**Background/Objective:**

This project is an unstructured project where you will identify data, pose a question to be modeled, build models and assess model fit. I am requiring a model that predicts an *estimate of a binary target*. Your goal is to take me through your thought process in as many screenshots as necessary with your notes and interpretations along the way.

**Groups:**

* This project is to be done in groups of 2 to 4. 10% deduction for each person above 4 group members. **Register your group in eCentennial.**
* I will allow individuals to work alone, but your group consultation time with me will be cut in half.
* The group should assign one group leader who is responsible for turning in the final group project.
* Additionally, each group member is to upload a peer evaluation *individually to their own assignment dropbox*.

It is up to the groups to decide how to work in a group (i.e. roles and responsibilities). However, it is to be initially agreed that however the work is divided, each group member initially has an equal opportunity to achieve 100% participation on the peer evaluation. (Note: In extreme circumstances, I reserve the right to deduct more than 20% due to extreme feedback on peer evaluations.)

**Data:**

* Search the internet for a dataset of interest that will allow you to build a model to predict a binary outcome (i.e. logistic regression). Kaggle.com, Stats Canada, US Census are examples of data sources.
* At a minimum, I suggest a dataset with at least 200 records and at least 6 independent variables. However, I require that **your optimal decision tree must contain at least 3 significant variables**.
* When sending data for approval indicate what your target variable will be and include a link to where you found the data.
* Data must be approved by 3:30PM Thursday, December 8th.

**Email me (**[**dparent@centennialcollege.ca**](mailto:dparent@centennialcollege.ca)**) with the details on the data you will be using** and the website where you found it. No group will be allowed to use the same dataset, so first come first served. Please check the document for datasets already claimed prior to submitting data. Selection and approval of data is worth 4 of the 30 total marks for this assignment.

**Details:**

Complete a full data mining and predictive modeling project and report. At a **minimum** your report should contain:

- A decision tree model. At a minimum show screenshots for splitting the training and validation dataset, the maximal tree, and the optimal tree.

- A logistic regression model. At a minimum how screenshots for imputation, replacement and transformations, if necessary. If not necessary, so screenshots that support your decisions.

- A neural network model. At a minimum show screenshots for full neural network as well as the neural network model generated from only the variables retained by a regression model.

- Provide an assessment of the models with screenshots and a conclusion.

- Provide a screenshot of the final diagram.

- Be sure to cite your data source.

- All analysis should be completed in Enterprise Miner. (Data import into SAS may be performed in Enterprise Guide.) *No data manipulation may be done in Excel without my explicit approval.*

I encourage you to provide comments on your decision making, intuition and interpretation of results either “along the way” and/or in a summary report and appendix.

**Deliverables:**

- Submit the raw data (if a link is not available) and the SAS dataset used in your project to the assignment dropbox.

- Submit the data approval email from me.

**-** Submit a typed report of your analysis including screenshots. You are not limited to just the screenshots above; share enough to fully cover your thought process. As with building models, the least complex report that tells the full story is a good place to start. There is no page minimum or maximum.

**Assistance:**

**-** This is a group project. I expect each group to work independently and abide by the academic integrity policies: <https://www.centennialcollege.ca/pdf/mycentennial/AcademicHonestyPolicy.pdf>

**Submitted by: Pushkar Mishra**

**Student ID: 300980265**

**GROUP 1 (Single Member).**

Dataset: Suggested by David, Obtained from Kaggle:

<https://www.kaggle.com/datasets/blastchar/telco-customer-churn>

Original File Name as listed on Kaggle: WA\_Fn-UseC\_-Telco-Customer-Churn.csv

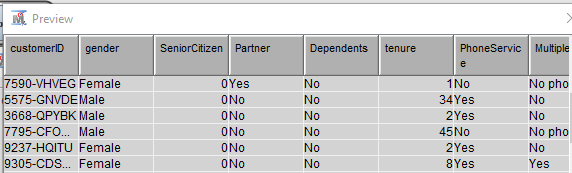
Business Issue: The telecom company is looking for a “Prediction Model” predicting the “Churn” of their customers as they are looking for ideas to get retention offers for their retention team to save those customers who will be tagged for “Churn” as the output of this project.

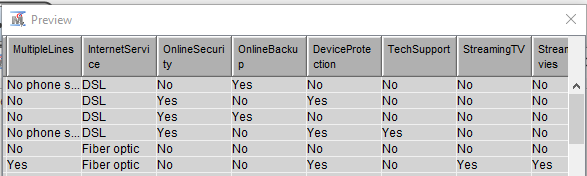
This project is limited to predict churn, and NOT for any retention offers as that would be the assignment for subject matters expert, who would look at the plans, packages, devices, services and the monetary impact of the offers to those customers whom the company tags as “Eligible for Churn Offers” following the suggestions and recommendations of this project.

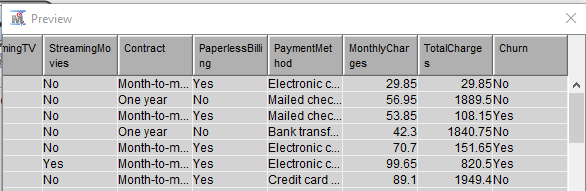
1. **FILE IMPORT NODE:**

Dataset Description and Exploratory Data Analysis:

Dataset Preview and Populated Columns:

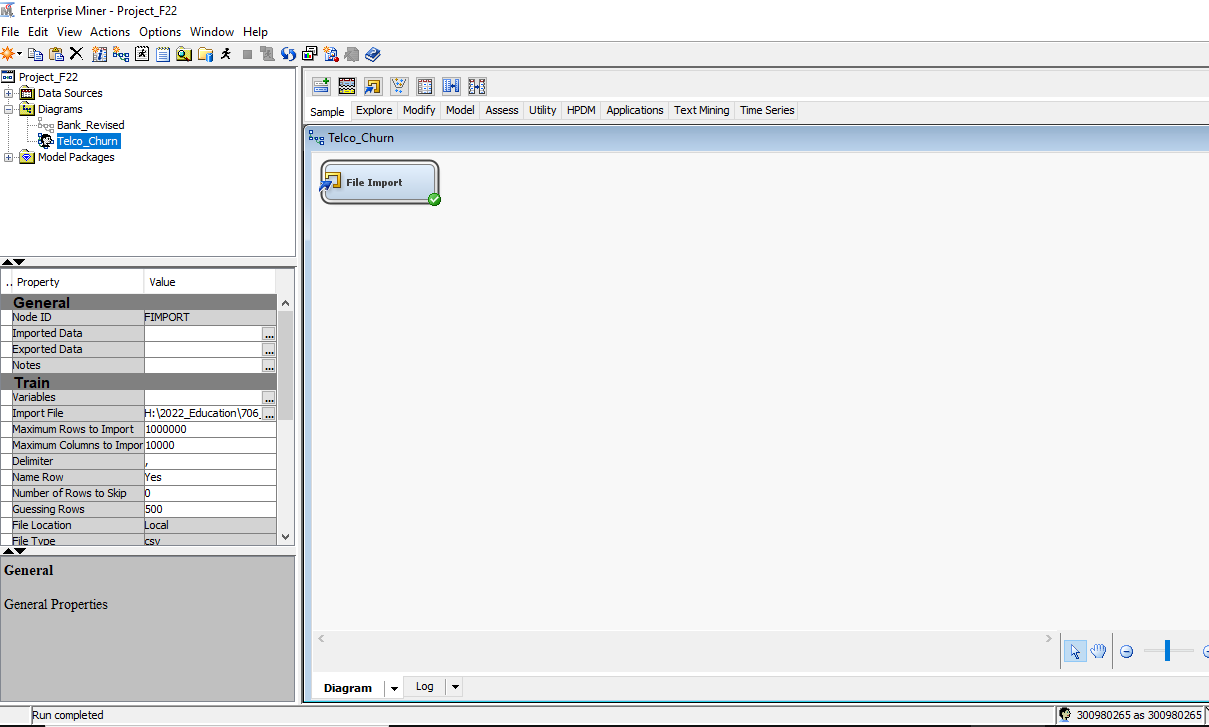




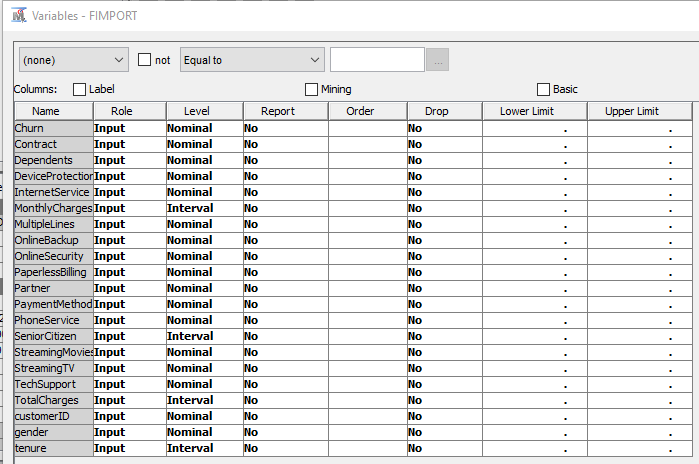


No Data has been manipulated anywhere outside of SAS Enterprise Miner and the entire dataset has been adopted in “As Is Condition”.

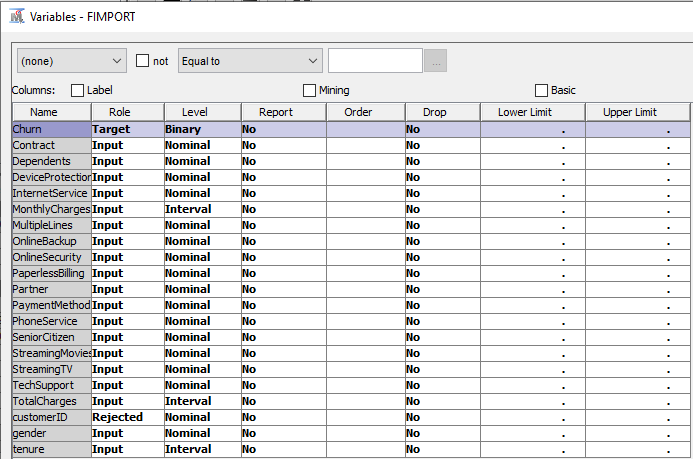
Populating the “File Import” Node and Importing the csv file:



**VARIABLES, their Roles and Levels in it’s Original Format after the file import in SAS EM:**

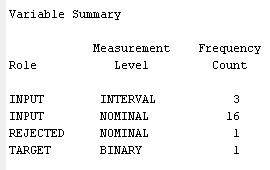


**VARIABLES with their Changed Roles and Levels, manipulated in SAS EM:**

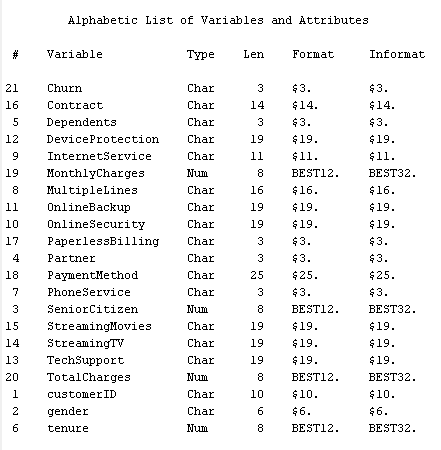


**SET BINARY TARGET VARIABLE: “CHURN”.**

Variable Summary after setting the target and rejecting the customer id:

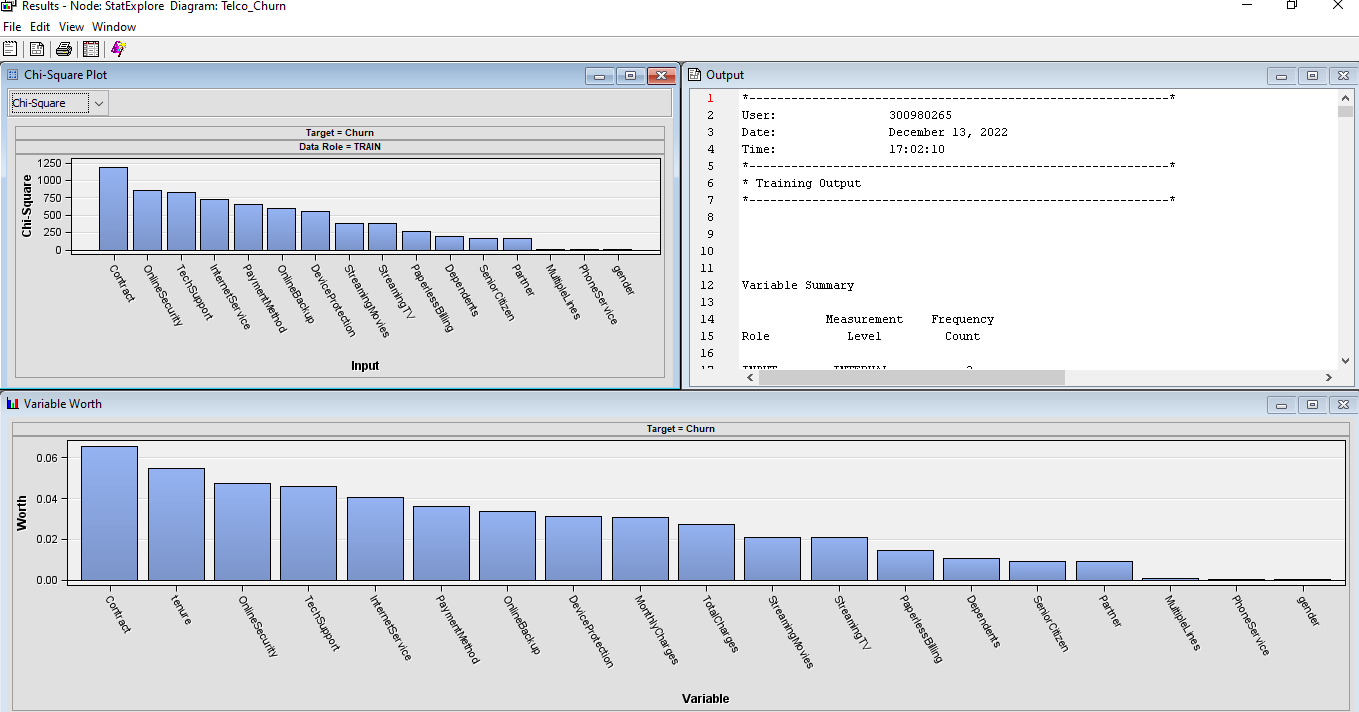


Alphabetic List of variables and their attributes as defined by SAS EM:

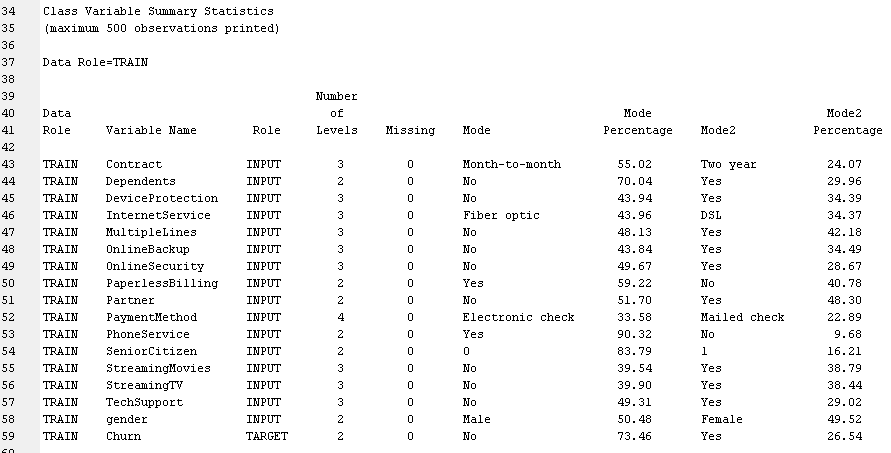


1. **STAT EXPLORE NODE:**

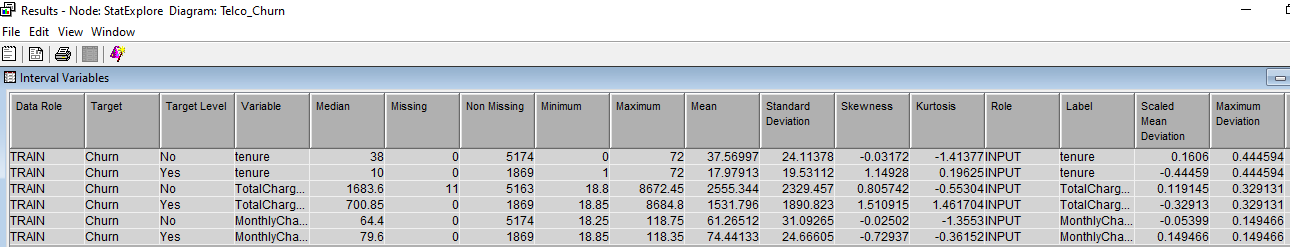
Initial Exploratory Results after attaching “Stat Explore” Node to the dataset:



Class Variables Summary Statistics:



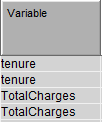
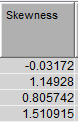
INTERVAL VARIABLES:



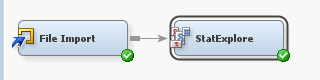
Interval Missing Variables:

Total\_Charges: Total 11 in number.

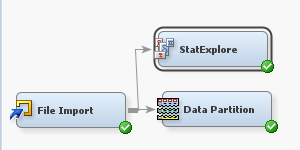
Interval Skewed Variables: 2 variables, viz. tenure and TotalCharges, does show the skewness, which is well defined as the customers staying with the company longer will have a higher numerical value, and accordingly their TotalCharges would be higher than a non-tenured or a new customer.

There is no need for any recoding to be done as the Exploratory Data Analysis using the Stat Explore Node was trustworthy.

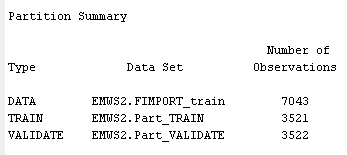


1. **DATA PARTIITION NODE:**



Having 7,043 records, which is enough to have a 50 – 50 partition, the dataset is split accordingly as illustrated below.

Data Partition Summary:



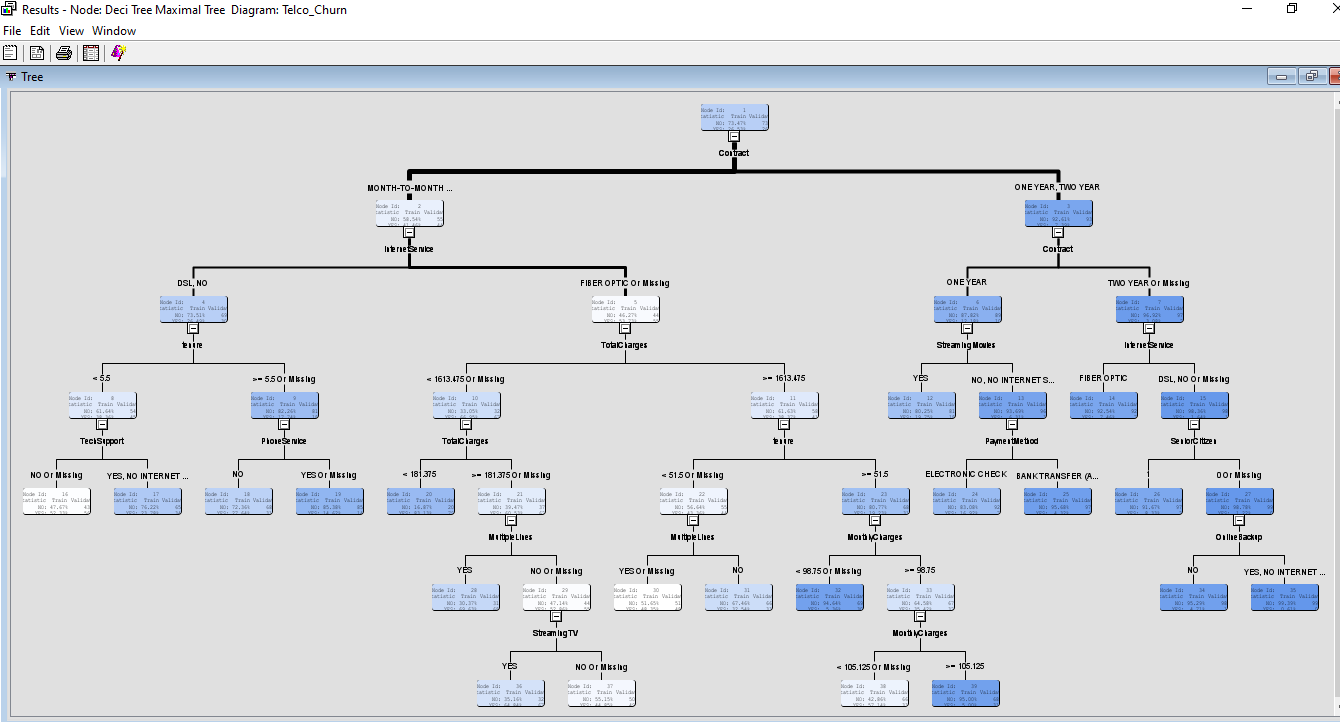
3 different kinds of “Decision Tree” would be called for and deployed, viz.:

1. Maximal Tree,
2. Classification Tree, &,
3. Probability Tree, or, Average Squared Error Tree in other words.
4. **DECISION TREE NODES – Deci Tree Maximal Tree:**

A: Maximal Tree: It’s defined as any split where the log worth is greater than 0.7, with the most important log worth as the top split. Interactive Method was not built, however the Optimizations Performed are –

1. Subtree method selected is Largest.
2. The Assessment measure is left as the default choice of Decision.

The “Full Tree”:

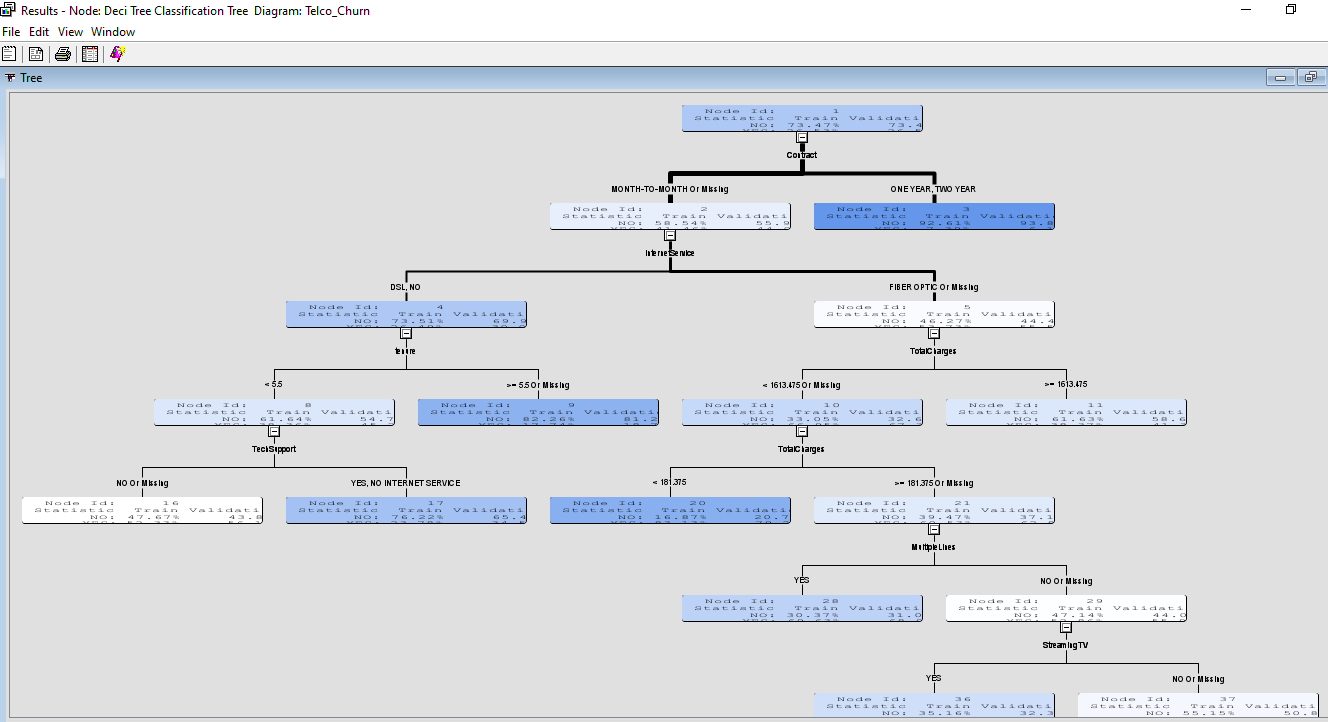


|  |  |  |  |
| --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.13797077110488032 |

1. **DECISION TREE NODES – Deci Tree Classification Tree:**

B: Classification Tree: We are looking for any pruning that could be done to make the model any better than before, hence we do the following optimization –

1. The Assessment measure is changed to Misclassification.

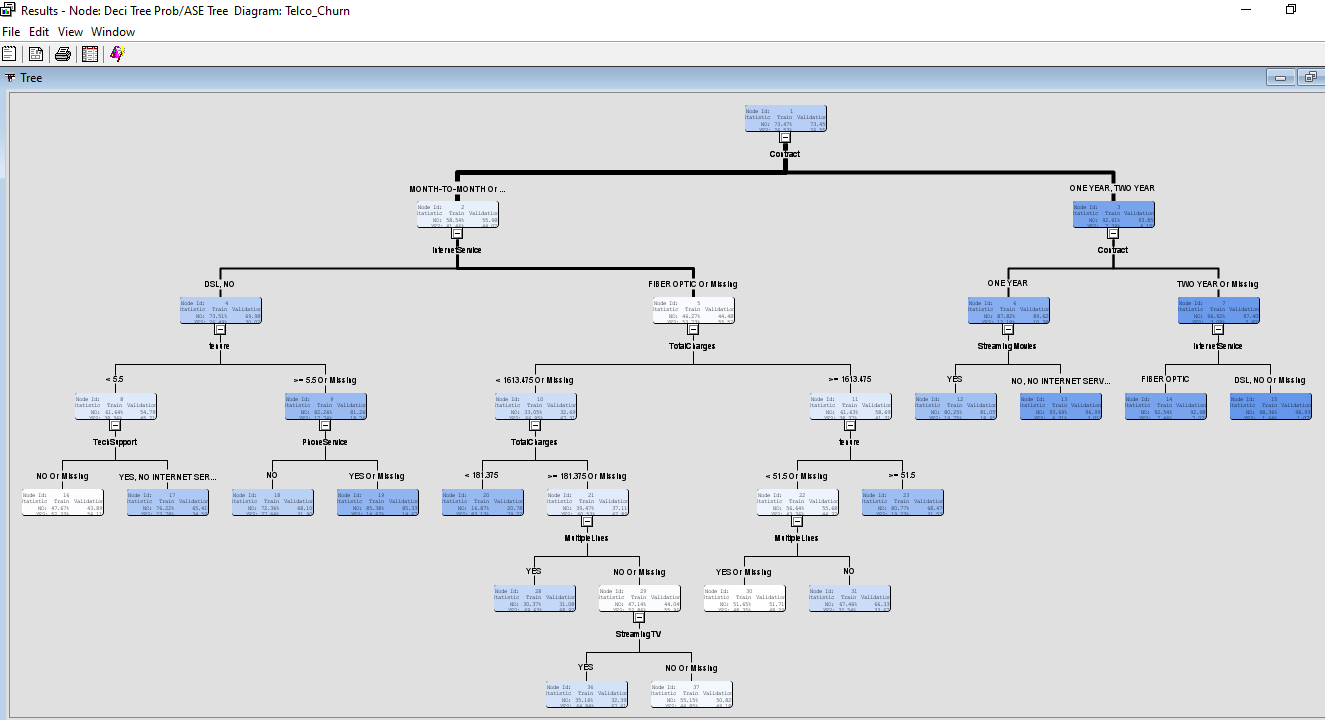


|  |  |  |  |
| --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 |

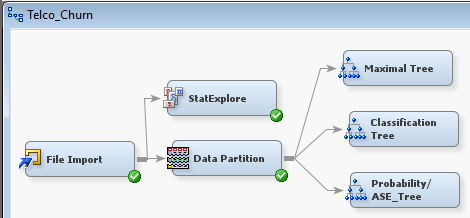
1. **DECISION TREE NODES – Deci Tree Prob/ASE Tree:**

C. Probability/ASE Tree: Trying one more kind of decision tree, we change –

1. The Assessment measure to Average Squared Error.



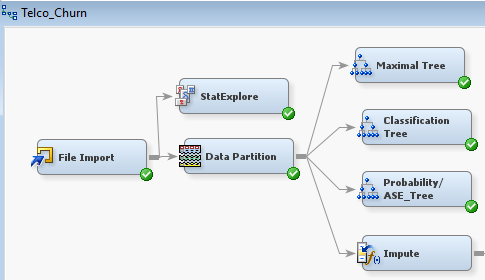
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |



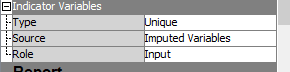
Since the Trees always start as the Maximal Tree, hence the Root Node and the First Split would be the same for all, but the following splits may or may not be different, depending upon the statistical method chosen.

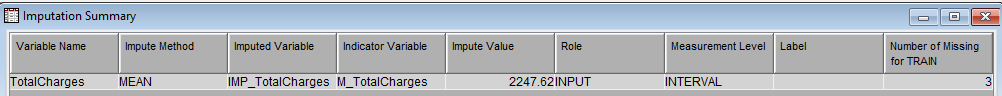
1. **IMPUTE NODE: (Handle the Missing Variables).**

The Impute Node will replace the Numeric Missing Variables (Interval) with the “Mean” and the Categorical Variables (Nominal) by the “Mode”. No tuning is required as SAS would identify the requirements as a default action by running the Impute Node.

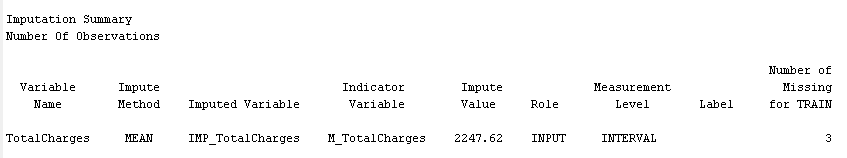


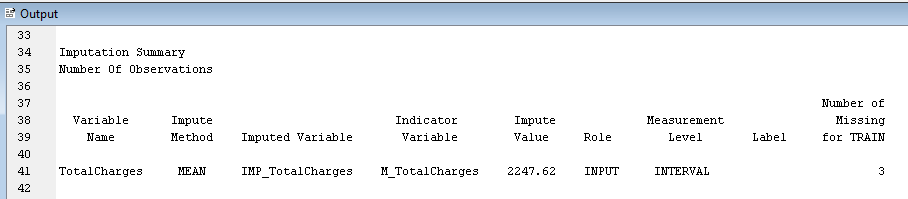
The results as displayed, indicates one interval variable had missing values (identified earlier) and is imputed by the mean value of the column.





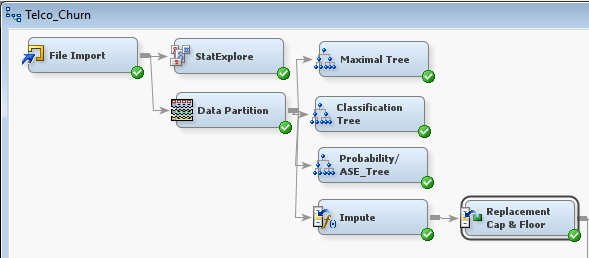
The summary:



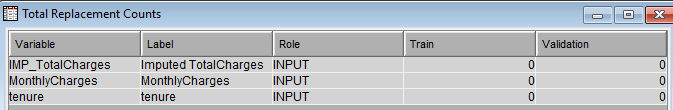


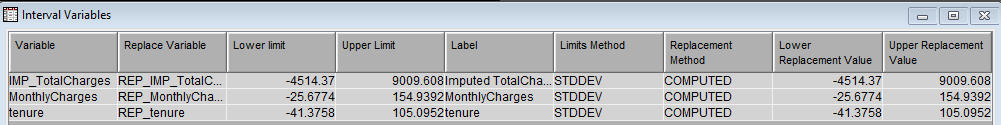
1. **REPLACEMENT NODE: (Setting the CAP & FLOOR to handle the SKEWED Datasets)**

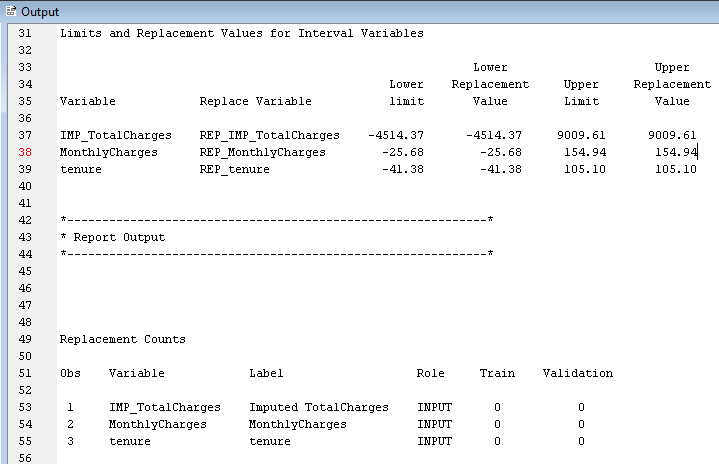
No optimization done as the replacement node will calculate the skewed data columns using the standard statistical method of 3 standard deviations on either side of the mean and would adjust it accordingly, producing the adjusted datapoints within the thresholds of skewness, i.e. Within the range of 1 or less.



The results state that none of the interval datapoints in the dataset were either capped (on the upper side), or floored (on the lower side) as no datapoint was found to be beyond the 3 standard deviations on either side. Hence NO REPLACEMENT WAS DONE.

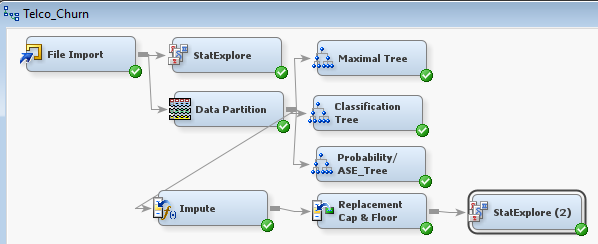


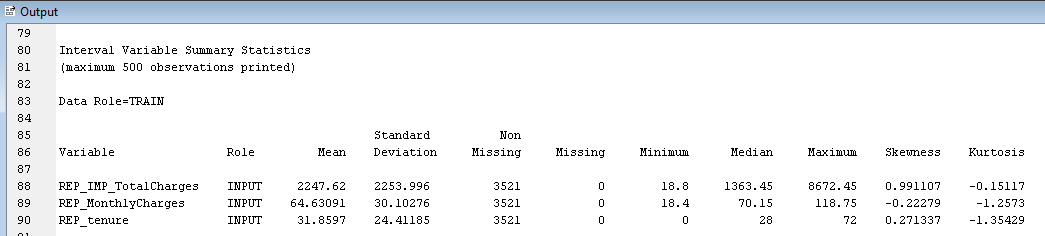




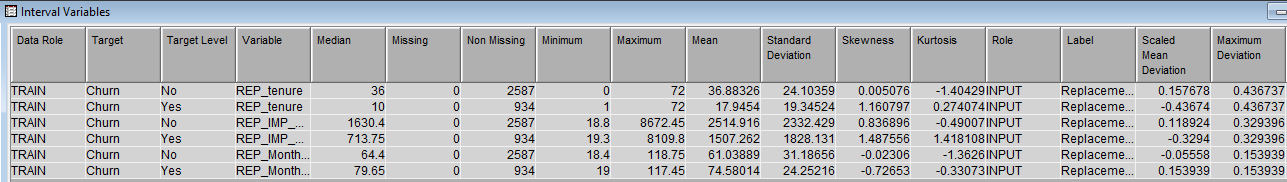
1. **STAT EXPLORE NODE off REPLACEMENT CAP & FLOOR:**

Verifying the dataset after running the replacement node to check the status of skewness following replacement.

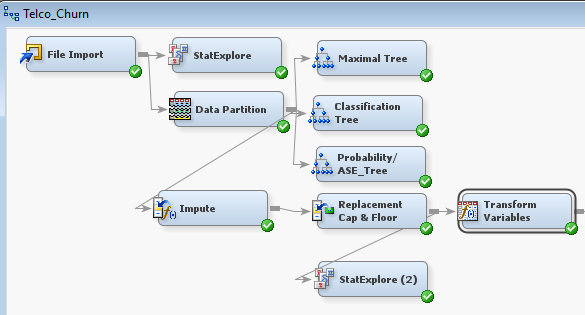




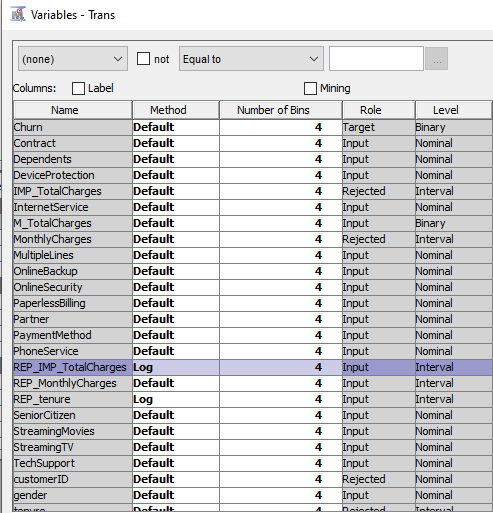
Checking the skewness on the interval variables from Stat Explore tells me that the Skewness for the 2 identified variables earlier, are still there as it was before, and that is quite intuitive since the cap and floor did not replace any value. This means we have to Transform these 2 variables to get it within the acceptable skew range. Running the Replacement Node has created New Variable Names for the 3 numeric interval variables that begins with REP\_.



1. **TRANSFORM VARIABLES NODE:**

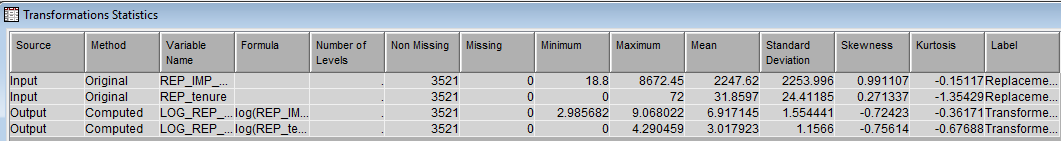


The 2 skewed variables as identified above and their Replacement variables names, viz. REP\_TotalCharges and REP\_tenure are set for Log Transformation to get the skewness within the acceptable threshold of 1 by getting into Edit Variables mode.

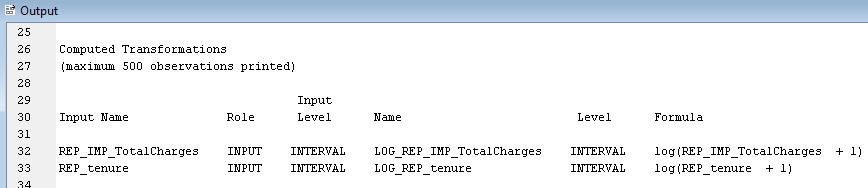


The results as shown below, has done 2 things:

1. Removed the skewness, as was the desired outcome, &,
2. Created 2 new variables, viz. LOG\_REP\_TotalCharges & LOG\_REP\_tenure.

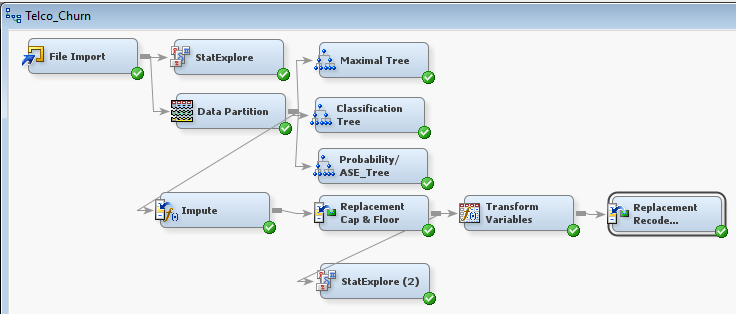


Had the skewness still persisted, then I could have employed either Standardization or Range Standardization. But it isn’t needed as the dataset is Not Skewed anymore with the new variables.

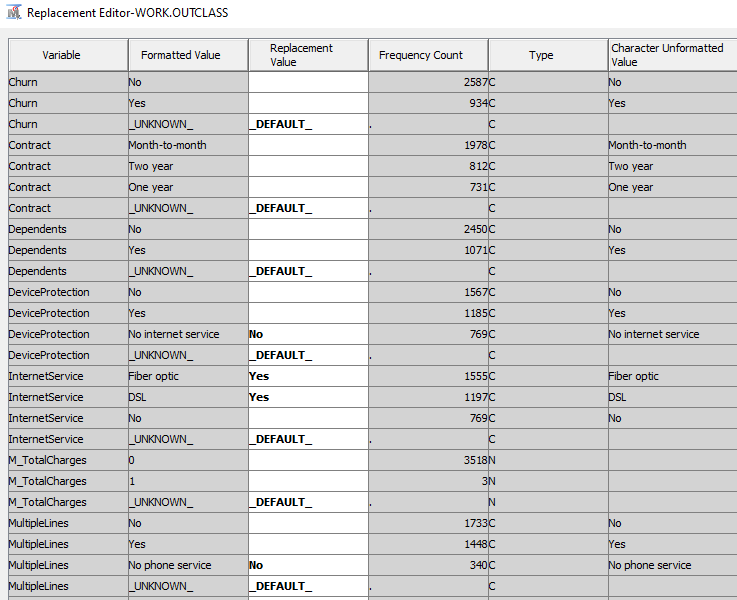


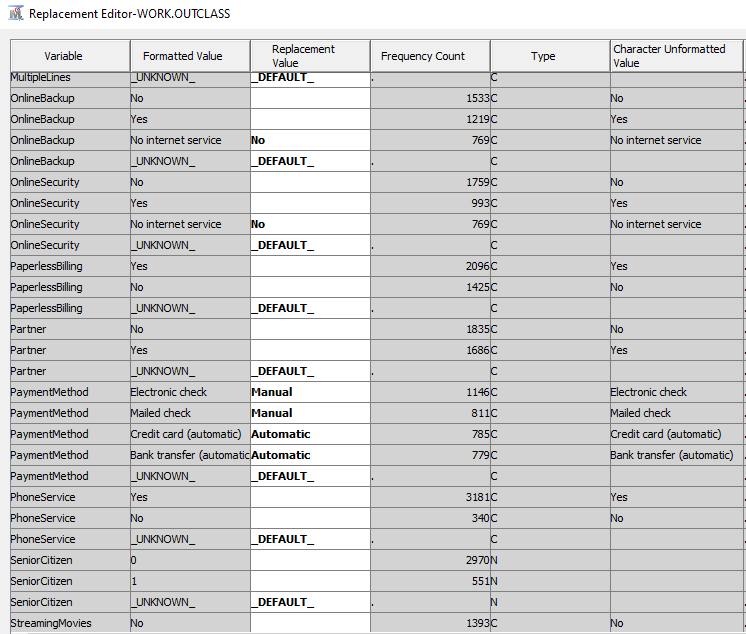
The skewness of one variable, the LOG\_REP\_tenure changed and went further down into a negative value.

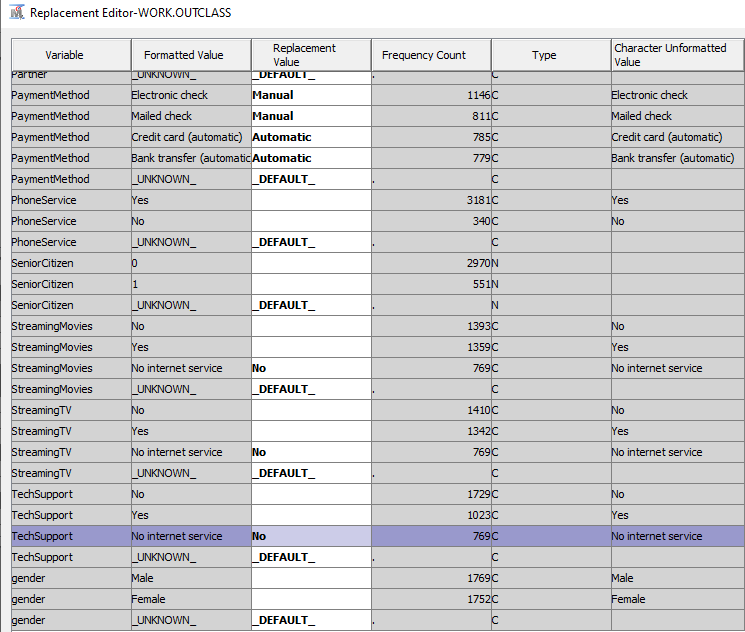
1. **REPLACEMENT NODE off Transform Variable Node to RECODE the Categorical Variables (Nominal) to create “DUMMY VARIABLES” for statistical computations and to reduce the “Curse of Dimensionality” by creating n – 1 number of variables for the same kind of Categorical Variables:**



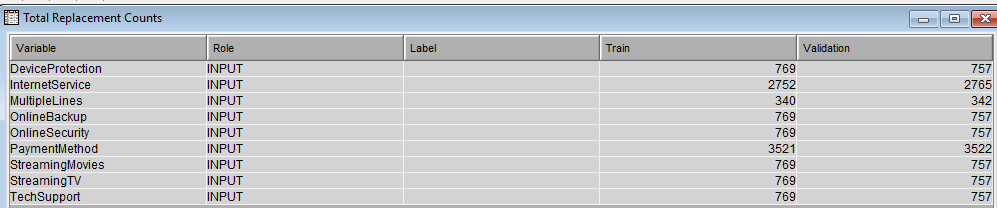
The node was altered with the “Default Limits Method” to None, instead of Standard Deviation from the Mean as we are taking care of Categorical Variables. Being a Class Variables or Nominal Variables, we need to tell SAS “HOW” we are going to do the recoding of dummies; hence we recoded and combined a majority of Class Variables as follows:



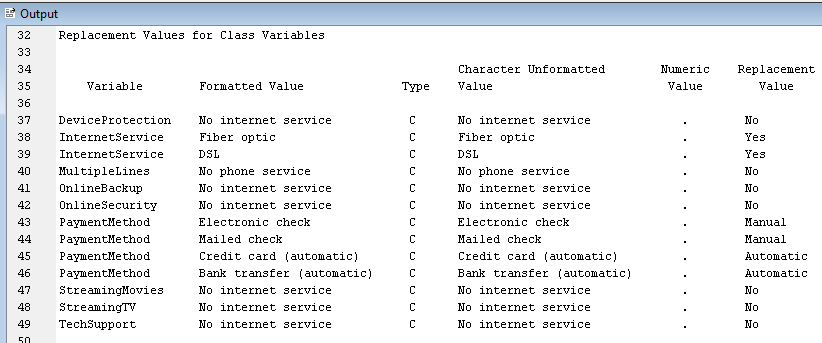




The results show that following Replacement were done and their respective counts are as follows:



And they are replaced with the following values, combining the similar variables into one or as required to reduce the “Curse of Dimensionality”:



So far, we have dealt with the following sequentially:

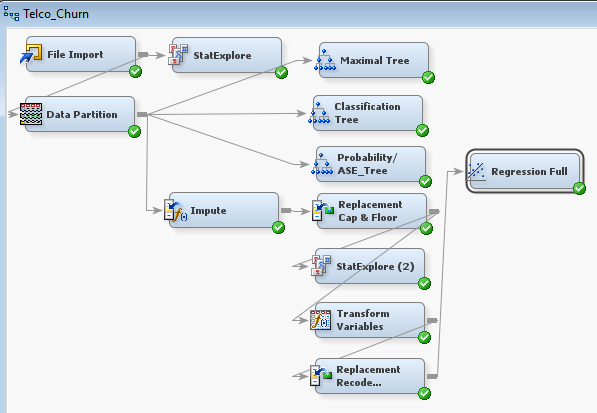
1. Impute node: Dealt with missing values,
2. Replacement node labelled as Replacement Cap and Floor: Dealt with extreme values,
3. Transform variables: Dealt with unusual distributions, &,
4. Replacement node labelled as Replacement Recode Dummies: Dealt with “Curse of Dimensionality” or minimizing Categorical responses.

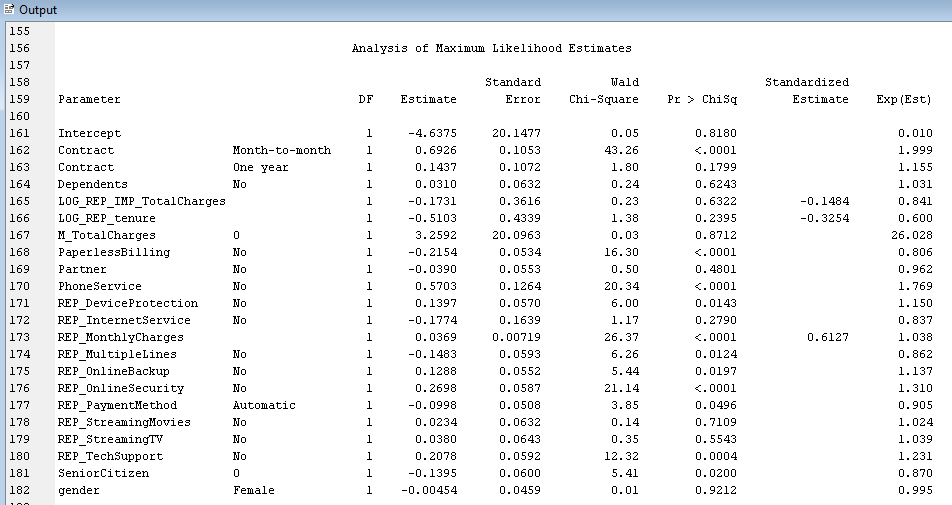
So now we are all set and ready to run our regressions, which I’ll be doing next.

1. **REGRESSION NODES – REGRESSION FULL:**

This method is used where ALL the variables are used and everything goes into the model. No tuning is done to run the model. SAS is capable of deciding if Linear or Logistic regression is needed to run based on the Target variable. In this case since we have a Binary target, hence a “Logistic Regression” will be deployed by SAS by default. The regression will be done with every single variable defined so far.



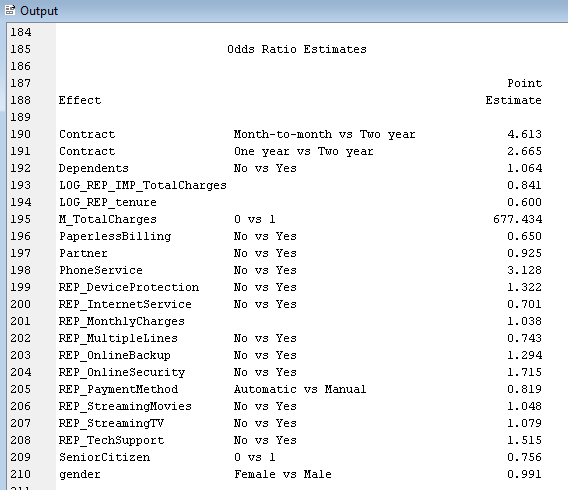




The results indicate the following variables are statistically extremely significant based on the threshold of 0.05 or less and they are listed in ascending order (any Pr > ChiSq < .0001 takes the highest precedence):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Parameter |  | DF | Estimate | Error | Chi-Square | Pr > ChiSq |
| Contract | Month-to-month | 1 | 0.6926 | 0.1053 | 43.26 | <.0001 |
| PaperlessBilling | No | 1 | -0.2154 | 0.0534 | 16.30 | <.0001 |
| PhoneService | No | 1 | 0.5703 | 0.1264 | 20.34 | <.0001 |
| REP\_MonthlyCharges | No | 1 | 0.0369 | 0.00719 | 26.37 | <.0001 |
| REP\_OnlineSecurity | No | 1 | 0.2698 | 0.0587 | 21.14 | <.0001 |
| REP\_TechSupport | No | 1 | 0.2078 | 0.0592 | 12.32 | 0.0004 |
| REP\_MultipleLines | No | 1 | -0.1483 | 0.0593 | 6.26 | 0.0124 |
| REP\_DeviceProtection | No | 1 | 0.1397 | 0.0570 | 6.00 | 0.0143 |
| REP\_OnlineBackup | No | 1 | 0.1288 | 0.0552 | 5.44 | 0.0197 |
| SeniorCitizen | 0 | 1 | -0.1395 | 0.0600 | 5.41 | 0.0200 |
| REP\_PaymentMethod | Automatic | 1 | -0.0998 | 0.0508 | 3.85 | 0.0496 |

The result tells us about the “Significance of the Relationship”, NOT the “Size of the Relationship”. The “Size of the Relationship” is measured by the “Odds Ratio” as below:



Reading the Odds Ratio, we can say that a Month-to-month customer or no contract person is 4.61 times more likely to leave as compared to a customer with a two-year term contract, or a customer with a one-year term is 2.66 times more likely to leave as compared to a customer with a customer with two-year term contract. The starkest observation is for the Binary M\_TotalCharges (0 versus 1) where a customer without Total Charges is 677.43 times more likely to leave than a customer with Total Charges, which does make sense as a customer on a month to month service, is definitely much more likely to leave when compared to a customer on contract with monthly charges.

With the Base Group of “Two Year”, we can see that both, the month to month or a customer with a 1-year contract, are much likely to leave, which is quite intuitive.

Odds Ratio Estimates:

Effect Point Estimate

Contract Month-to-month vs Two year 4.613

Contract One-year vs Two-year 2.665

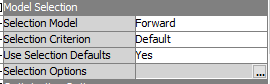
Some of the Negative relationships where customers are more likely NOT to leave and to stay are: PaperlessBilling, Partner, REP\_InternetService, REP\_MultipleLines, REP\_PaymentMethod (Automatic versus Manual), SeniorCitizen (0 versus 1) and gender (Female versus Male).

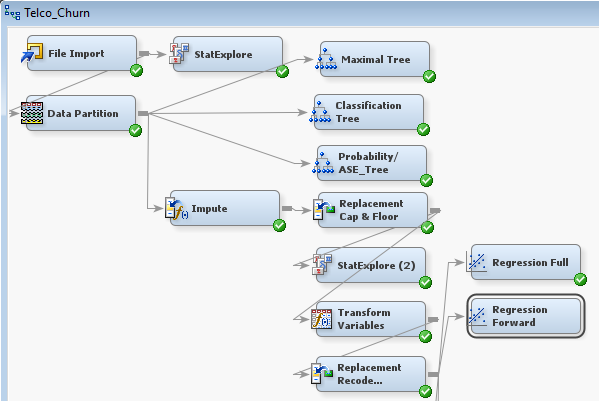
Since this model has all the variables included, so now we can try some other regressions hoping that the ASE and the Misclassifications gets better by tuning the variables and feeding in the most statistically significant variables only.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |

1. **REGRESSION NODES – REGRESSION FORWARD:**

Model Selection is changed to “Forward” from “None”.



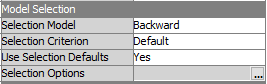


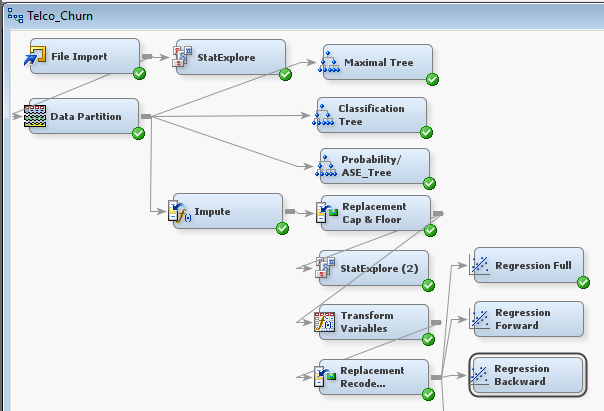


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |

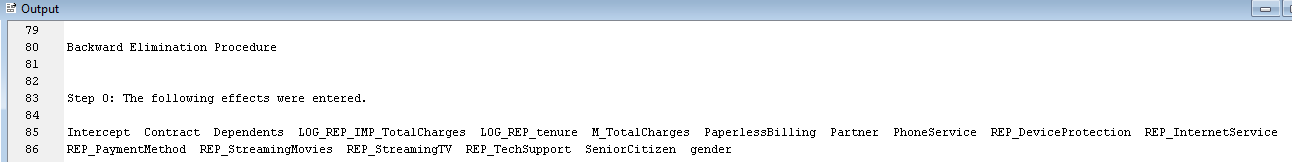
1. **REGRESSION NODES – REGRESSION BACKWARD:**

Model Selection is changed to “Backward” from “None”.





ALL of the variables were entered into the model to begin with and it followed the elimination process as it progressed, until it stopped.



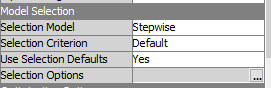


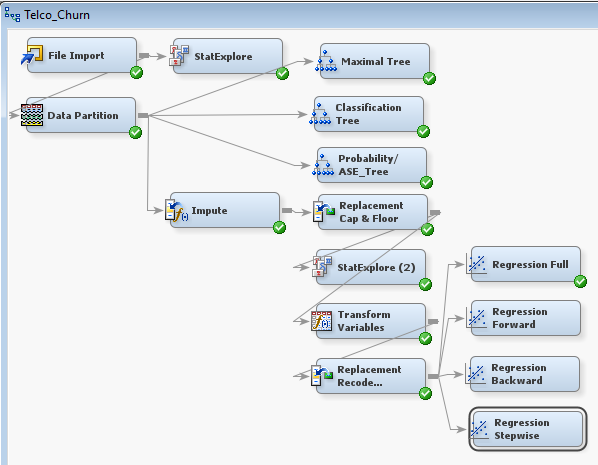


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |

1. **REGRESSION NODES – REGRESSION STEPWISE:**

Model Selection is changed to “Stepwise” from “None”.





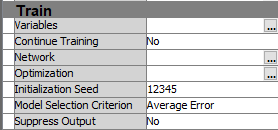


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |

1. **NEURAL NETWORKS – N N Impute:**

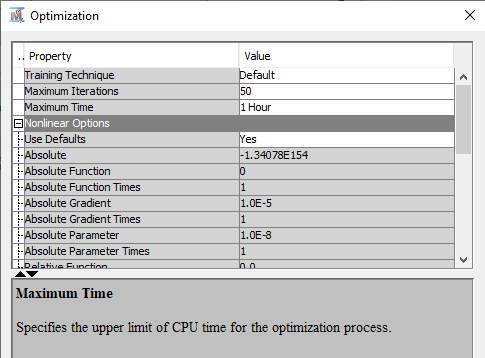
As a core characteristic, Neural Networks are very susceptible for imputations because of their non-acceptance of “Missing Values”. Neural Networks also have a set of regressions built in them, so they can be called off the Regression Nodes as well. This character will be utilized as well from the cleansed data that was done previously as that would help us create models that could be compared with other models to check if we get a better error or misclassification. Some “Optimization” would be done as follows:

1. Change the “Model Selection Criterion” from Profit/Loss to Average Error as this is the metric we are following from beginning.

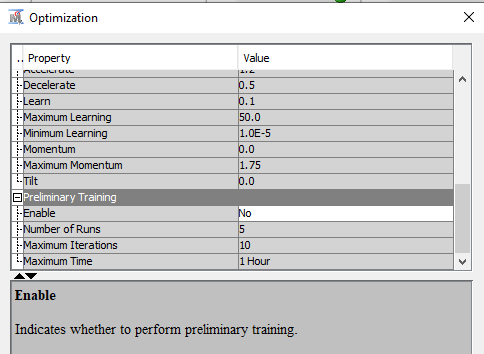


Optimization:

1. Optimize the Maximum Run Time to 1 hour instead of default 4.



1. Preliminary Training Enable set to “No”.



1. Iterations left at the default value of 50.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |

1. **NEURAL NETWORKS – N N Repl Cap & Floor:**

Even though the Neural Networks are not susceptible to uneven distribution, still it’s not a bad idea to run a model off the Replacement Cap and Floor to check the errors and see its viability in the project. Optimized as above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |

1. **NEURAL NETWORKS – N N Transform Vari:**

This node would see if the reduced “Curse of Dimensionality” does make our model any better or not. Optimized as above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |

1. **NEURAL NETWORKS – N N Full Repl Recode Dumm:**

Just like a Full Regression, this node runs the Neural Network utilizing the entire path that was used to run Regressions. Optimized as above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |

1. **NEURAL NETWORKS – N N Back Regr Def 3 Hid Units:**

Instead of using the full list of all the variables to create the Neural Network, we can use the list of the most significant variables as provided by the Regression modules and use their output as the list of input variables. While selecting the list, we can see which Regression module has the lowest errors. In this case the best error was the Full Regression with al the variables, but that is already used in the “N N Full Repl Recode Dumm” Node, hence I’ll be running the Neural Networks from the Backward regression and the Forward Regression separately. Optimized as above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |
| 12. | Neural Network | N N Back Regr Def 3 Hid Units | 0.131647 | 0.193356 |

1. **NEURAL NETWORKS – N N Forw Regr:**

Running this node to see if the variable selection makes the Neural Network any better or not as they are sensitive to Dimensionality and any reduction of variables may impact the error score. Optimized as above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |
| 12. | Neural Network | N N Back Regr Def 3 Hid Units | 0.131647 | 0.193356 |
| 13. | Neural Network | N N Forw Regr | 0.131847 | 0.193924 |

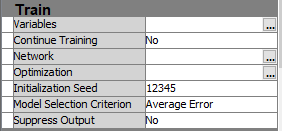
1. **NEURAL NETWORKS – N N Prob/ASE Deci Tree:**

Even though the ASE for all the Decision Tree nodes are much higher than any of the Regression Nodes, still I would like to see the Neural Network running off the best of the errors out of the 3 Regression nodes to check if the variable list coming out of the Decision Tree makes the Neural Network any better or not. Optimized as above.

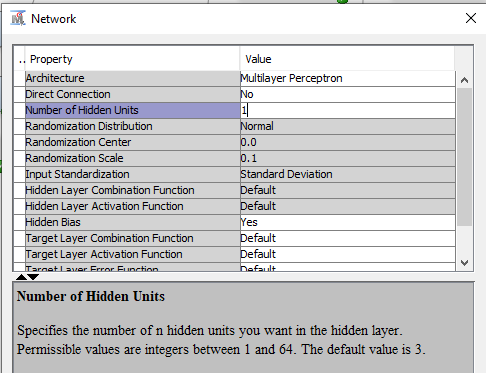
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |
| 12. | Neural Network | N N Back Regr Def 3 Hid Units | 0.131647 | 0.193356 |
| 13. | Neural Network | N N Forw Regr | 0.131847 | 0.193924 |
| 14. | Neural Network | N N Prob/ASE Deci Tree | 0.13539 | 0.200738 |

1. **NEURAL NETWORKS – N N Back Regr 1 Hid Units:**

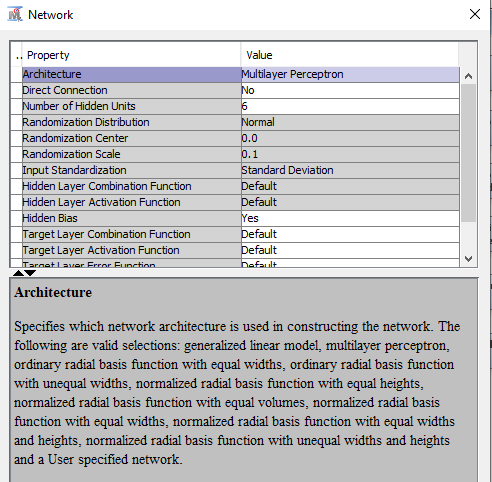
The Default assignment for a Neural Network is “2 Hidden Units”, but to see if we can get better models, we can manipulate the hidden units and we will be utilizing this feature and would run different number of Hidden Units to find better results. Optimized as above. In addition, under the “Network” settings, setting the “Network Size”, with the “Hidden Units” modified to 1 as follows:



And the “Hidden Units” to 1.



The “Architecture” is left with the default choice of “Multilayer Perceptron” as below.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |
| 12. | Neural Network | N N Back Regr Def 3 Hid Units | 0.131647 | 0.193356 |
| 13. | Neural Network | N N Forw Regr | 0.131847 | 0.193924 |
| 14. | Neural Network | N N Prob/ASE Deci Tree | 0.13539 | 0.200738 |
| 15. | Neural Network | N N Back Regr 1 Hid Units | 0.130611 | 0.19222 |

1. **NEURAL NETWORKS – N N Back Regr 2 Hid Units:**

Optimized as above. In addition, under the “Network” settings, setting the “Network Size”, with the “Hidden Units” modified to 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |
| 12. | Neural Network | N N Back Regr Def 3 Hid Units | 0.131647 | 0.193356 |
| 13. | Neural Network | N N Forw Regr | 0.131847 | 0.193924 |
| 14. | Neural Network | N N Prob/ASE Deci Tree | 0.13539 | 0.200738 |
| 15. | Neural Network | N N Back Regr 1 Hid Units | 0.130611 | 0.19222 |
| 16. | Neural Network | N N Back Regr 2 Hid Units | 0.131088 | 0.192788 |

1. **NEURAL NETWORKS – N N Back Regr 4 Hid Units:**

Optimized as above. In addition, under the “Network” settings, setting the “Network Size”, with the “Hidden Units” modified to 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |
| 12. | Neural Network | N N Back Regr Def 3 Hid Units | 0.131647 | 0.193356 |
| 13. | Neural Network | N N Forw Regr | 0.131847 | 0.193924 |
| 14. | Neural Network | N N Prob/ASE Deci Tree | 0.13539 | 0.200738 |
| 15. | Neural Network | N N Back Regr 1 Hid Units | 0.130611 | 0.19222 |
| 16. | Neural Network | N N Back Regr 2 Hid Units | 0.131088 | 0.192788 |
| 17. | Neural Network | N N Back Regr 4 Hid Units | 0.131126 | 0.190233 |

1. **NEURAL NETWORKS – N N Back Regr 5 Hid Units:**

Optimized as above. In addition, under the “Network” settings, setting the “Network Size”, with the “Hidden Units” modified to 5.

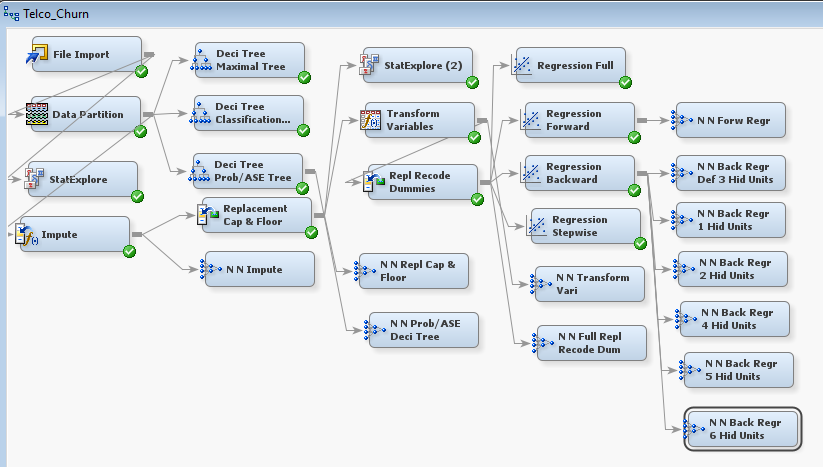
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |
| 12. | Neural Network | N N Back Regr Def 3 Hid Units | 0.131647 | 0.193356 |
| 13. | Neural Network | N N Forw Regr | 0.131847 | 0.193924 |
| 14. | Neural Network | N N Prob/ASE Deci Tree | 0.13539 | 0.200738 |
| 15. | Neural Network | N N Back Regr 1 Hid Units | 0.130611 | 0.19222 |
| 16. | Neural Network | N N Back Regr 2 Hid Units | 0.131088 | 0.192788 |
| 17. | Neural Network | N N Back Regr 4 Hid Units | 0.131126 | 0.190233 |
| 18. | Neural Network | N N Back Regr 5 Hid Units | 0.131333 | 0.193356 |

1. **NEURAL NETWORKS – N N Back Regr 6 Hid Units:**

Optimized as above. In addition, under the “Network” settings, setting the “Network Size”, with the “Hidden Units” modified to 6. Ran up till 6 hidden units to see if it gets any worse and would stop here.

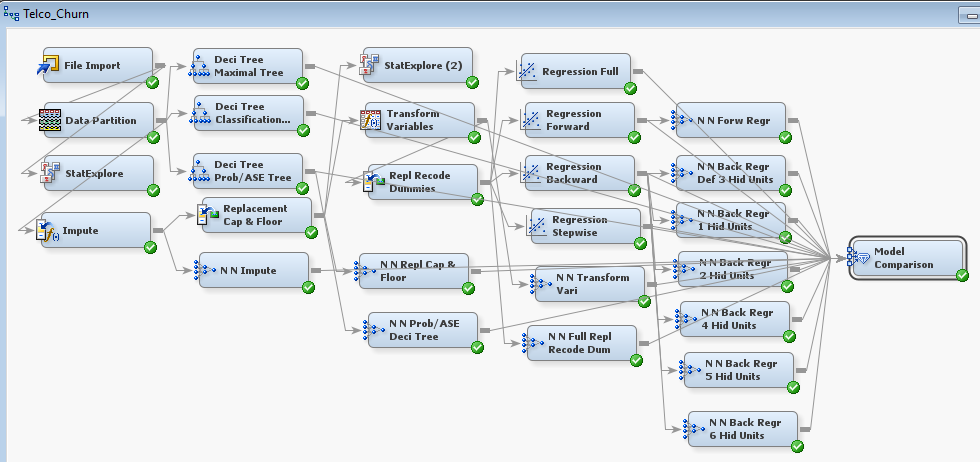
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model No. | Statistical Model Run | Node Name | Average Squared Error | Misclassification Rate |
| 1. | Decision Tree | Deci Tree Maximal Tree | 0.137970 | 0.205565 |
| 2. | Decision Tree | Deci Tree Classification Tree | 0.139646 | 0.202726 |
| 3. | Decision Tree | Deci Tree Prob/ASE Tree | 0.136356 | 0.202726 |
| 4. | Logistic Regression | Regression Full | 0.130661 | 0.193924 |
| 5. | Logistic Regression | Regression Forward | 0.131003 | 0.194776 |
| 6. | Logistic Regression | Regression Backward | 0.130608 | 0.193924 |
| 7. | Logistic Regression | Regression Stepwise | 0.131003 | 0.194776 |
| 8. | Neural Network | N N Impute | 0.134773 | 0.193072 |
| 9. | Neural Network | N N Repl Cap & Floor | 0.137251 | 0.199035 |
| 10. | Neural Network | N N Transform Vari | 0.132015 | 0.195344 |
| 11. | Neural Network | N N Full Repl Recode Dumm | 0.132641 | 0.193356 |
| 12. | Neural Network | N N Back Regr Def 3 Hid Units | 0.131647 | 0.193356 |
| 13. | Neural Network | N N Forw Regr | 0.131847 | 0.193924 |
| 14. | Neural Network | N N Prob/ASE Deci Tree | 0.13539 | 0.200738 |
| 15. | Neural Network | N N Back Regr 1 Hid Units | 0.130611 | 0.19222 |
| 16. | Neural Network | N N Back Regr 2 Hid Units | 0.131088 | 0.192788 |
| 17. | Neural Network | N N Back Regr 4 Hid Units | 0.131126 | 0.190233 |
| 18. | Neural Network | N N Back Regr 5 Hid Units | 0.131333 | 0.193356 |
| 19. | Neural Network | N N Back Regr 6 Hid Units | 0.133423 | 0.197047 |

It’s visibly clear that the “N N Back Regr 4 Hid Units” is the Best of the Neural Networks for the Hidden Units as its getting worse by adding more hidden units.

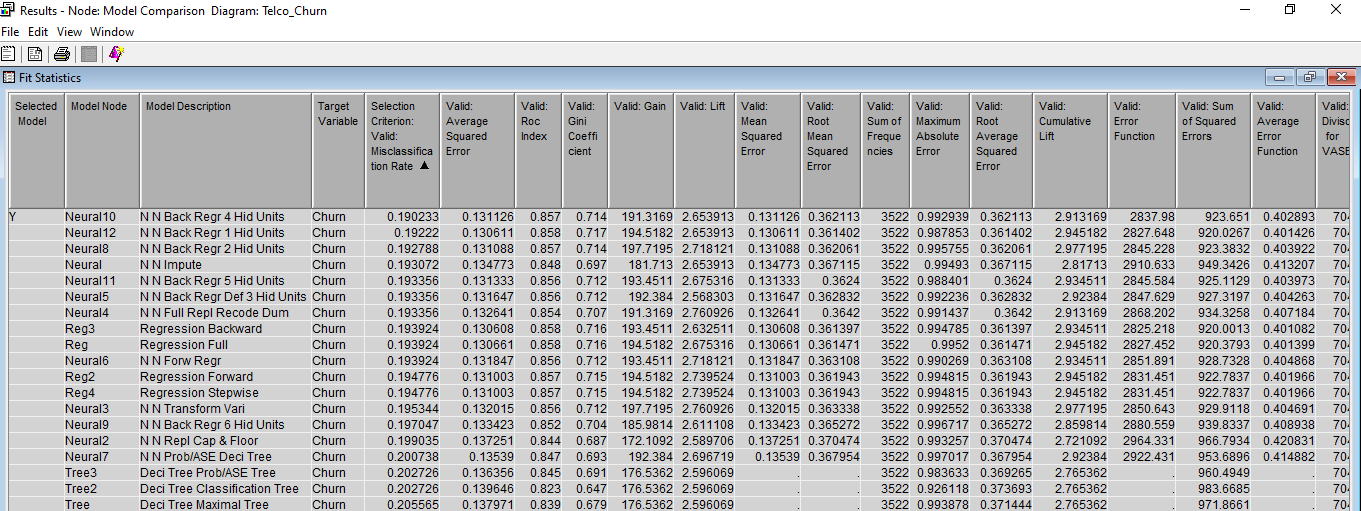


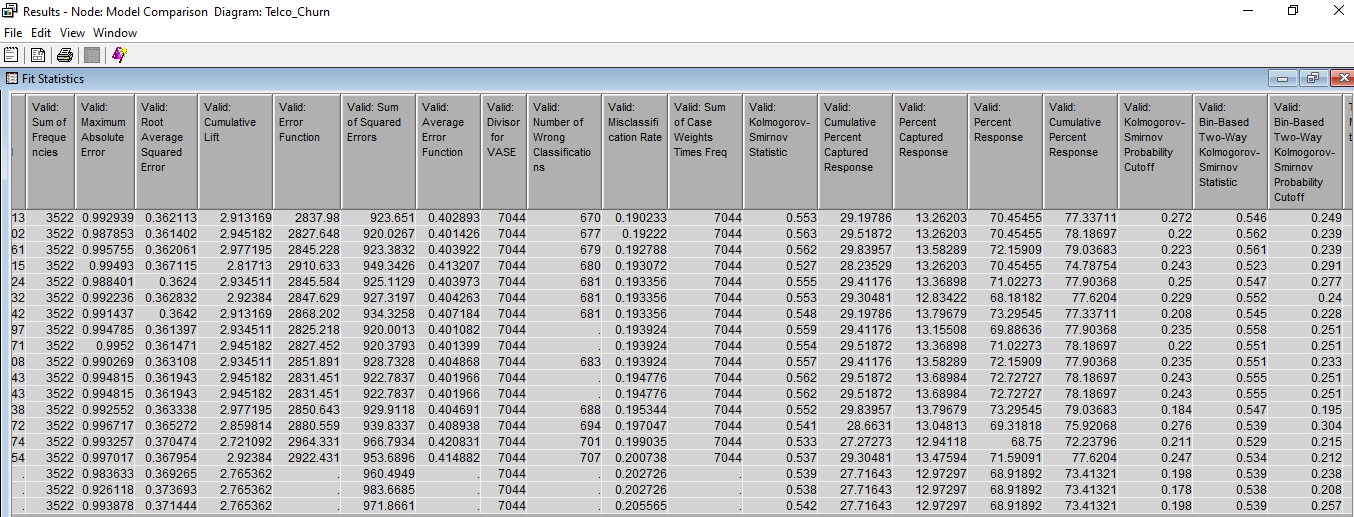
1. **MODEL COMPARISON:**

Now comes the time to look at ALL the “Statistics” from ALL the models created so far, compare them side by side and let SAS select the “CHAMPION MODEL” as the BEST model from ALL the models deployed in this project. No optimization done.

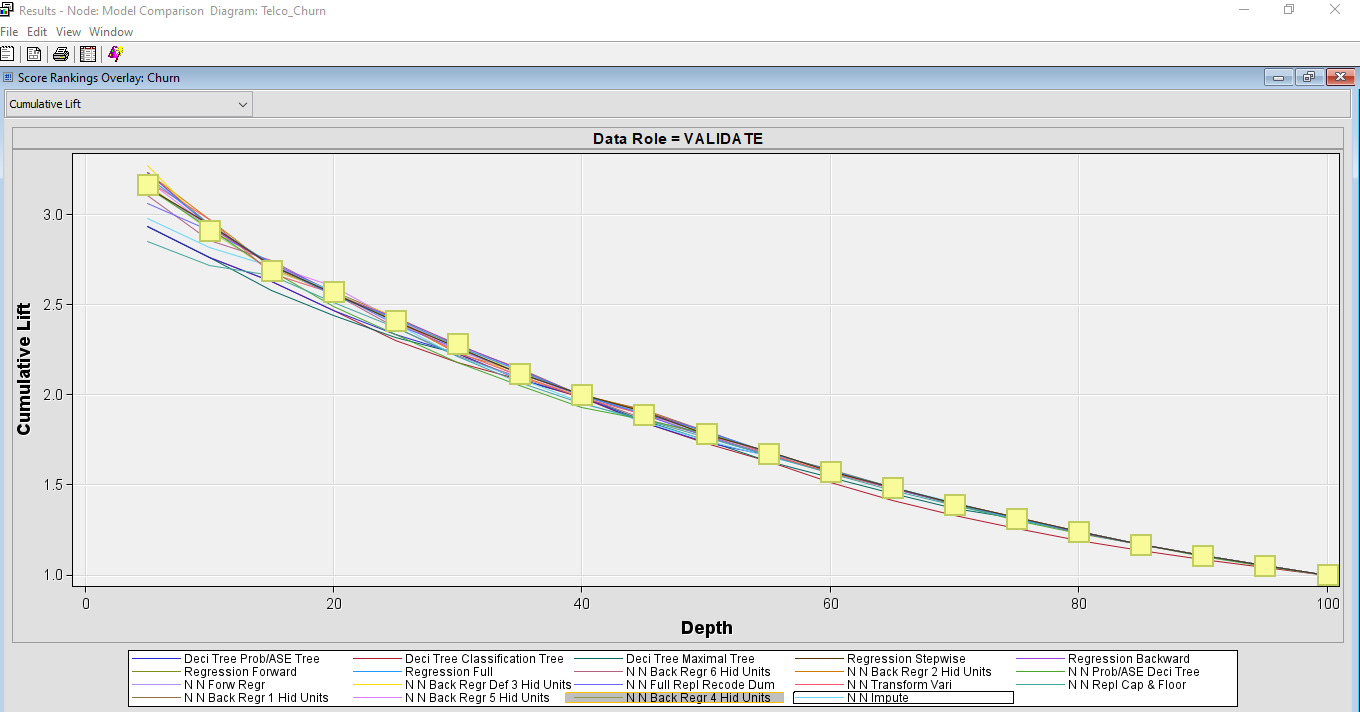


The “Fit Statistics” Results window as displayed below, summarizes all the Statistical Summaries of all the models created and run, along with the “Selected Model” by SAS. It also identifies the reason for selection, which in this case is the “Misclassification Rate”. Only the Relevant Validation Columns are selected from the results.

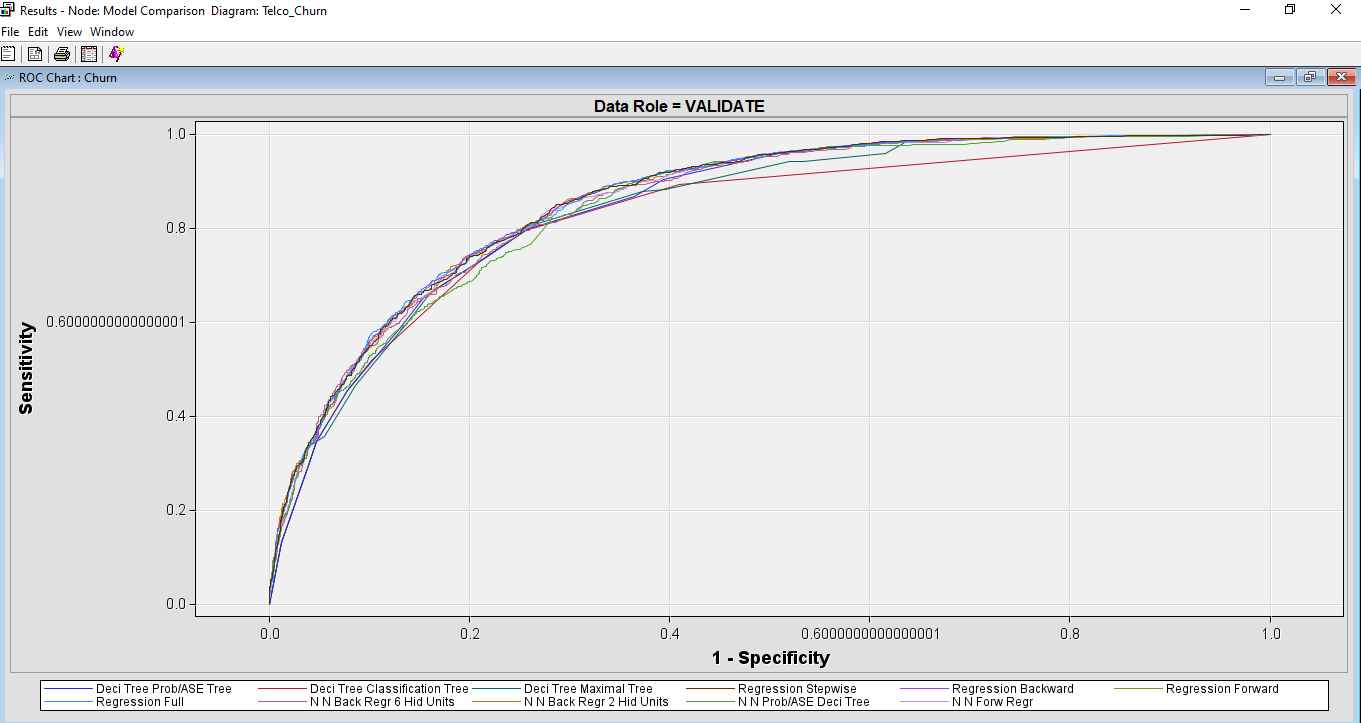




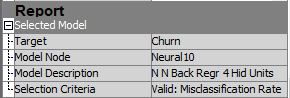
Marked “Cumulative Lift” of the selected model:



The “ROC Chart” is NOT displaying the ROC Chart for the selected model to be identified and mark it separately to show its relative comparison with the other models.



The **SELECTED MODEL**: **“4 HIDDEN UNITS NEURAL NETWORK OFF BACKWARD REGRESSION NODE”.** The **SELECTION CRITERIA**: **“MISCLASSIFICATION RATE”.** The **“ROC Index”** for the selected model is 0.857 which is relatively in the higher bracket and states that 85.7% of data is defined by the curve and is quite reasonable and acceptable output for the selected model.



CONCLUSION:

After running all the models as described above, the model stated above is selected based on the criteria as mentioned and is proposed to the telco company for their “Churn” reduction. Further models can be created looking after the Monetary recommendations and other business considerations for the customers individually. The misclassification rate score of 0.190233 for the selected model is also acceptable as 81% of the times the model is classifying the customers correctly for True Positives and True Negatives on the validation dataset.

**David’s Comments**

Overall Feedback

Optimal trees done correctly but stated "the following splits may or may not be different, depending upon the statistical method chosen" which isn't quite correct as they may be the same or fewer but not different.

Adjusting outliers done correctly, but stated "producing the adjusted datapoints within the thresholds of skewness, i.e. Within the range of 1 or less."  Cap and floor adjusts extreme values to 3 standard deviations not based on skewness.

Nice work recoding to collapse nominal variables to fewer dimensions.

Flipped interpretation of 0 and 1 for the missing indicator.  0 indicates charges were present.  I would have liked to see discussion about how to interpret log transformed variables.

Very thorough optimization of Neural Networks.

Since you included the lift chart, I would have liked to see more discussion about it.  Additionally, I would have liked to see a comparison of ROC and ASE in your assessment section to further show if they agree.

I would have liked to have the conclusion be more of a "stand alone" section where we could read it without going back to other sections referenced.  Additionally, I would have liked more discussion of variables driving churn rates in this section with potential recommendations on how to use them to mitigate churn.

Some small typo issues (e.g., "viz. tenure", "log worth" instead of "logworth").  I can definitely see where you were going by including the new models in the comparison table as you went, but it might have been a bit easier to follow by only including for each model type and then using the full table in the assessment section.