|  |
| --- |
| **MIS 6334.501**  **Advanced Business Intelligence**  **Prof.Geng**  **Group 7** |
| Data and Model Analytics Using SAS Enterprise Miner |
| Project Final Report |
|  |
| **Siyang Zan, Kaier Ying, Jingwen Zhang, Xuerong Zhang** |
| **10/02/2015** |

Table of Contents

[Part I. Basic Data Preprocessing 3](#_Toc431848811)

[Part II. Prediction Models, Model Comparison and Champion Model Evaluation 6](#_Toc431848812)

[1. Prediction Models 7](#_Toc431848813)

[*a. Decision Tree* 7](#_Toc431848814)

[*b. Regression* 8](#_Toc431848815)

[*c. Neural Network* 9](#_Toc431848816)

[2. Model Comparison 10](#_Toc431848817)

[3. Model Evaluation 10](#_Toc431848818)

[Part III. Improve Model Performance 11](#_Toc431848819)

[1. Data Modification and Input Selection 12](#_Toc431848820)

[2. Data Imputation 14](#_Toc431848821)

[3. Transform Variables 14](#_Toc431848822)

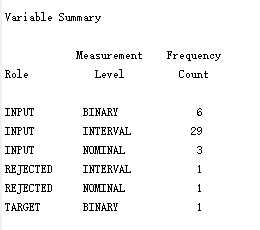
[4. Data with high variance 15](#_Toc431848823)

[5. Data Sampling 16](#_Toc431848824)

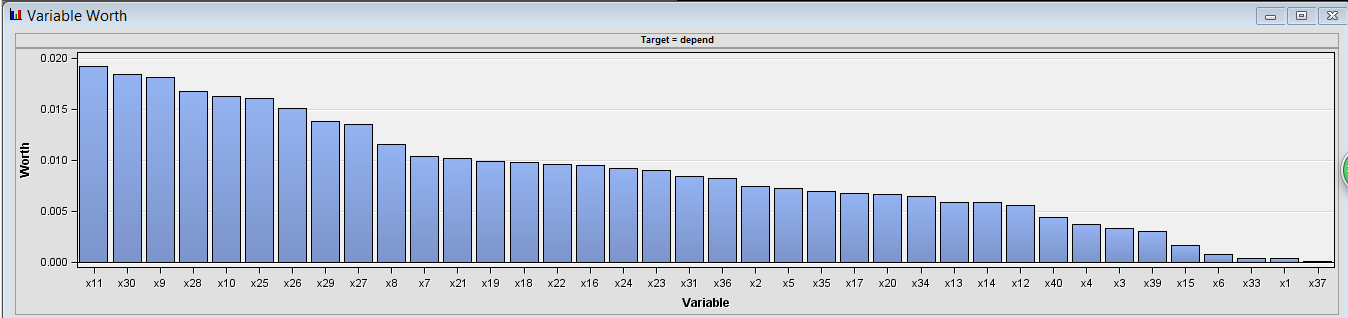
[6. Ensemble 17](#_Toc431848825)

[Part IV. Summary 18](#_Toc431848826)

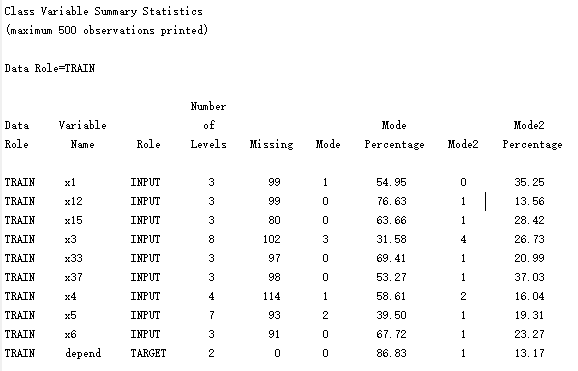
# Part I. Basic Data Preprocessing

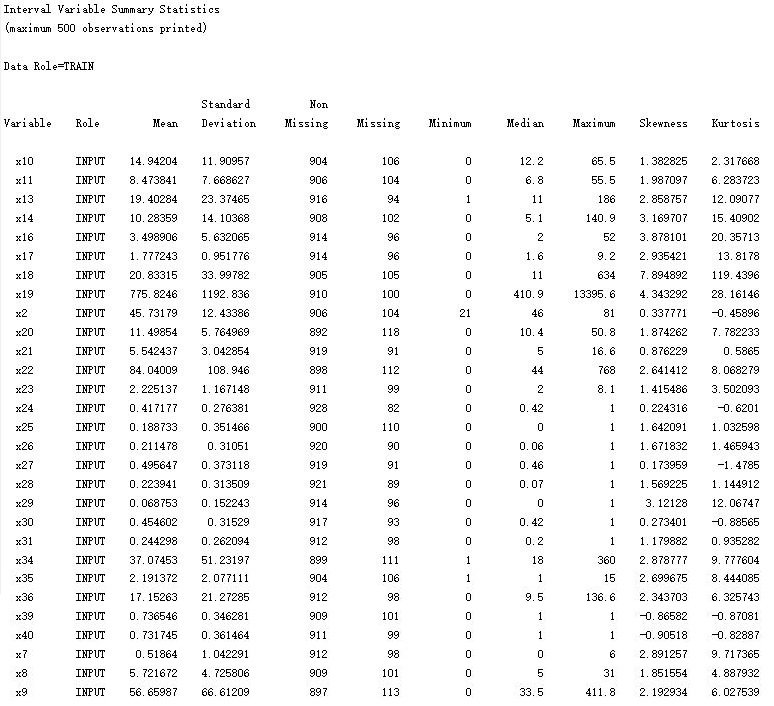


There are 41 variables in this data set, one binary target, 29 interval inputs, 9 nominal inputs and two rejected variables (rejected because of too many missing values).

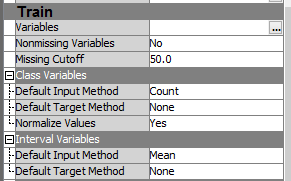


Among the 38 inputs, x11, x30 and x9 have the highest variable worth and could probably be most useful in predicting the target response.



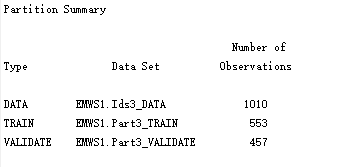


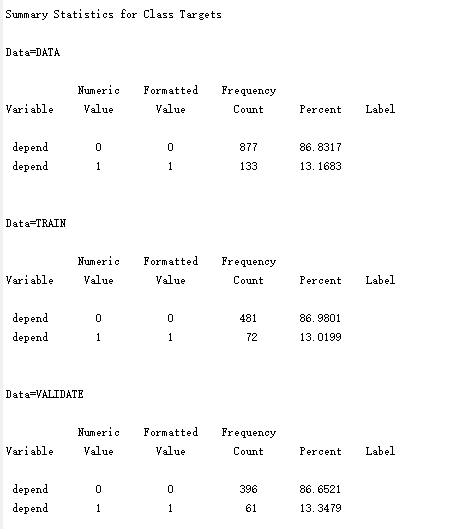
By looking at the Class Variable Summary Statistics and the Interval Variable Summary Statistics, we find there are approximately 100 missing values for each input.



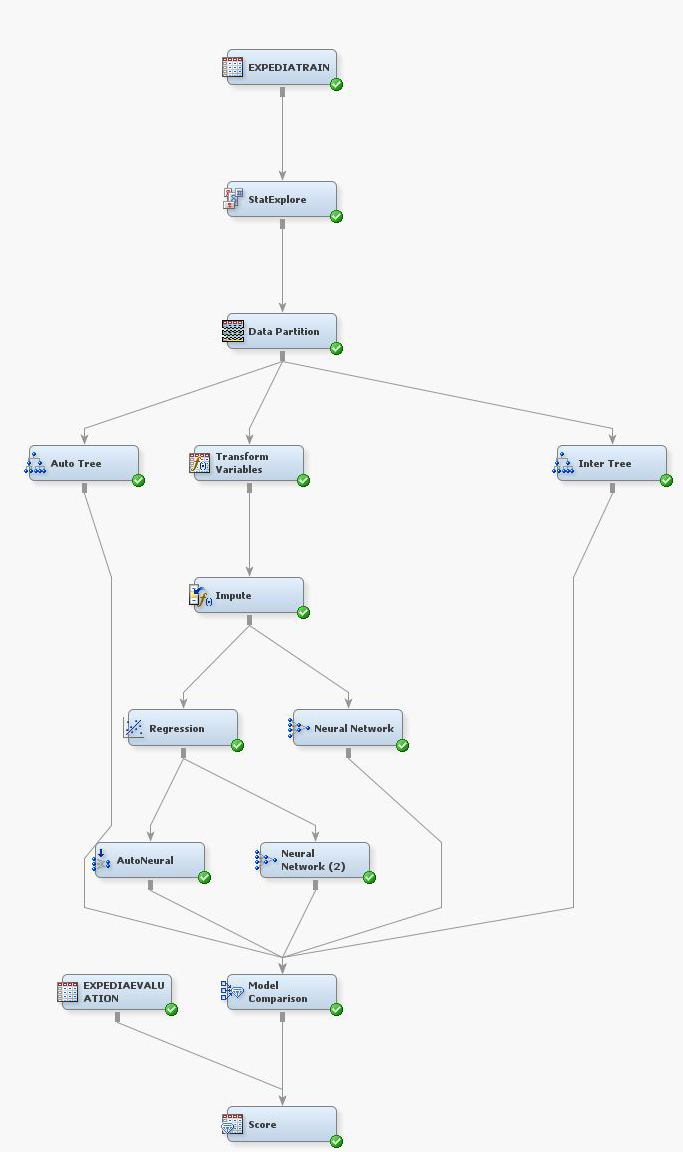
Therefore, for regression and neural network, we replace the missing value with the mean of the non-missing values for the interval input and replace the missing value with the most frequent category for the nominal input.

Here is the result of data partition, 55% training data and 45% validation data.



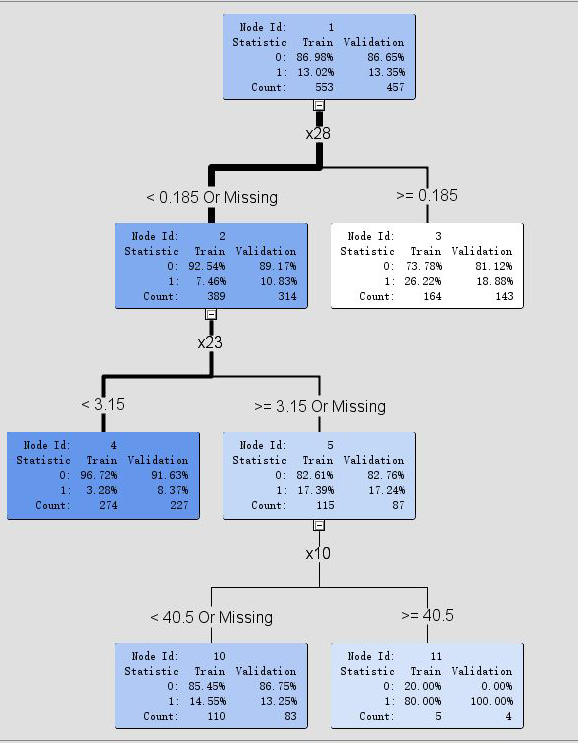


# Part II. Prediction Models, Model Comparison and Champion Model Evaluation

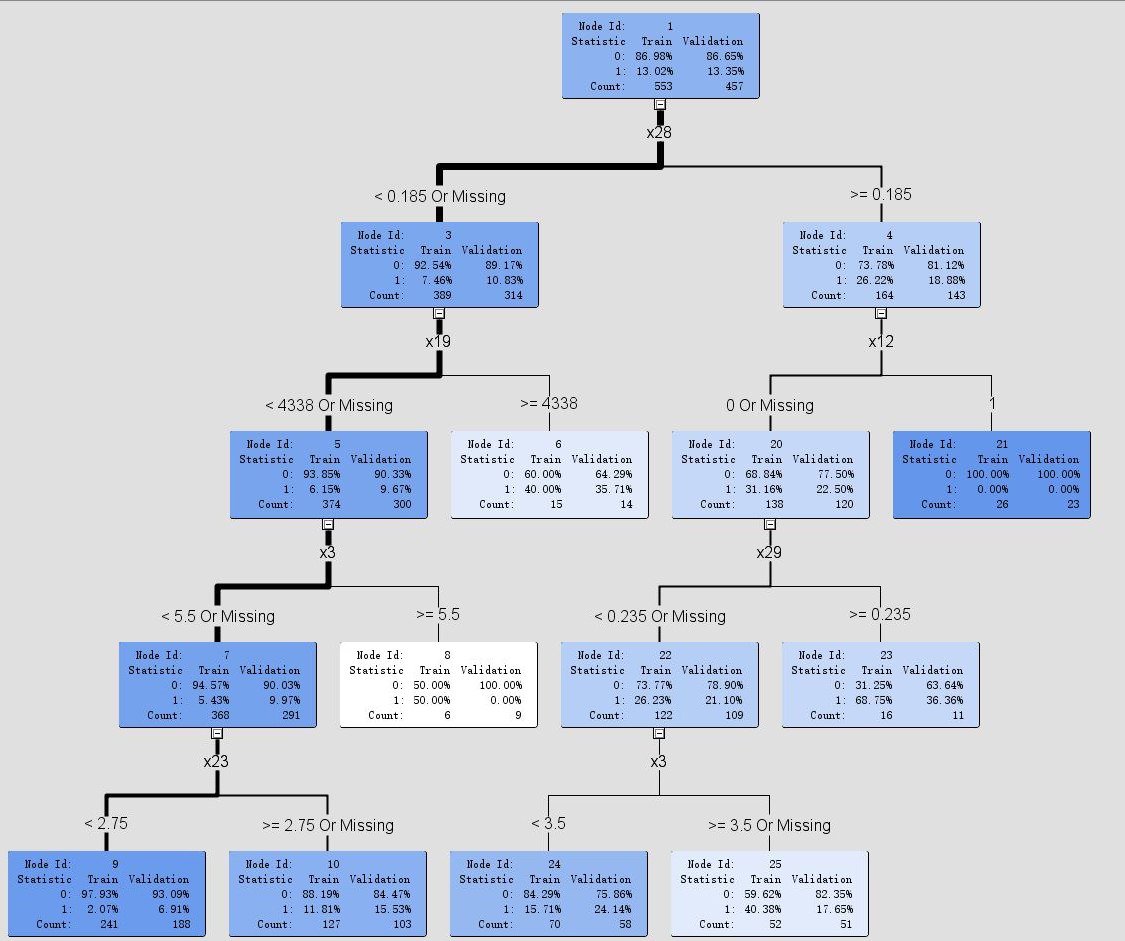


## 1. Prediction Models

### *a. Decision Tree*



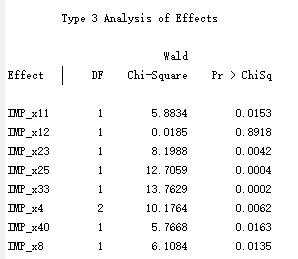
Auto Tree



Interactive Tree

### *b. Regression*

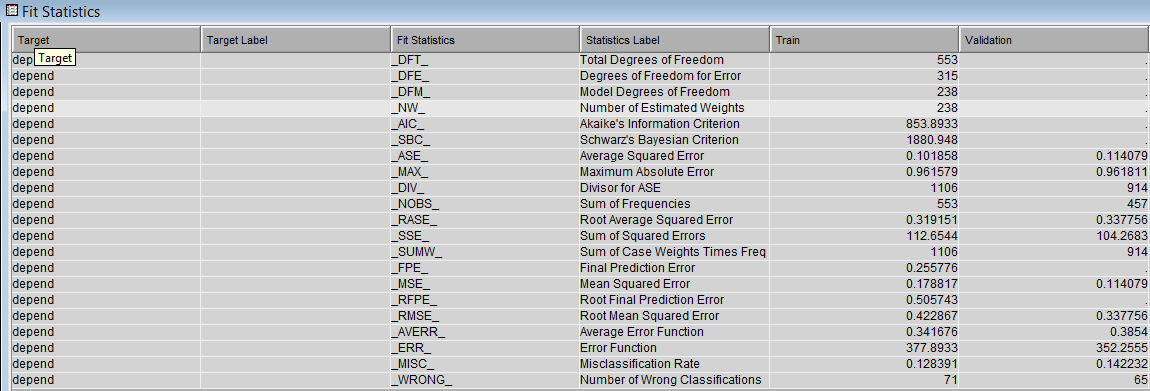
By exploring the whole dataset, we found a plenty of inputs are highly skewed (e.g. x36, x11, x9). Regression and Neural Network models are sensitive to extreme or outlying values in the input space. Inputs with highly skewed or highly kurtotic distributions can be selected over inputs that yield better overall predictions. For these inputs, we use the log transformation to regularize the skewed distributions. By this way, the order of magnitude of the underlying measure predicts the target rather than the measure itself.



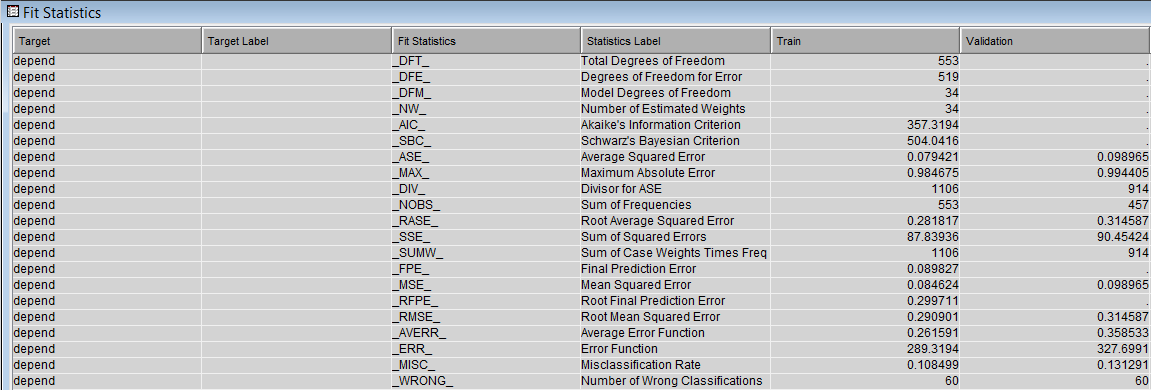
Above is the result of Regression. The selected model, based on the misclassification rate for the validation data, is the model trained in Step 8. It consists of the following effects:

Intercept, IMP\_x11, IMP\_x12, IMP\_x23, IMP\_x25, IMP\_x33, IMP\_x4, IMP\_x40, IMP\_x8

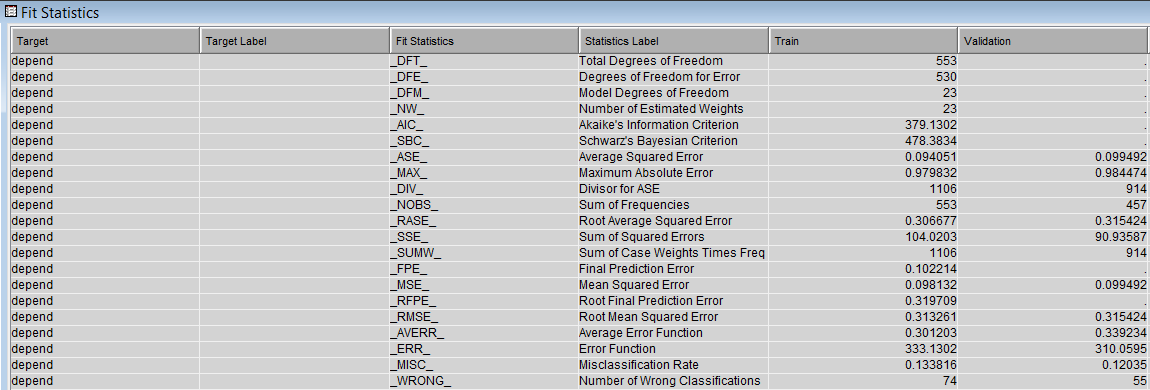
### *c. Neural Network*



Neural Network from Impute Node (238 estimated weights, 14.2% misclassification rate)



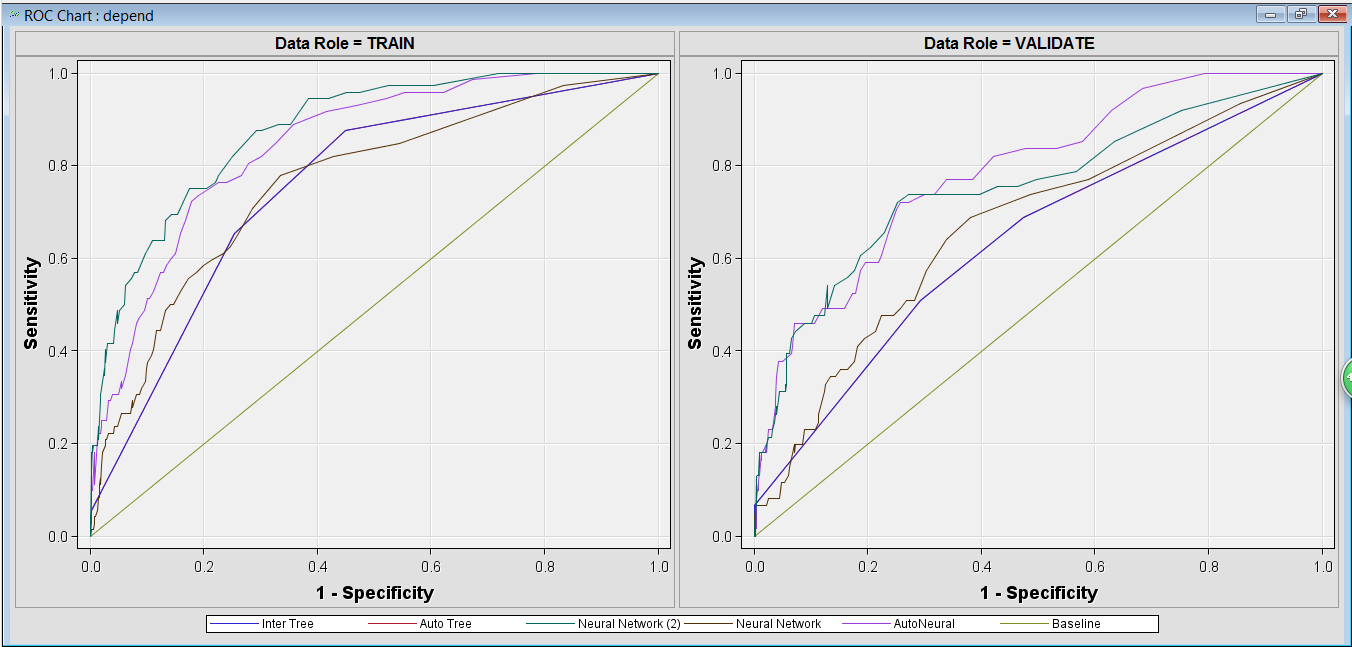
Neural Network from Regression Node (34 estimated weights, 13.1% misclassification rate)

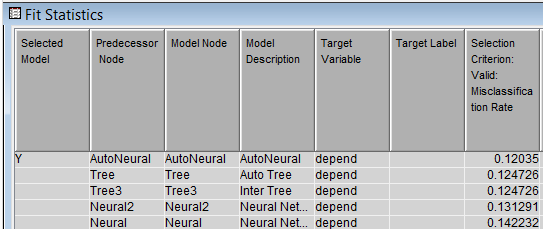


AutoNeural (23 estimated weights, 12% misclassification rate)

From this, we can see by using the variable selected by Regression, we can reduce the estimated weights and misclassification rate.

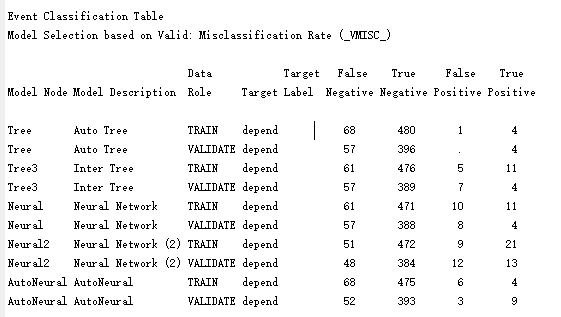
## 2. Model Comparison





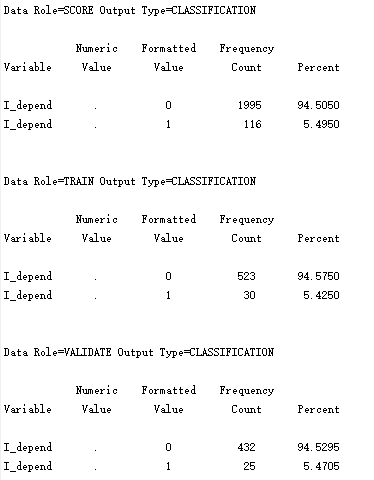
Thus, AutoNeural has the lowest misclassification rate of 12% (Based on 13.2% benchmark).

## 3. Model Evaluation



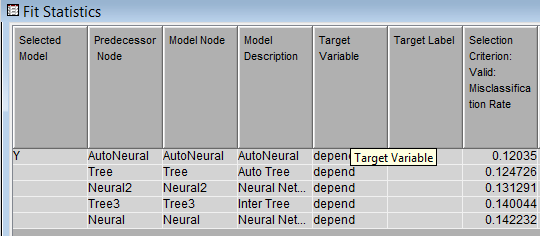
From the output of the Comparison node, we found the Event Classification Table. In this table, we can see the source of error: false positives and false negatives. And as we know, the cost of misclassifying 1 as 0(false negatives) is 5, the cost of misclassifying 0 as 1\*(false positives) is 1, so we should calculate the total cost by 5\*(# of false negatives) plus 1\*(# of false positives) and we can see that neural network 2 has the minimum number of total cost, which is 264 for the train data and 252 for the validation data.

Here is the result of scoring the new evaluation dataset using the champion model.

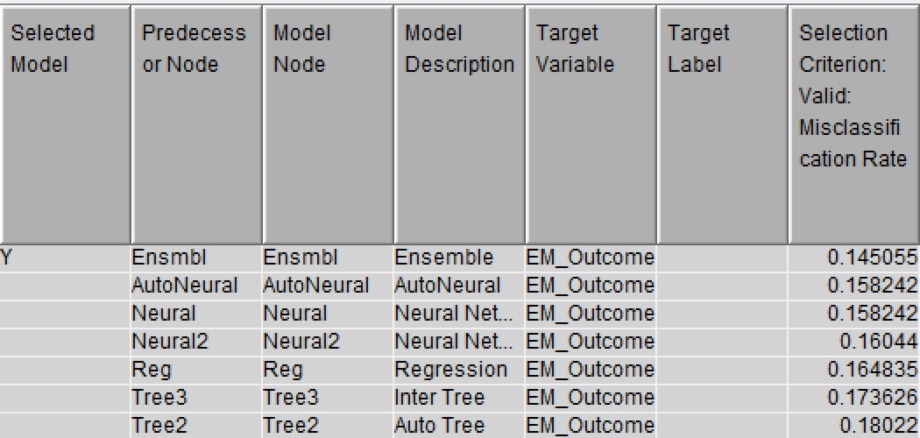


# Part III. Improve Model Performance

By exploring the dataset, we found that among the total 1010 observations, there are 133 records with the target value of 1, which means the benchmark of our final prediction should be approximately 13.2%. However, the misclassification rate of our champion model is 12%, one percent lower than the benchmark. Thus, we tried to further improve our models by implementing several methods from different perspectives. Through some adjustment and manipulation on the data, we improved the misclassification rate of our champion model from 12% (based on benchmark of 13.2%) to 14.5% (based on benchmark of 26.7%). We will demonstrate our adjustment in detail in the later part.



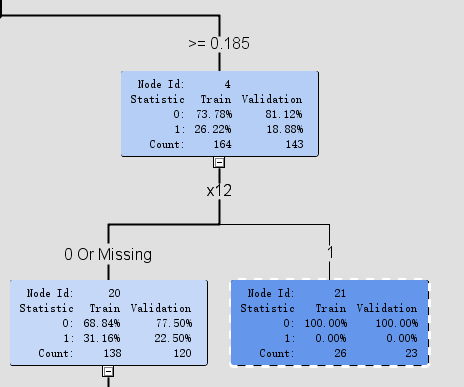
Before Adjustment (Benchmark 13.2%)



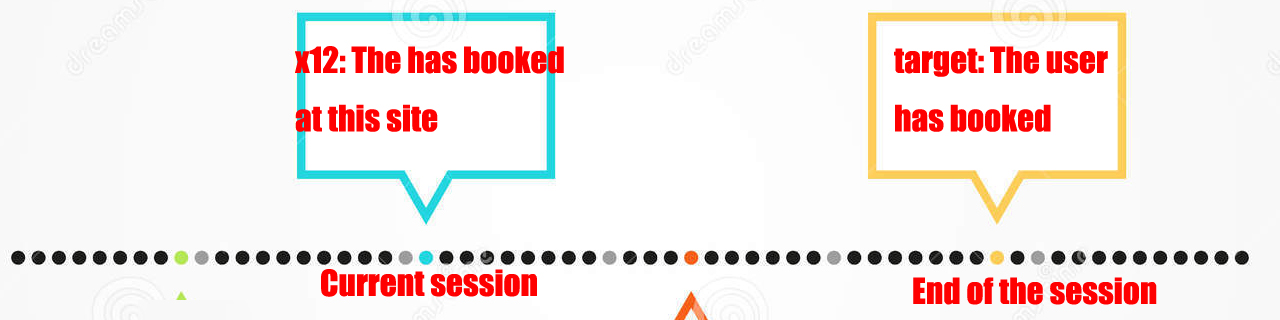
After Adjustment (Benchmark 26.7%)

## 1. Data Modification and Input Selection

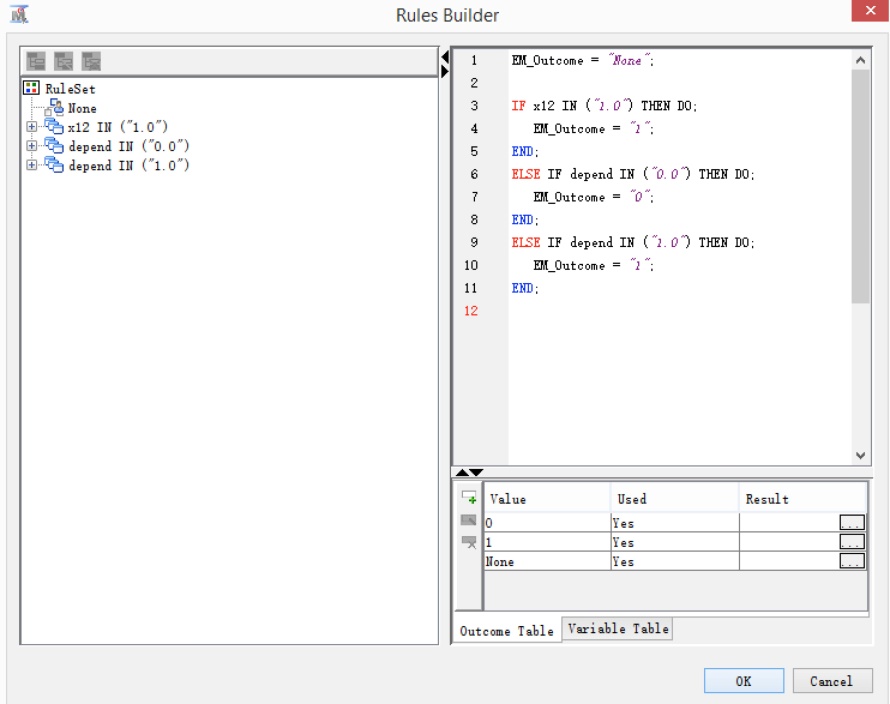
By looking into the interactive tree we made, we found a leaf perfectly classified the target. When x12 is 1, all the value of the target is 0, both for train data and validation data.



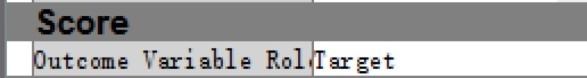
By further looking into the variable description, we found x12 indicates whether the user has booked at this site up to this point in the current session. And our goal is to predict whether the user is going to book in the remainder of the session. This explains why all the target value is 0 when x12 is 1. Therefore, we should exclude this trouble input to eliminate this noise. However, we could not easily delete the entire row because other inputs in the record are useful for our prediction. Also, directly deleting the x12 is not appropriate. Because when x12 is 1, the user has already made a book in this session at this time point, if we do not need to consider x12 as an input, then the target value should be 1 instead of 0.



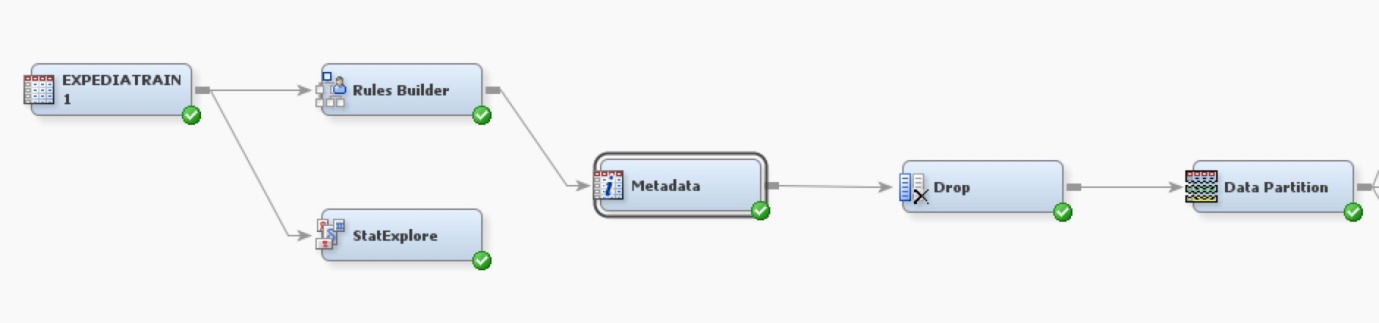
So when x12 is 1, we should first change all the target value from 0 to 1 then drop x12. To achieve this, we could edit the dataset directly in excel, but it is unrealistic if the dataset is very large. We could use SAS code to achieve this, too. Actually, in SAS Enterprise Miner, we could use rule builder node to complete this process and we can easily see how we modified the original dataset as well as edit the modification if we want.



After this step, we got a new input called EM\_Outcome and we should use this new input as our new target. We should note that after this target change, we got 270 records with target value of 1 among all the 1010 observations. Thus, the new benchmark for the modified dataset should be 26.7%.



However, EM\_Outcome is a nominal target and we want to change this to binary so that it would fit all the classifiers used in the next step. Then we use Metadata to change the new target from nominal to binary. And we use Drop node to exclude the former target and x12.



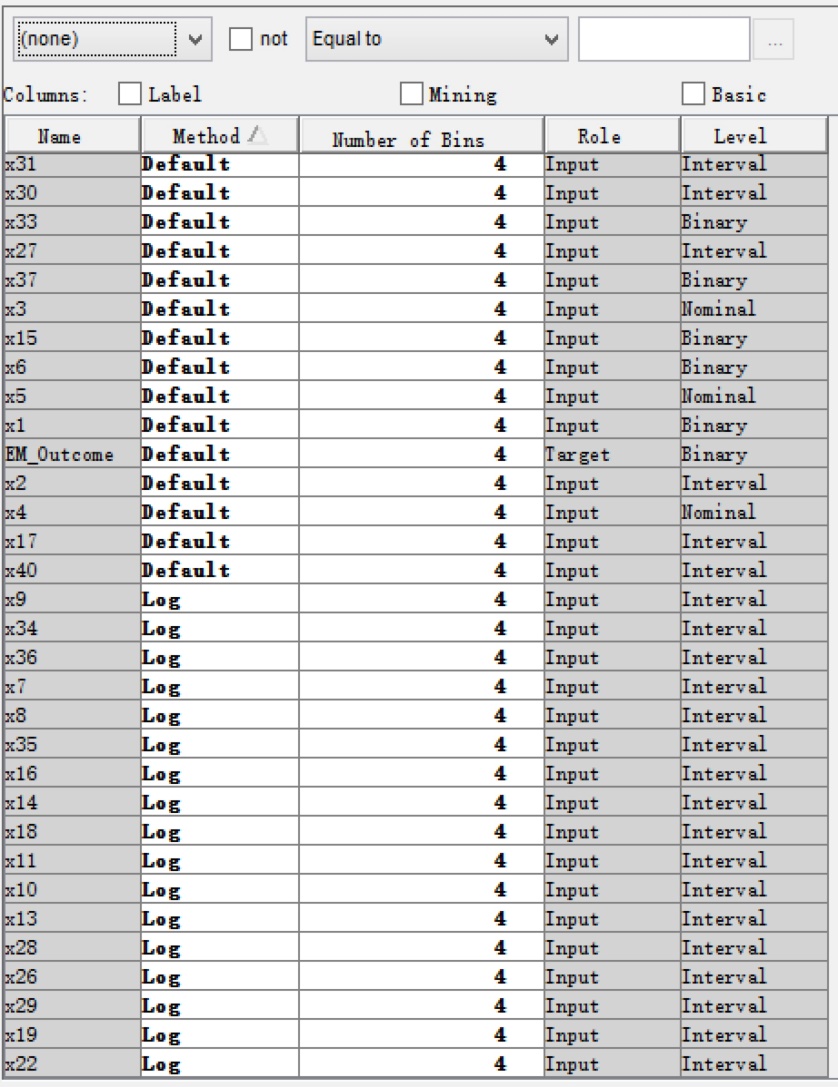
Similarly, x33(whether the user has booked at any sites up to this point in the current session) is also a potential trouble input. From the description of x33, we can esily find x33 should include x12, which means if x33 is 1, then x12 must be 1 and target value must be 0. However, in the data set, we found the values of these 3 variables are contradictory. But we are not sure it is because of inproper variable description or wrong data collection.

## 2. Data Imputation

For the three classifiers we used, only the data for regression and neural network need to be imputed. Because decision tree could categorized the missing value in either side of the leaves, which means there is no difference between the imputed data and the original one.

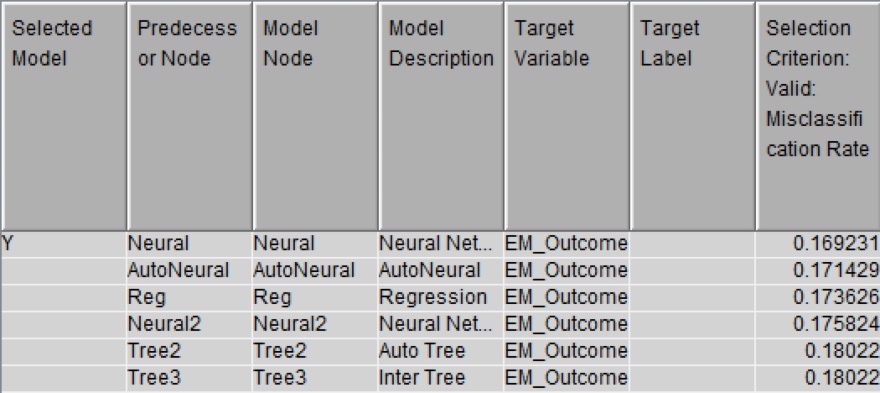
## 3. Transform Variables

By exploring the whole dataset, we found a plenty of inputs are highly skewed (e.g. x36, x11, x9). Regression and Neural Network models are sensitive to extreme or outlying values in the input space. Inputs with highly skewed or highly kurtotic distributions can be selected over inputs that yield better overall predictions. For these inputs, we use the log transformation to regularize the skewed distributions. By this way, the order of magnitude of the underlying measure predicts the target rather than the measure itself.

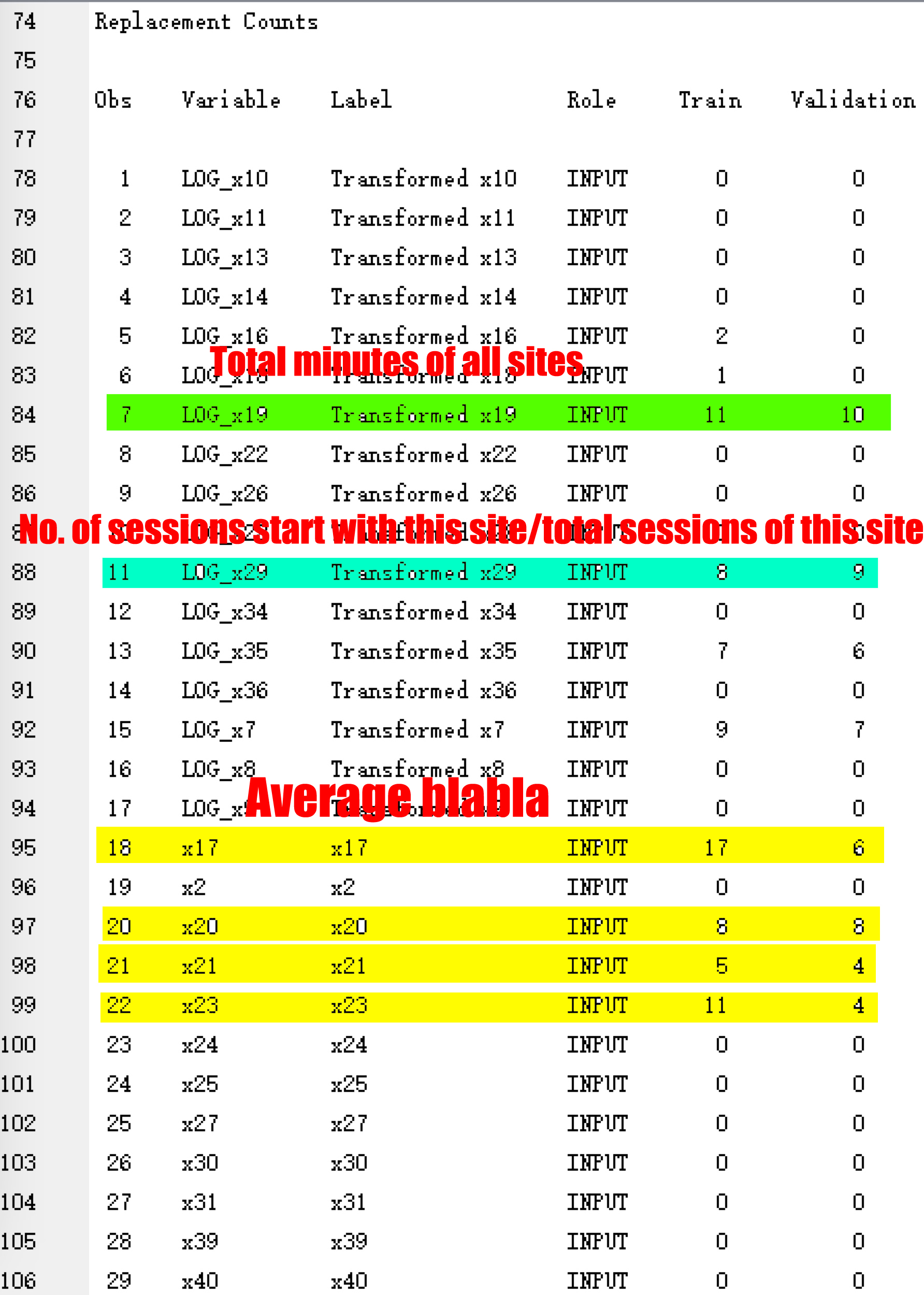


## 4. Data with high variance

We found the attribute values of some inputs have a high variance. According to what we have learned, we could use the replacement node to eliminate some extreme value or outlier. However, after using this node, we found that the performance of all the classifiers did not improve as we expect.

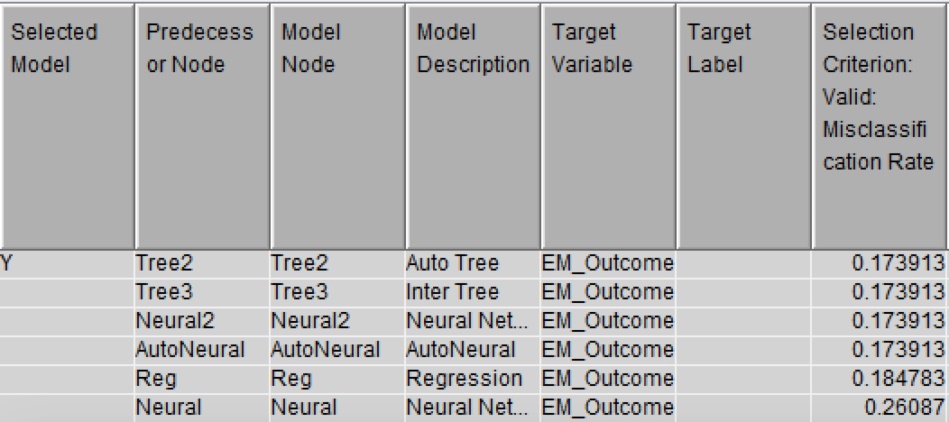


Therefore, we further look into the replacement result. The reason why the model performances are being worse is that most of these inputs have some extent of correlations with each other. For example, we change total 21 observations of x19 (total minutes of all sites) with missing value, and after imputation, these missing values became the average of all the non-missing values. However, this change will make some other inputs invalid because they have some kind of relations. In this situation, the value of x28 (minutes spent at this site divided by total minutes of all sites) will not match with the new value of x19 if we only change x19. Therefore, this kind of replacement would mess up other related inputs. Similarly, like x29 (number of sessions start with this site divided by total sessions of this site), x17 (average sessions per site), all the change we conducted on these inputs will affect the effectiveness of other related inputs. Because of these implicit correlations between different inputs, we cannot rashly replace the outlier with missing value. Thus, we cannot use the replacement node to improve our model performance.

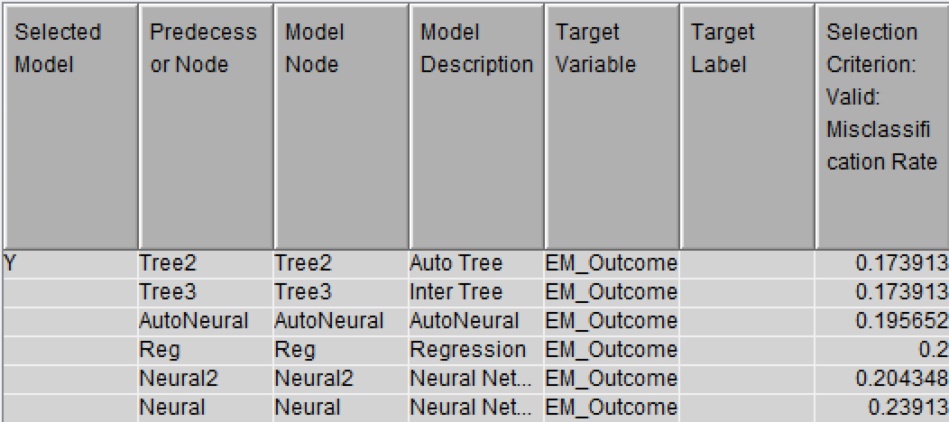


## 5. Data Sampling

There are only 1010 observations in this dataset, we do not think this is a large dataset and there is no need for us to do an extra sampling job. If the dataset is too small, it will lead to overfitting problem. However, we still tried to do sampling and see whether it would help the model performance. Below is the result of sampling for 20% and 50% respectively, we can see the performance get even worse and it proves the conclusion we made before.



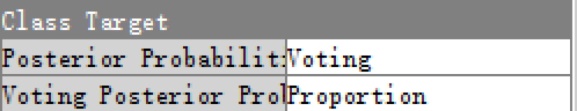
20% Sample



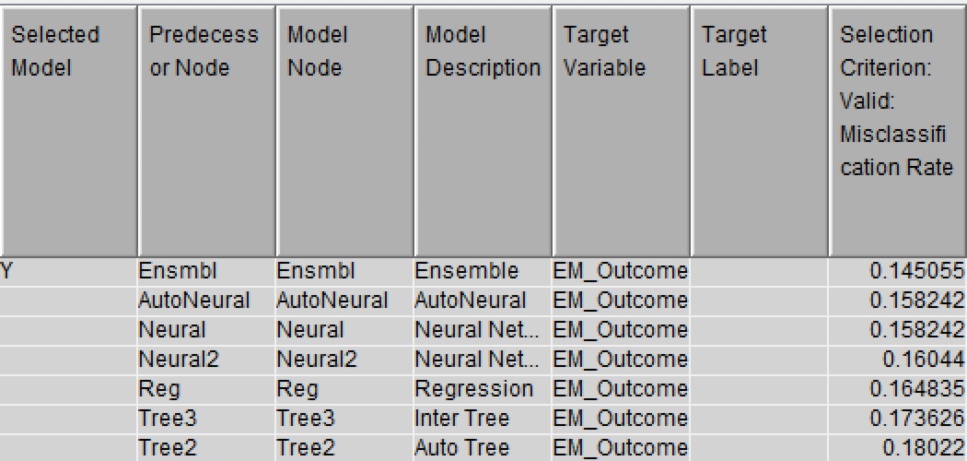
50% Sample

## 6. Ensemble

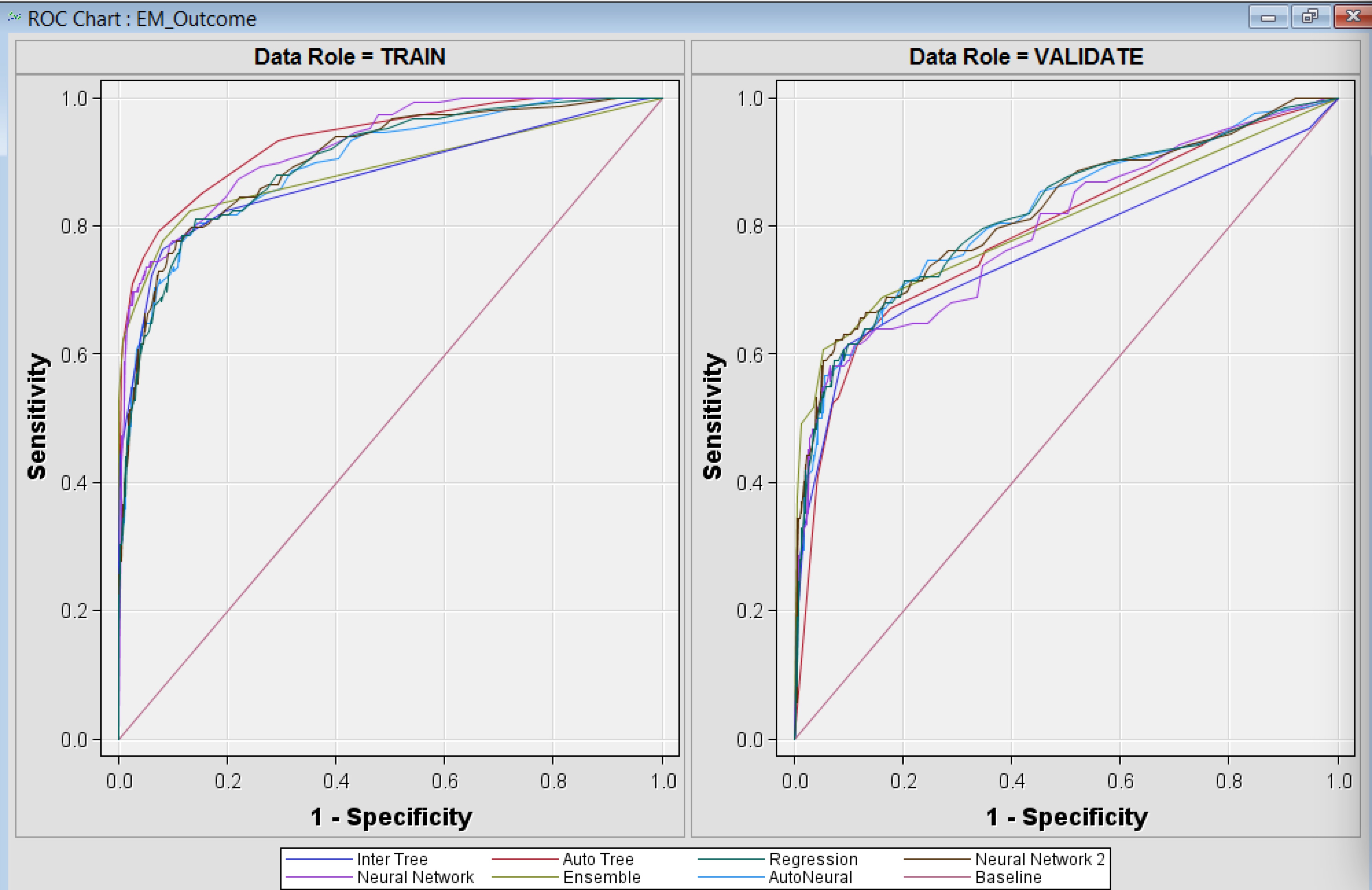
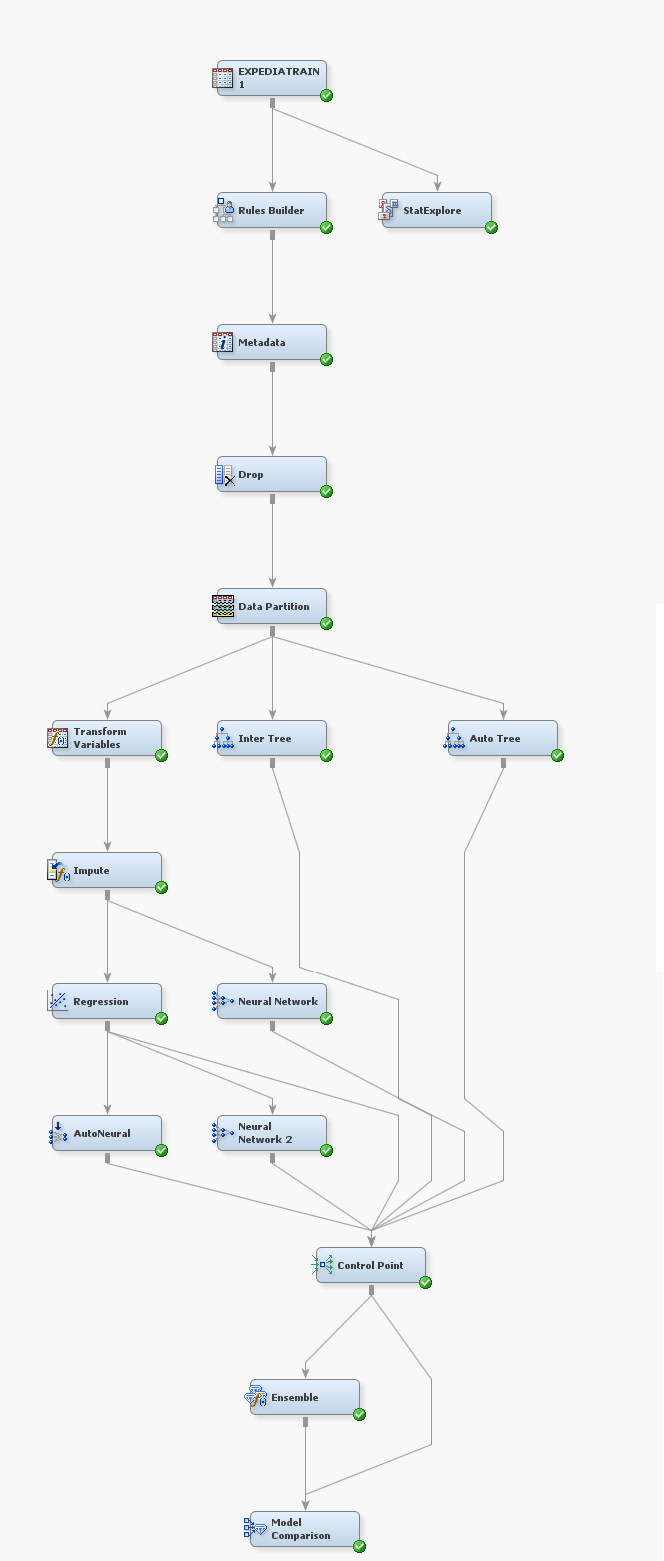
The Ensemble node creates a new model by combining the predictions from multiple models, when the predictions are decisions, this is done by voting. The average method averages the prediction estimates from the models that decide the primary outcome and ignores any model that decides the secondary outcome.

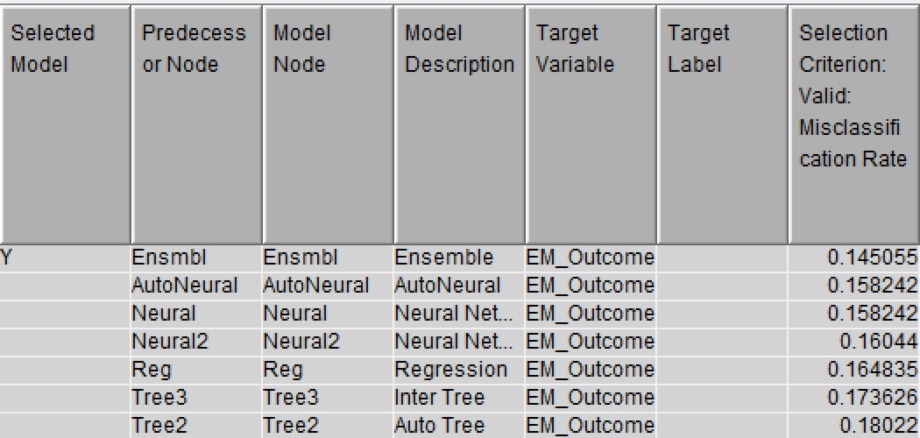


From the result of model comparison, Ensemble became the champion model.



# Part IV. Summary





To sum up, we have learned a lot throughout the whole project.

1. In both data preprocessing stage and model improvement stage, we should look into every variable in the dataset. We should drop the inputs which have correlations with our target. For instance, x12 in this dataset is a trouble maker which we need to drop. However, sometimes it is very difficult to discover the trouble variable directly by looking at the variable itself. We could use our classifier to help us to focus on the suspicious variable, especially when these variables lead to a perfect result; we need to be cautious and strict when using them as inputs. When decided to exclude an input, we found Rule Builder node is a very convenient tool for data modification.
2. When dealing with the inputs with high variance, we need to consider if we made the modification to these inputs, will other related inputs be affected and thus distort the whole dataset. For example, we cannot just use missing value to replace some outliers in some special inputs but keep everything else unchanged. In this project, if we changed the variables representing total minutes, average sessions and percentages, we need to adjust other related inputs correspondingly. Otherwise the whole dataset would be distorted and the classifier performances would get worse.
3. As for the skewed data, we need to think about whether these statistical properties will affect our model performance in what extent. We should take some methods to optimize our dataset so that it could be fit properly for our classifiers.
4. In the classifier adjustment, we also learned some tricks through this project. For interactive tree, we do not need to select the inputs with the highest logworth, but we need to consider the variance between training data and validation data at the same time. For the neutral network, we could consider AutoNeural when the Neural Network tool did not perform well. Because the neural network models that you obtain with the AutoNeural and Neural Network tools are different, even if both networks have the same number of hidden units. When selecting inputs for neural network, we could use the regression node to select the variables.