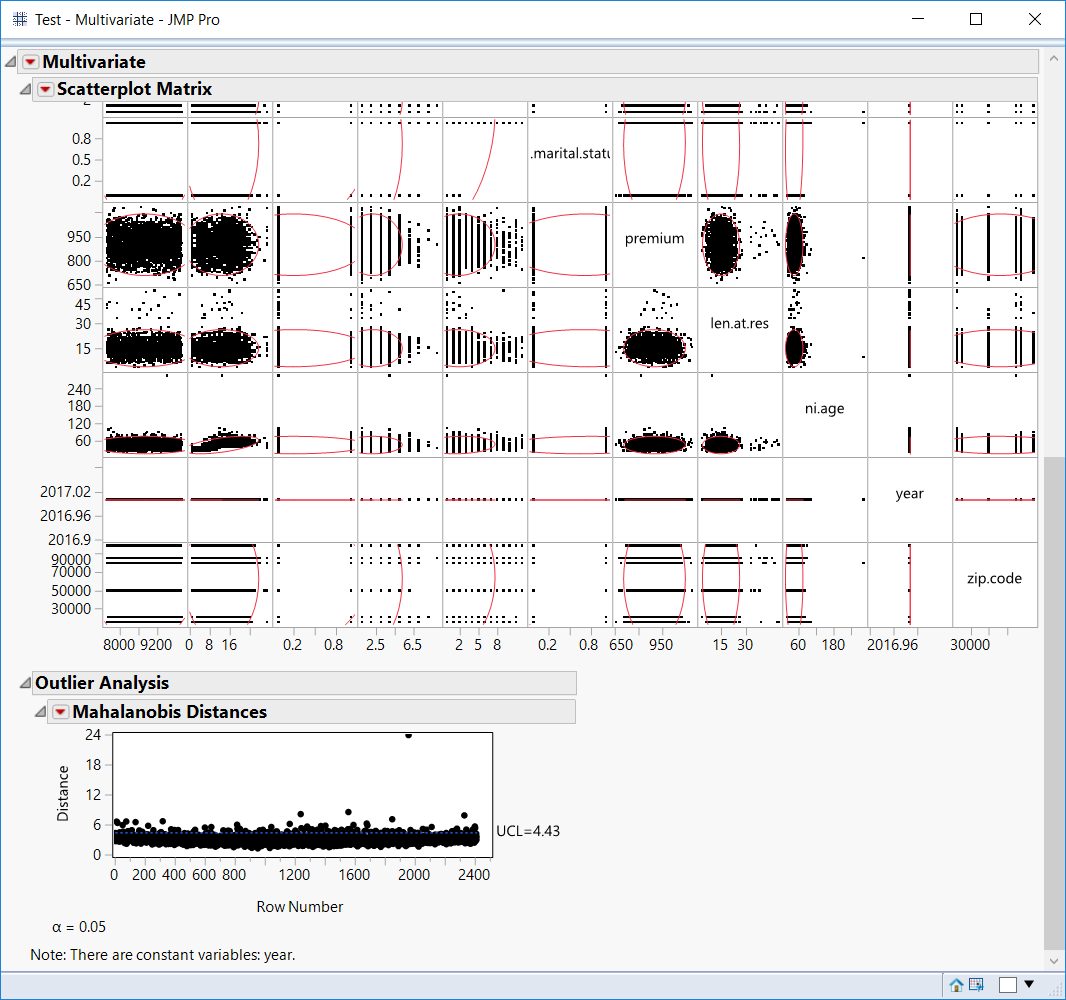


Figure 1

After removing the outliers

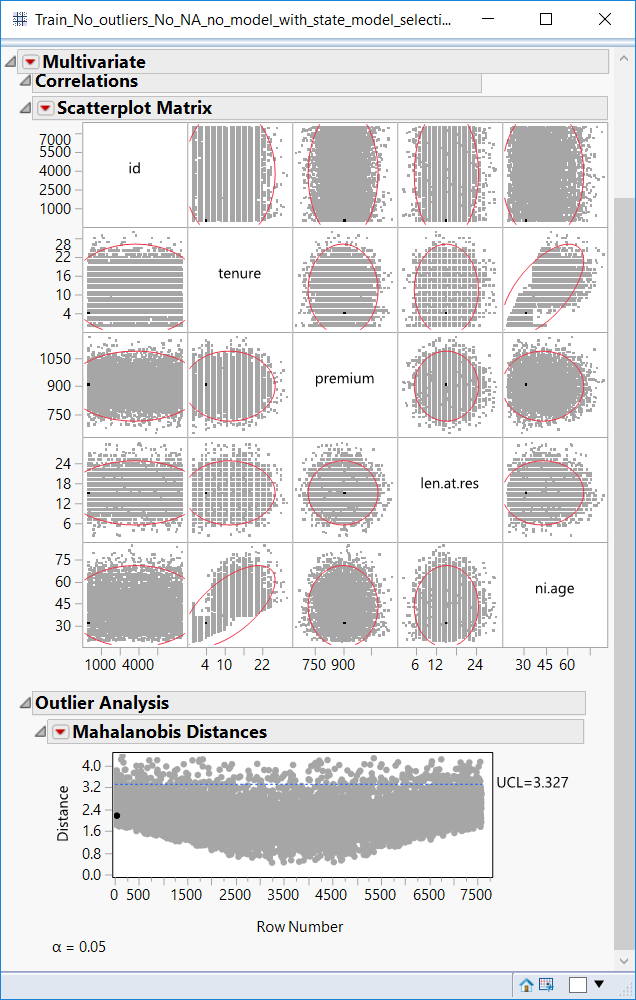


Figure 2

### 

### Processing Missing Data

Finding the missing data

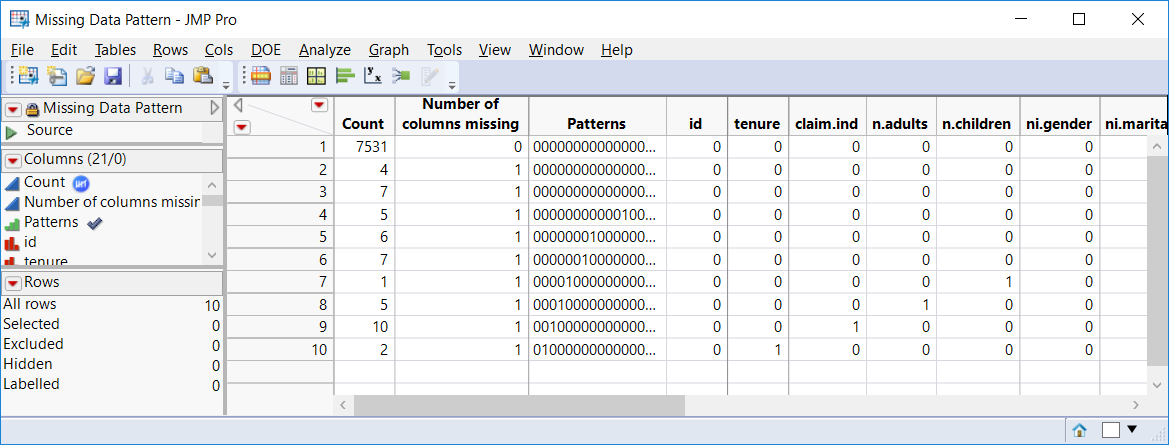


Figure 3

For continuous variables, we’ve used ‘Multivariate Normal Imputation’, as shown in Figure 5, to recode the missing data. Below is an example for variable ‘Tenure’

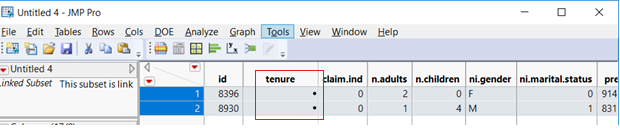


Figure 4

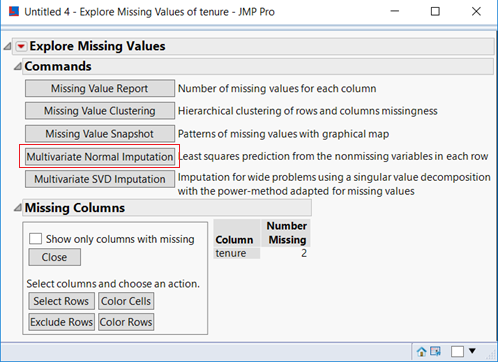


Figure 5

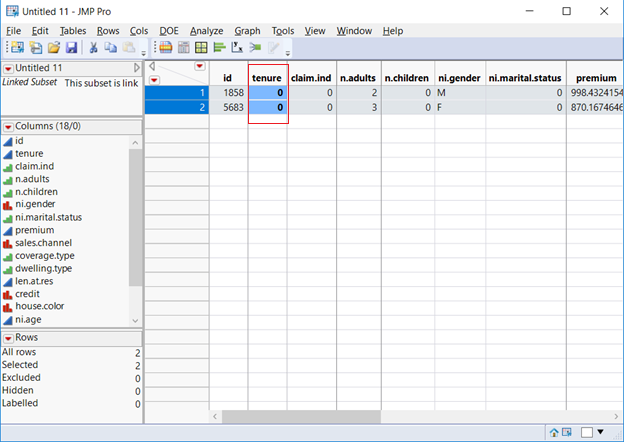


Figure 6

For all the other type of variables, we’ve replaced the missing values with the most occurring value. Below is the example for variable

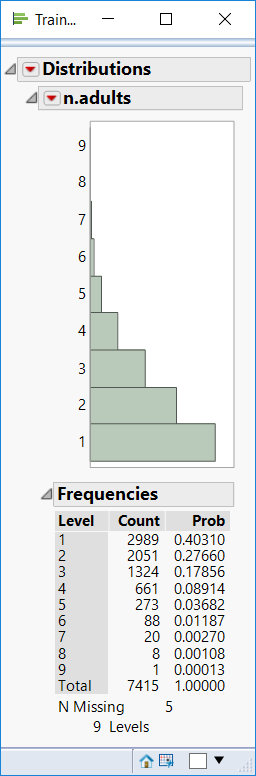
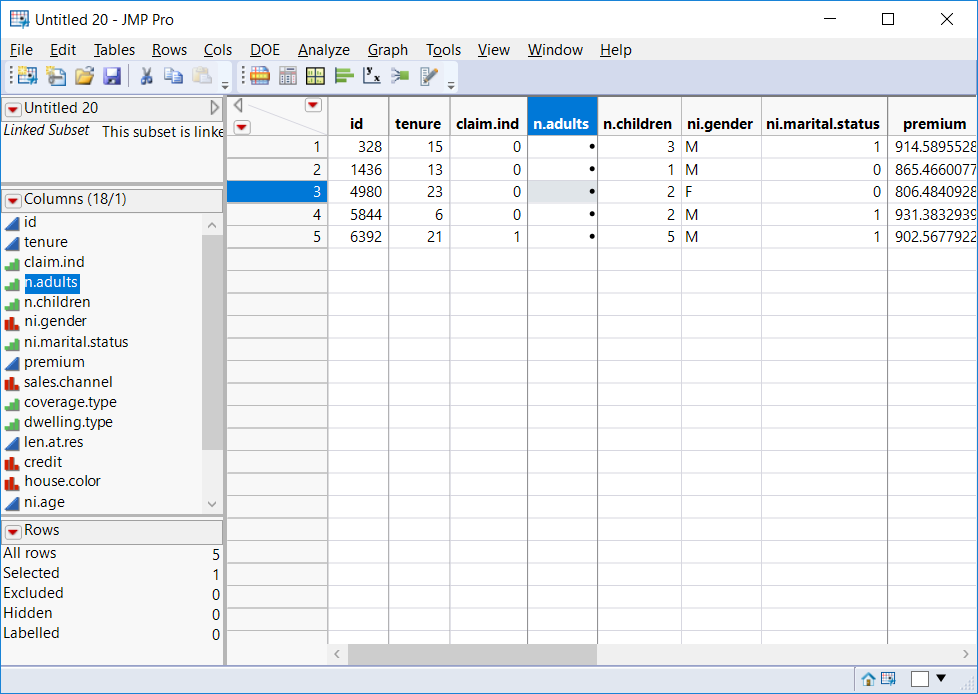
 Figure 8

Figure 7

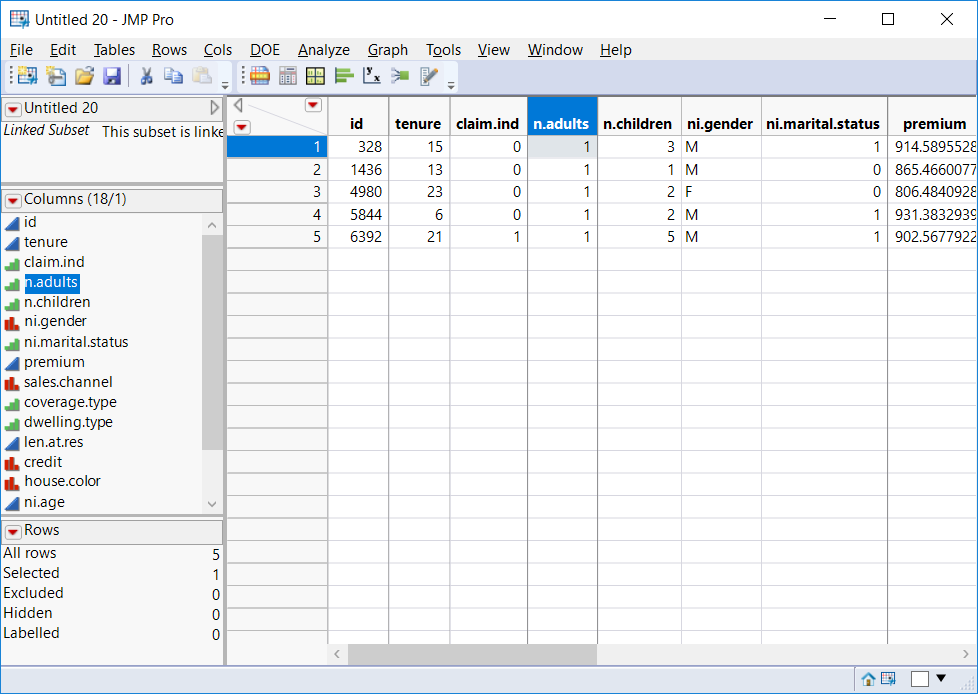


Figure 9

During model exploration phase, we created bootstrap forests where the impurity in validation data set is re-used to prune the trees until an optimal forest is built. Hence, validation set is not exactly "unseen" by the model. For this reason, we partitioned the training data into 3 parts - training (60%), validation (20%) and test (20%)

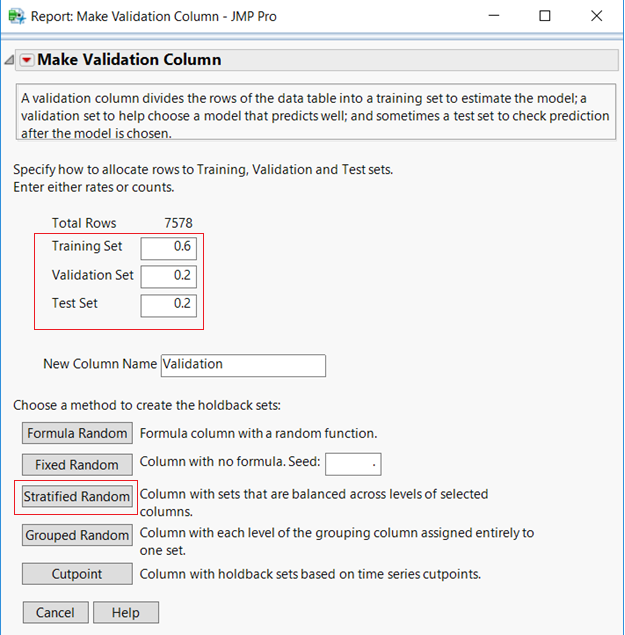


Figure 10

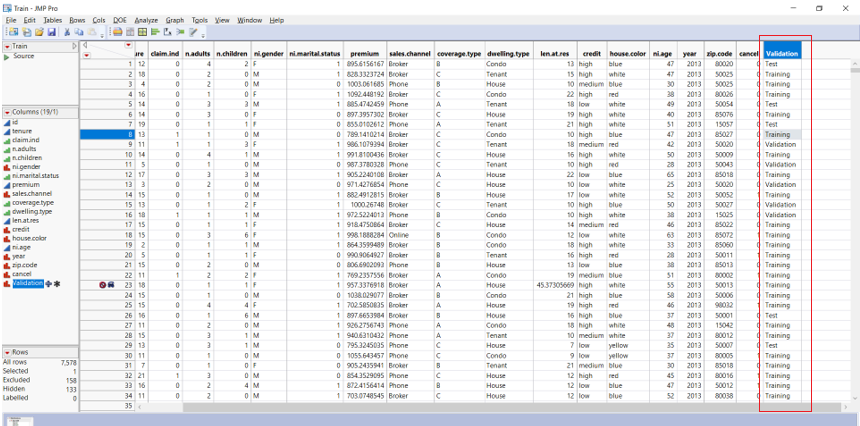
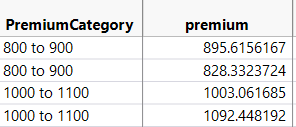


Figure 11

### Column Transformations

Bucketing:

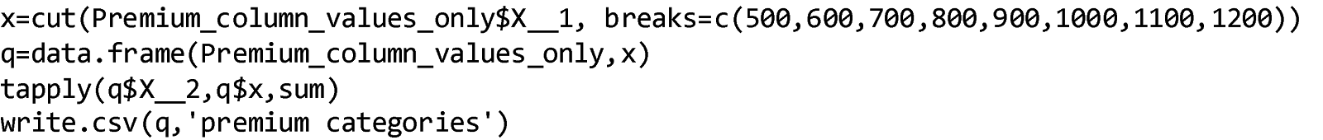
We recoded multiple continuous columns into categorical ones and analyzed the distribution of cancellations based on the category. For example, we split the premium column into multiple categories for values in intervals of 100. So, we made intervals like [600 – 700), [700-800) …… [1100,1200). This was done because we considered both the continuous as well as categorical cases one by one during our variable selection. While training, the neural network models performed better with the bucketed columns, while for the tree based models, the continuous columns performed better.

We tried this approach with the following columns.

1. Premium
2. Age
3. Tenure

Figure 12

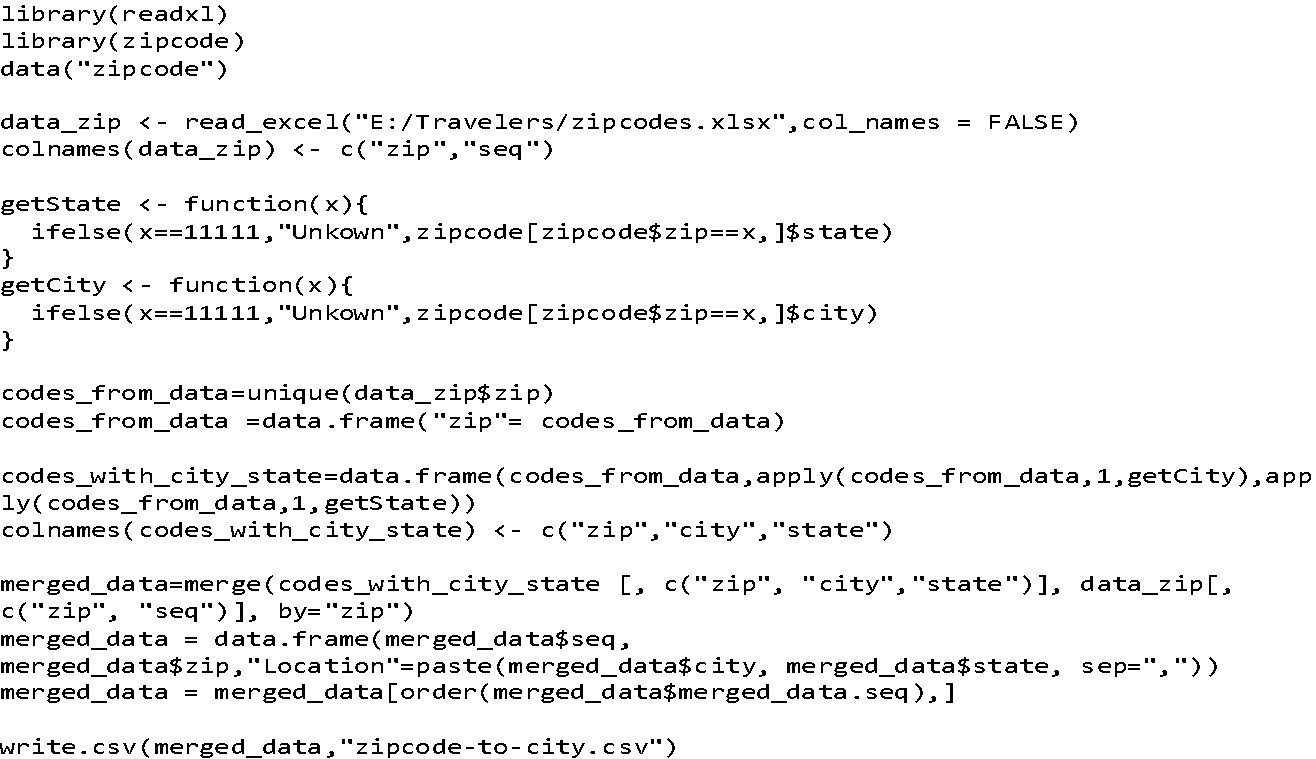
**R code used for premium:**



Zip code Mapping:

In total there are 222 unique values for zip codes present in the dataset. 222 factors are quite large and because of which their individual contributions are not very significant, more over these do not make a lot of sense because they represent a very specific small geographical area. To tackle the large number of factors, we derived the states from various zip codes. This made more sense because the number of factors decreased to 6 from 222. While this did improve the performance, there was a possibility that the cancellations might be related more to the cities of residence. So, we also created a column called Location which has the city and the state information in it. The state information was used again, because there was a chance that multiple states might have had cities with same names. Using the location column, we decreased the number of factors from 222 to 127.

R code used:



# Variable Selection

* To avoid redundant predictors from being used in the model, we have not used “Zip code” column. Moreover, the data in the form of zip code was not very insightful.
* “Year” column in the training data had values within the range 2013 to 2016. Using this column to train the model would restrict the model’s performance, if the values in test data don’t fall within this range. Hence, we have not included “Year” column in our model

We explored different model types viz. logistic regression, neural networks, decision trees, bootstrap forest and boosted trees. For bootstrap forest, decision trees and boosted trees, JMP platform has an in-built feature selection.

For logistic regression and neural networks, we tried Forward Selection, Backward Elimination and Mixed Stepwise methods for variable selection, of which forward selection method was observed to produce minimal set of predictors with improved model efficiency.



Table 1



Table



Table



Table



Table

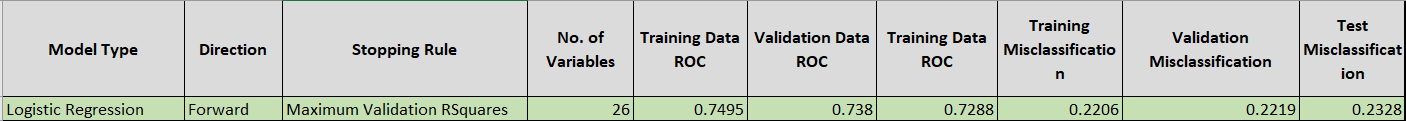
# Final Model Statistics

Logistic Regression models were observed to perform better in identifying the customers who are more likely to cancel their policy. Multiple logistic regression models were created using various Stopping Rules, Directions and Rules. From these Logistic Regression models, the best performing model was selected based on the ROC curve (Area under the curve) and the misclassification rates for all the three datasets, Training, Validation and Test

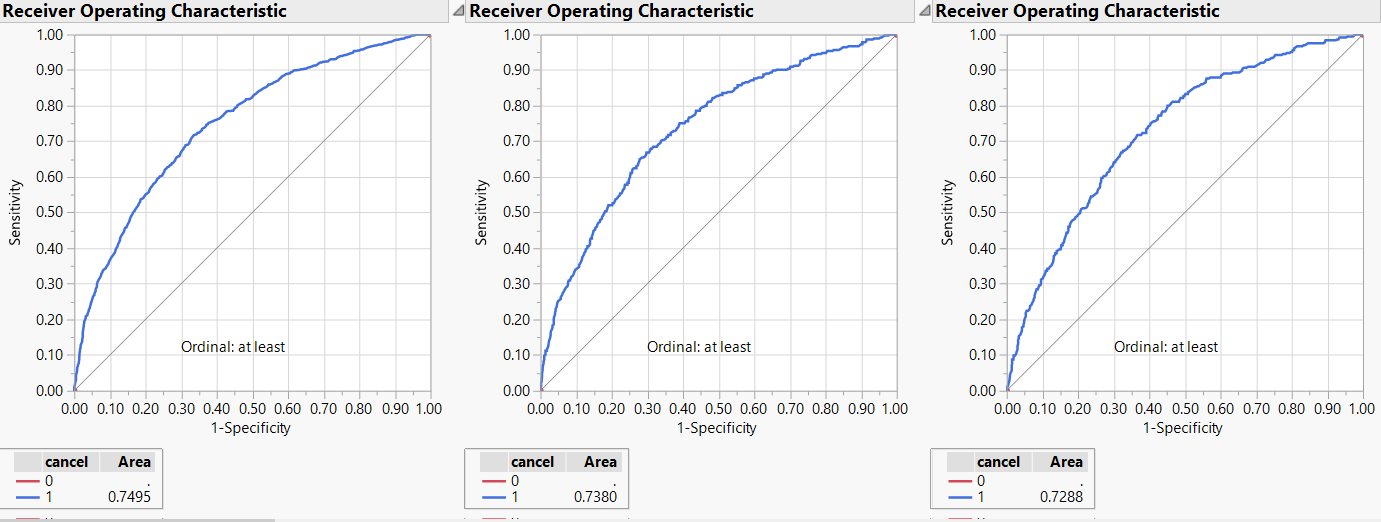
The columns used in our final model (for all the categorical column which are used, if they have K values, K-1 indicator columns are created and used in the model):

1. Credit
2. Location
3. Sales.channel
4. N.children
5. Ni.age
6. Claim.ind
7. Marital.status
8. Tenure
9. Coverage.type
10. N.adults
11. Len.at.res

The properties of the final logistic model selected is highlighted below:



**Training**                                                **Validation**                                             **Test**



https://lh4.googleusercontent.com/kw7hlvsOmg0IJOBOGc0KfP8oySSC6Kad3vyMQPpIz9OgVa28cWA9CoHf53VI01siTKop9SH6jV-04qGR0kjRcorxR1P5EuSH4IcBopp4GjxnwElX6Fuw9tgjuXpIgeCF-evbRCnWI7rStmiDXw

Figure 13

The overall c-stat of the training data is:

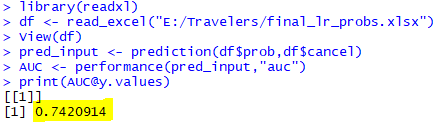


Figure 14