MIS 6334.002 advanced Business Analytics

Group 3

Chiranjeeva Dayananda

Mohamed Salihdeen Rafiqdeen

Surendar Rajavijayan Sundar

Vigneshwaran Kannan

**Table of contents**

[**Executive Summary:** 2](#_Toc463270252)

[**Basic Data Processing:** 2](#_Toc463270253)

[**Stat Explore:** 3](#_Toc463270254)

[**Impute:** 4](#_Toc463270255)

[**Building Decision Tree:** 4](#_Toc463270256)

[**Decision Tree:** 4](#_Toc463270257)

[**Interactive Decision Tree:** 5](#_Toc463270258)

[**Building Neural Networks and a Regression model:** 5](#_Toc463270259)

[**Assumptions:** 5](#_Toc463270260)

[**Logistic Regression:** 5](#_Toc463270261)

[**Neural Networks:** 5](#_Toc463270262)

[**Model comparison and model selection:** 6](#_Toc463270263)

[**Improve your model performance:** 7](#_Toc463270264)

[**Sampling:** 7](#_Toc463270265)

[**Skewness and Variable selection:** 7](#_Toc463270266)

[**Impute:** 8](#_Toc463270267)

[**Principal components:** 9](#_Toc463270268)

[**Bagging and Boosting:** 9](#_Toc463270269)

[**Ensemble:** 9](#_Toc463270270)

[**Summary:** 10](#_Toc463270271)

[**Learnings from the project:** 12](#_Toc463270272)

# **Executive Summary:**

The objective of the project was to build a prediction models to determine probability of a user completing the booking process for the dataset based on Expedia. We explored the dataset thoroughly and carried out various pre-processing techniques. After this, we trained and validated the dataset using various prediction models to identify the best fit for the dataset. Once we identified our best fit we tried to improve the model performance by applying concepts learned from the class. In this report we have included our analysis and findings we learned from the dataset.

# **Basic Data Processing:**

To understand the data better we used Variable selection node to identify variables with most influence on target. Based on its results we identified 20 variables using minimum r square stats. We also observed an intriguing trend in the dataset, target value is always “0” when the value x12 is “1”. Upon further investigation we found the x12 indicates whether a user already made a booking in the site during the current session. If the user has already made a booking there is a high chance that user won’t be making another booking during the current session. Due to anomalous nature of the variable we decided to merge it with the target and drop it from our analysis. (Since x12=1 means user has already made a purchase decision with the site.) We used SAS code to change the target value from 0 to 1 whenever x12=1.

**data** tmp1.projdataset2;

set tmp1.projdataset;

if x12=**1** then target = **1**;

else target=depend;

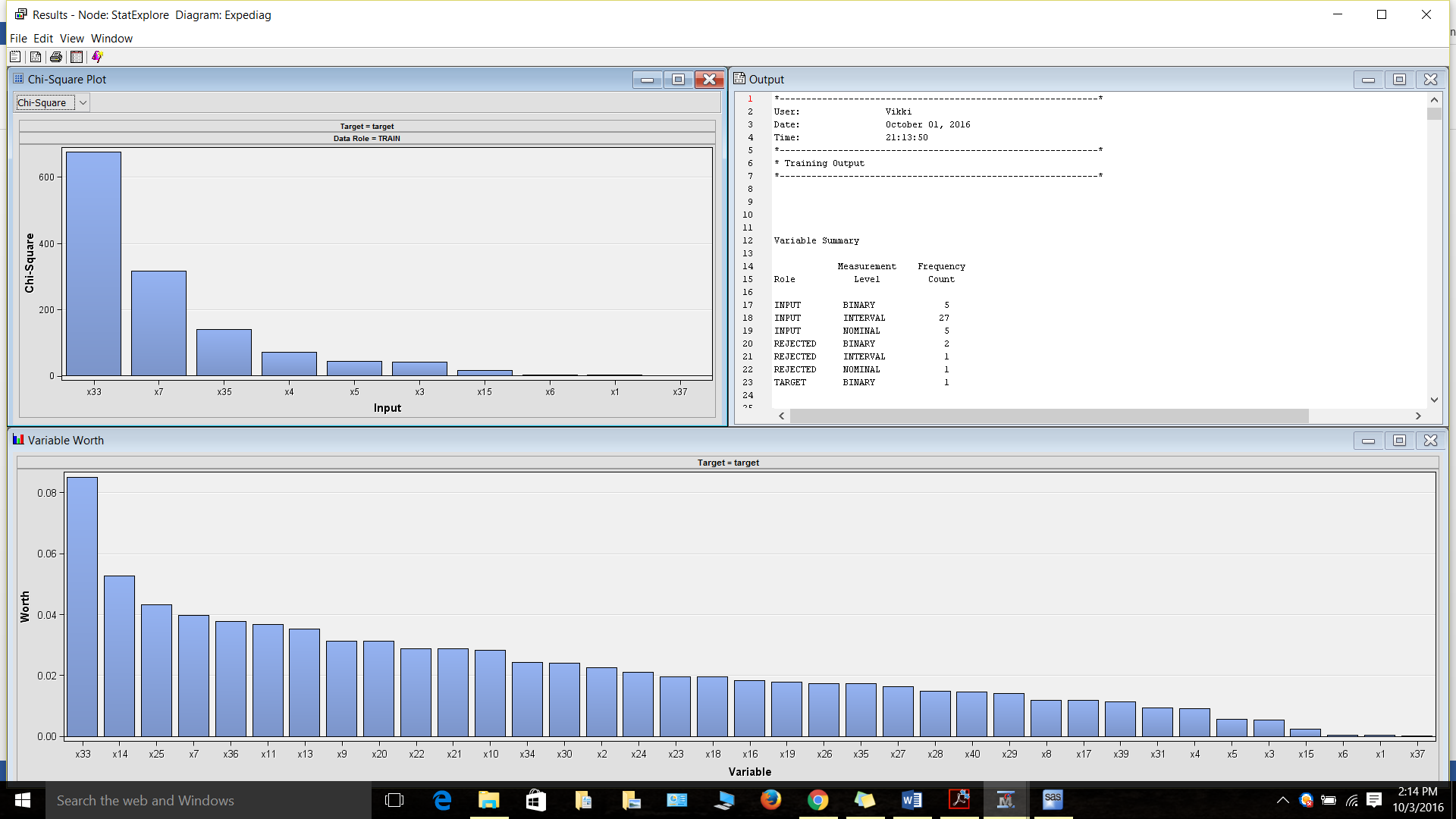
**run**;

Next step in preprocessing includes using of Transform Variable node to transform values, whose skew value falls outside -1 to +1 range. This is accomplished using log function.

## **Stat Explore:**

The dataset contains 41 variables including the target and to explore the data we used stat explore node by connecting to the data source. We observed the following:

1. All the variables have missing values.
2. We also observed few variables have skewed distribution.
3. We found out x33 has highest variable worth followed by x14 and x25.
4. The *Chi-square* plot orders the top 10 variables by their chi-square statistics.



## **Impute:**

Regression and Neural networks ignore records with missing values. So in order build solid prediction model the missing values should be imputed. We set the cut off limit for imputation as 50% (which rejects variables with more than 50% missing records), which rejects x32 and x38. Count is the default impute method for class variables whereas median is the default impute method for interval variables.

# **Building Decision Tree:**

## **Decision Tree:**

Once the preprocessing and imputation is completed, we split the data into training and validation dataset using data partition node. We feed output of the data partition to decision tree node to perform multi-way splitting of the dataset for analysis. We ran the decision tree using SAS Enterprise Miner’s default setting, SAS automatically ranks variables based on their contribution to the tree. This can be helpful in future analysis. Few of the important property of the decision tree are given below:

1. *Maximum Depth Splitting:* This was set to 6.
2. *Leaf Size:* This node was set to 5.
3. *Number of surrogate:* This node property was set to 5.

**Misclassification rate:** *0.15444*

## **Interactive Decision Tree:**

Next step we tried to create interactive decision tree. In this, the tree is made interactive by splitting the tree node at each step based on the highest log (p) values. We split the initial dataset into 2 nodes using x25, further we continued to split the tree nodes using x11, x9 and x4. After this we ran the interactive decision tree using SAS Enterprise Miner default settings.

**Misclassification rate:** *0.1985*

# **Building Neural Networks and a Regression model:**

## **Assumptions:**

Neural Networks and regression model doesn’t work well with data’s with missing values and values with skewness outside -1 to +1 range. During data pre-processing step we dealt with missing values using impute node. Next in order to deal with records with skewed values we used Log 10 from transform variable node in order to improve the overall fit of the model

## **Logistic Regression:**

We paired the regression node with output of transform variables node. For regression, we selected Stepwise as the selection model which begins logistic regression with the variables, which exerts influence over the target, until the significance level or stop criteria is met. We analyzed its results such as odds ratio, lift ratio with respect to training and validation to ensure the reliability of the model.

**Misclassification rate:** *0.1686*

## **Neural Networks:**

We used neural network to identify variety of non-linear relationship between the predictors compared to logistic regression. For neural network we used the following set of parameters:

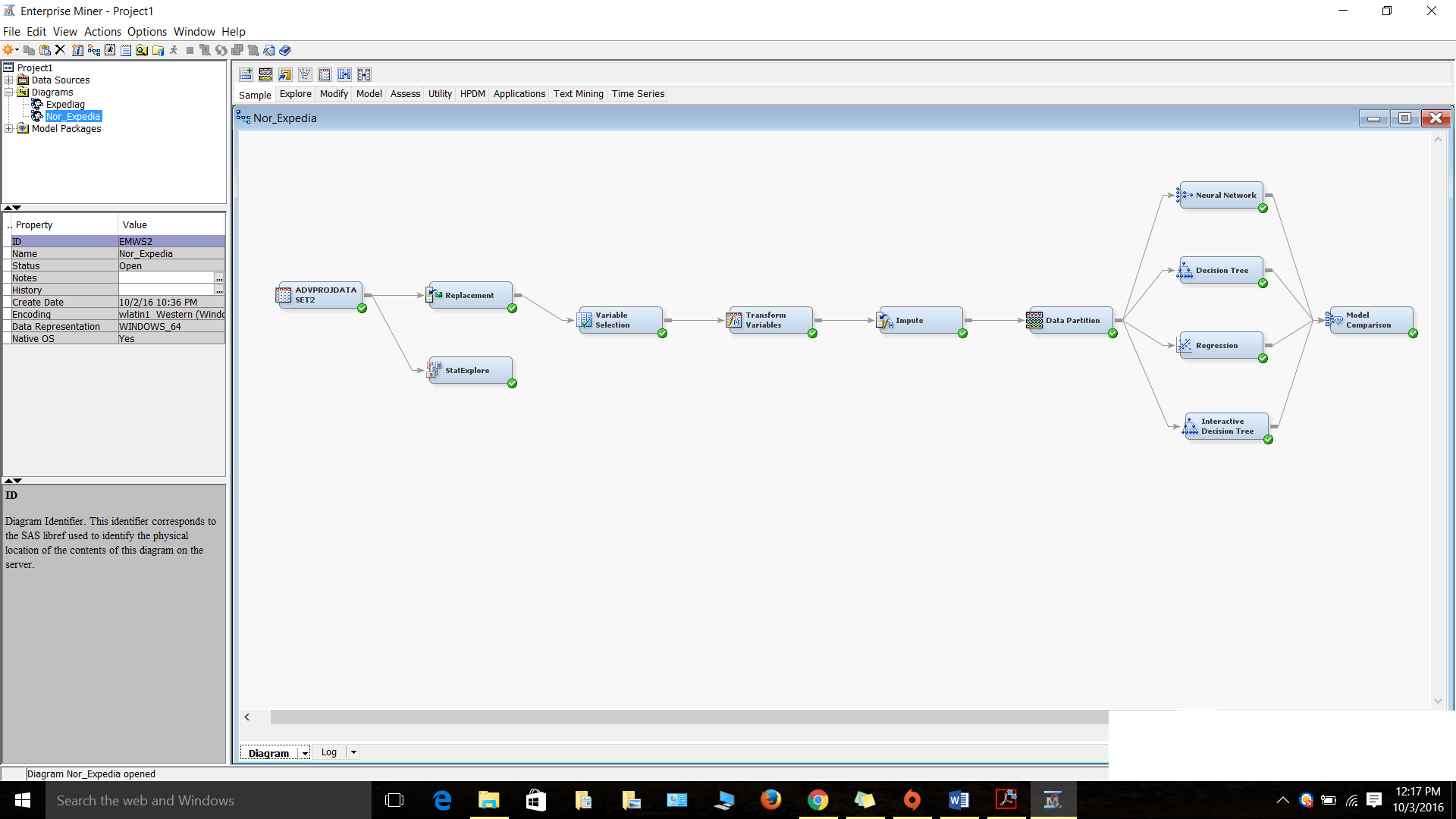
1. *Hidden units property:* Yes
2. *Standardization property:* Yes
3. *Direct connection:* Yes
4. *Number of hidden units:* 26

We ran the neural network node using the above property and took note of misclassification rate of training and validation dataset.

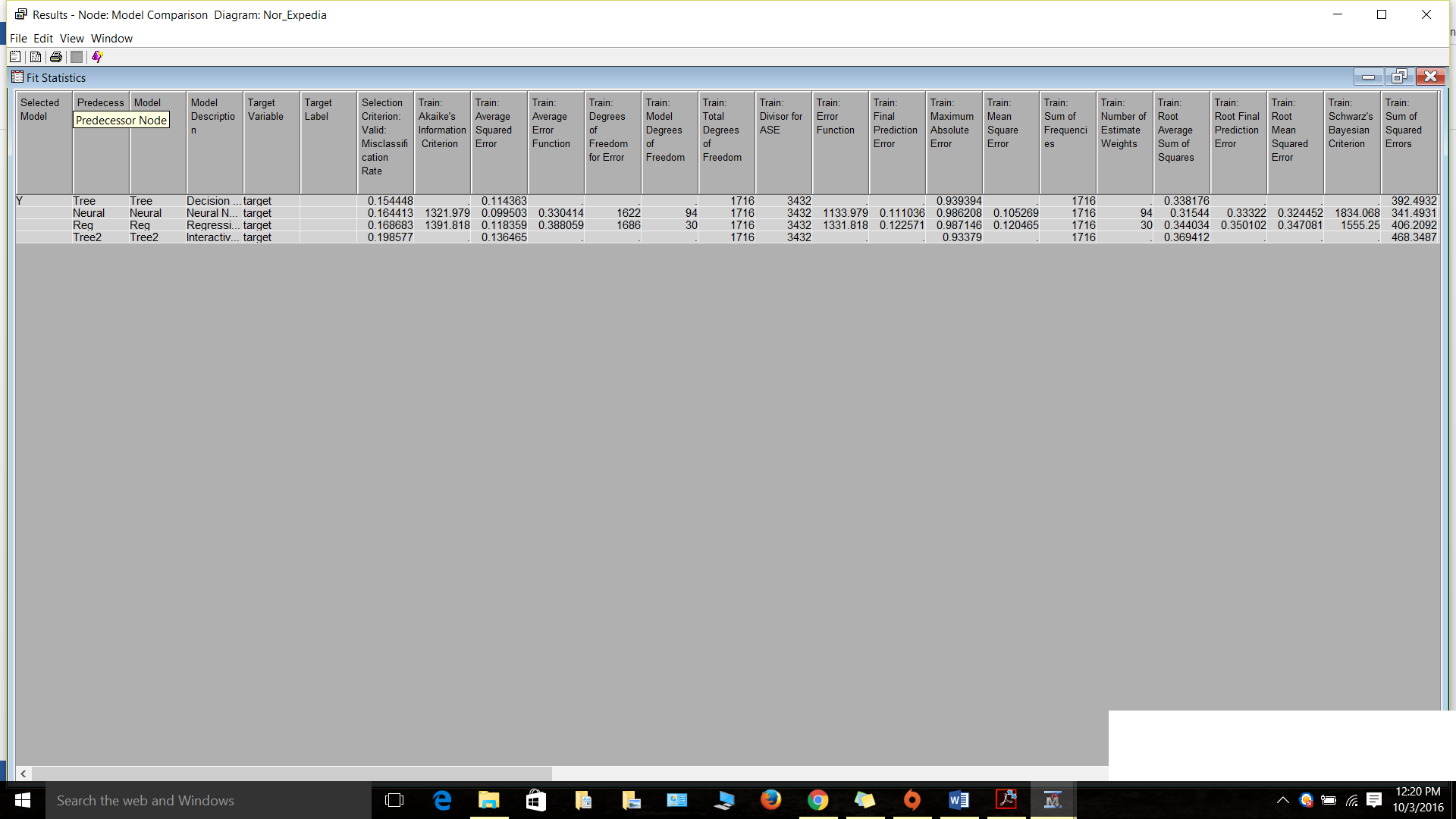
**Misclassification rate:** *0.1644*

# **Model comparison and model selection:**

So far we have analyzed the dataset using major prediction models to determine whether a user will complete his/her booking during the current session. We also created cost based matrix as per instructions while linking the data source with prediction models. Now to choose best prediction model among the candidates, we used model comparison node to compare these models built so far.



Once we execute our above diagram, the fit statistics from the model comparison node gives best fit from these 4 models, which is Decision Tree.



# **Improve your model performance:**

As mentioned in the above part, decision tree gave better misclassification rate compared to other models. When we explored more on the impute node, we changed the default input method to tree surrogate for class variables and Mean for interval variables. Now, we had better misclassification rate for neural network (0.12669), hence we chose that as champion model. Minimum cutoff was changed to 25 instead of 50 but there was no change in the rate.

## **Sampling:**

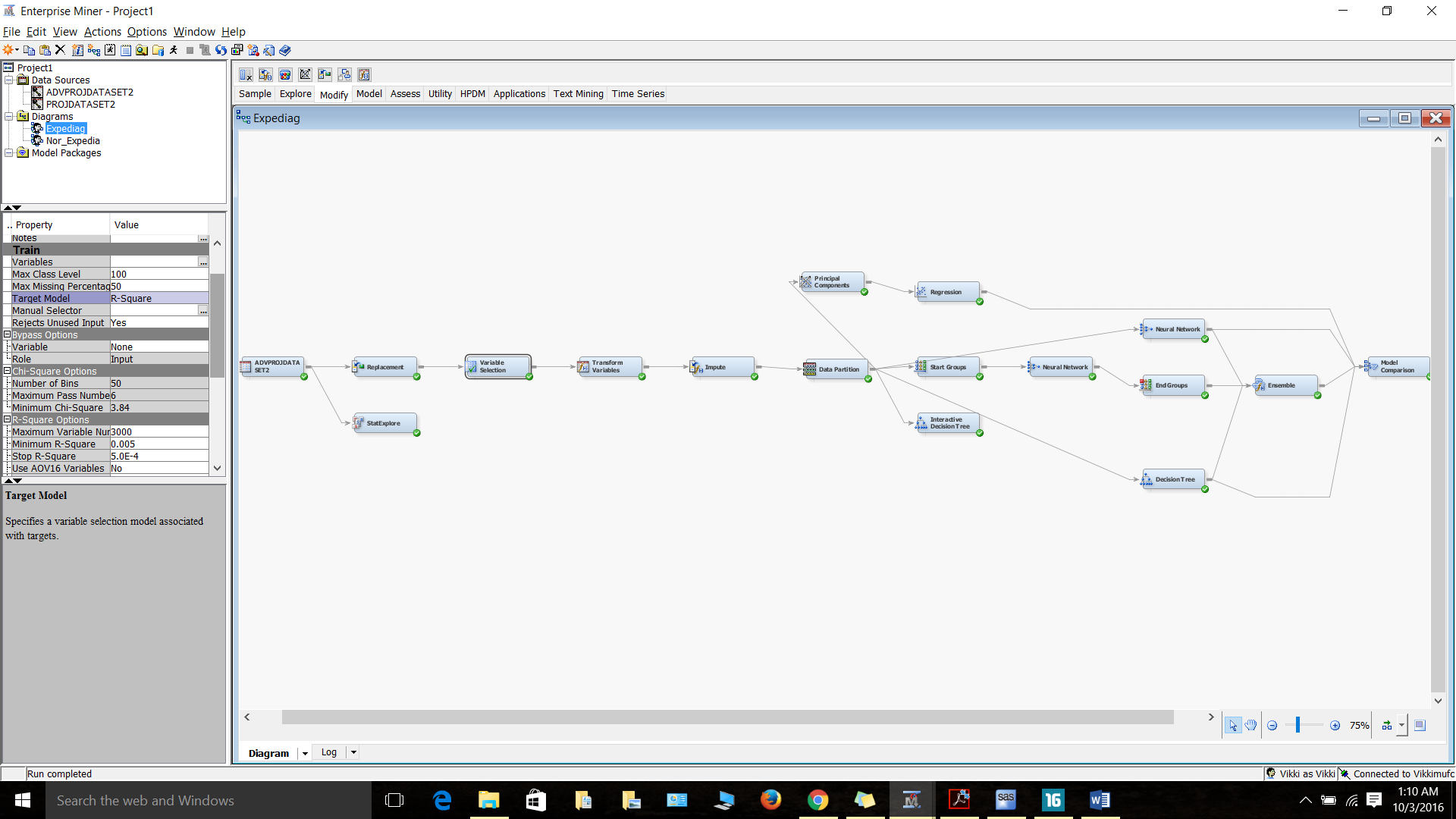
We tried sampling the data using sample node by using 75% of the data but did not improve the performance of any of the model instead the rate rose up to 0.26 and deteriorated our performance. Hence, we concluded sampling the data cannot fetch us the results which we intended for.

## **Skewness and Variable selection:**

All the variables cannot contribute in predicting the target and hence variable selection node was used in order to find the top variables which contributed in predicting the target. We chose the below options:

1. *Target model:* R-square
2. *Minimum R square:* 0.005

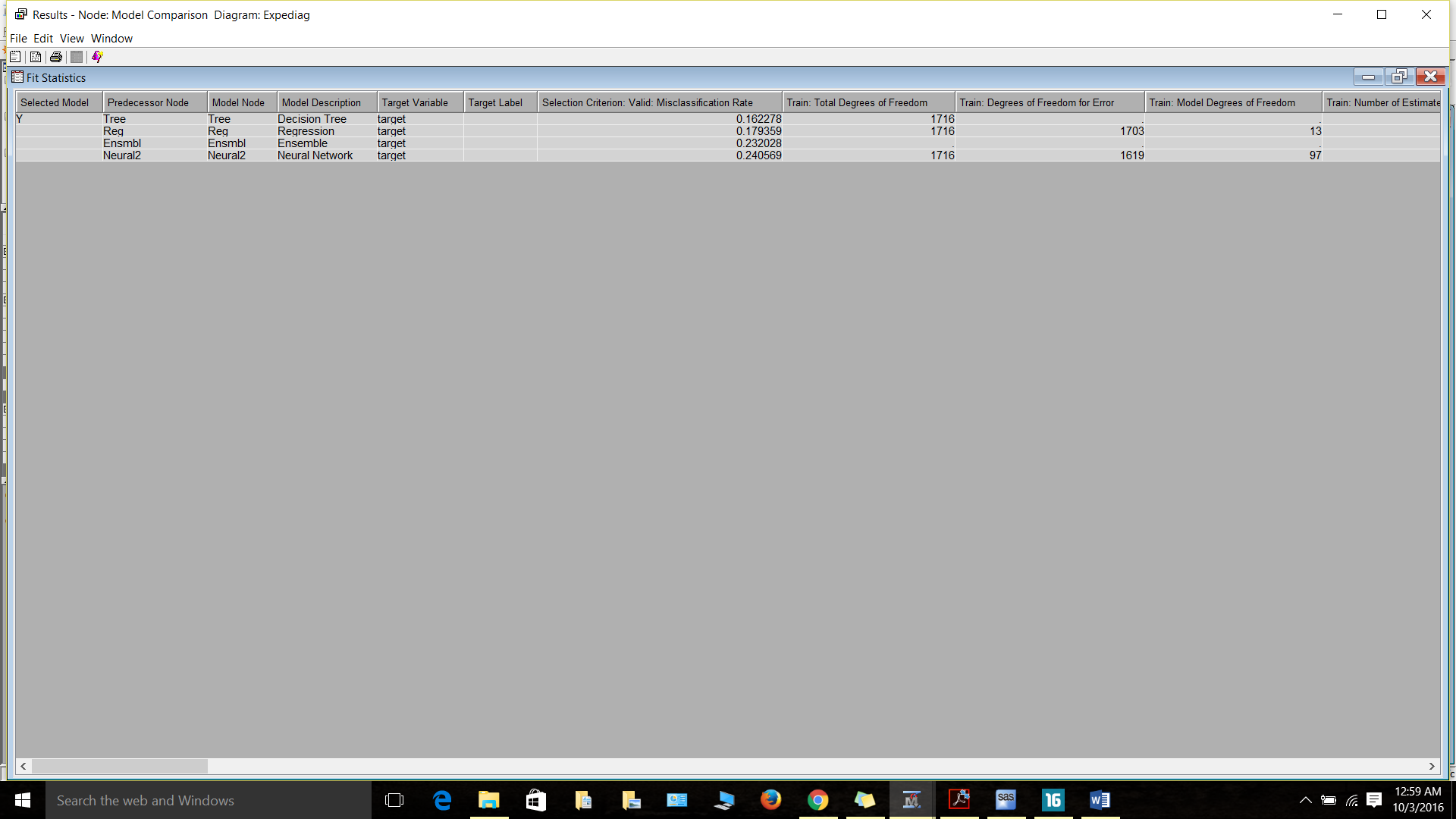
When we tried to change the minimum R square value, it increased the misclassification rate and hence we opted for 0.005.



Variance is directly proportional to skewness hence data with high variance has to be handled. We used Stat Explore node to check the skewness and variance of the variables. Variables which has skewness more than 1 and less than -1 was chosen and log 10 method was used to remove skewness. This output was given as input to impute node for further classification.

## **Impute:**

Without use of impute node, the results were as given below. The misclassification rate of the validation data increased and hence affected the model performance.



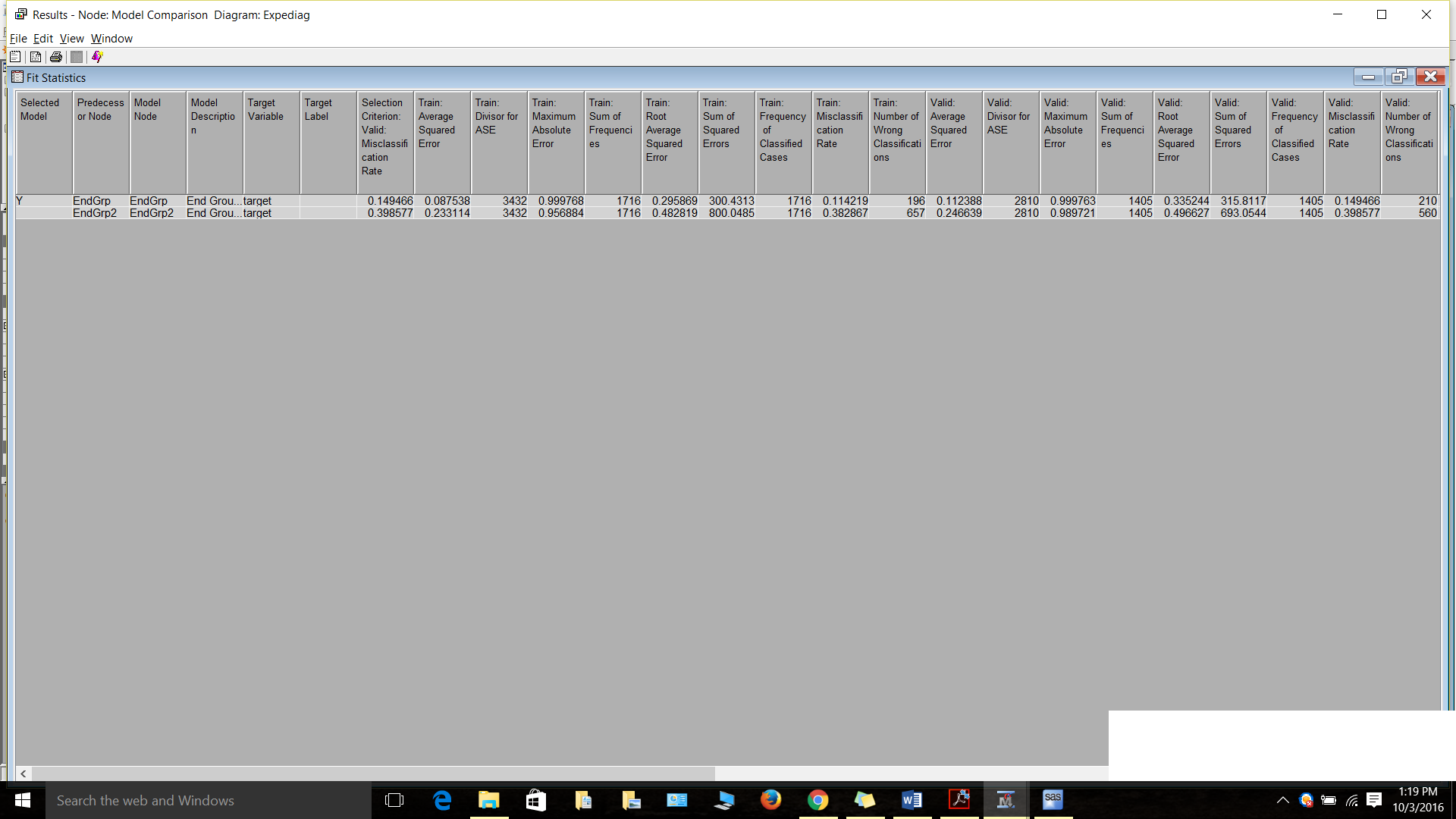
There were 18 variables chosen in variable selection node as input and all the 18 variables were used in imputation. Impute node was used in all the classifiers as there was significant contribution from the node.

## **Principal components:**

We used principal components node preceding the Regression model to see if there are any significant contribution to the resulting misclassification rate, but there was no impact in the rate. Hence we did not consider using this node.

## **Bagging and Boosting:**

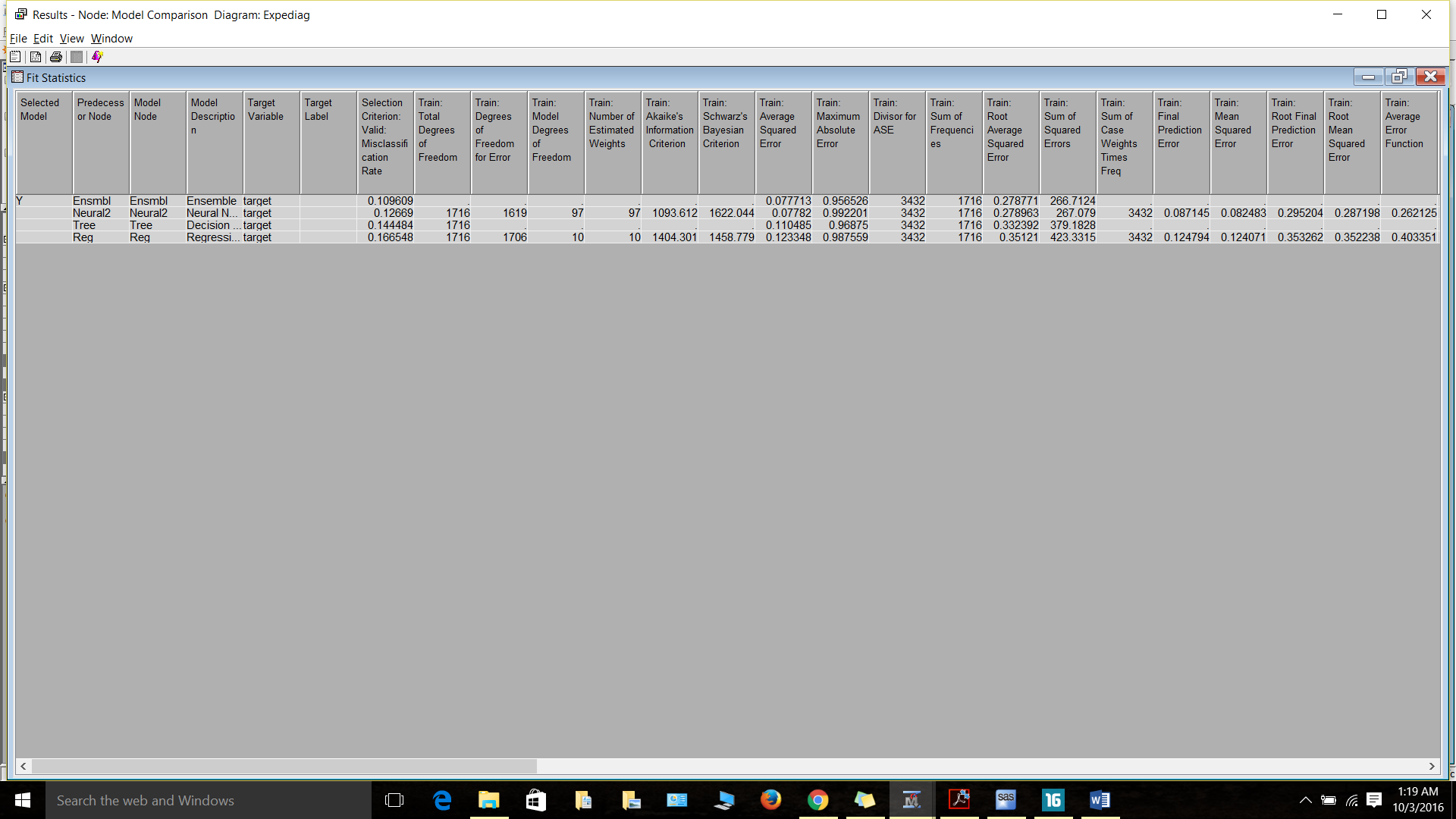
In order to fine tune the model, we created one more neural network trying bagging and boosting with it. This shows that boosting does not fit our model. Bagging with index count of 5 and percentage of 80 gave 0.1494 as misclassification rate and boosting with index count of 7 and percentage of 80 gave 0.3985. Boosting gave misclassification rate ranging between 0.26 and 0.44 with different index count and hence rejected it completely.



## **Ensemble:**

Ensemble node uses the output of the model results as input and predict the target variable. Neural network, neural network with bagging and decision tree is given as input and we ran the ensemble node. The resulting misclassification rate was 0.109 which was considered a good rate.

Misclassification rates of different models are given below in the fit statistics window:



This gives us the best misclassification rate and hence ensemble was chosen the better model compared with other models.

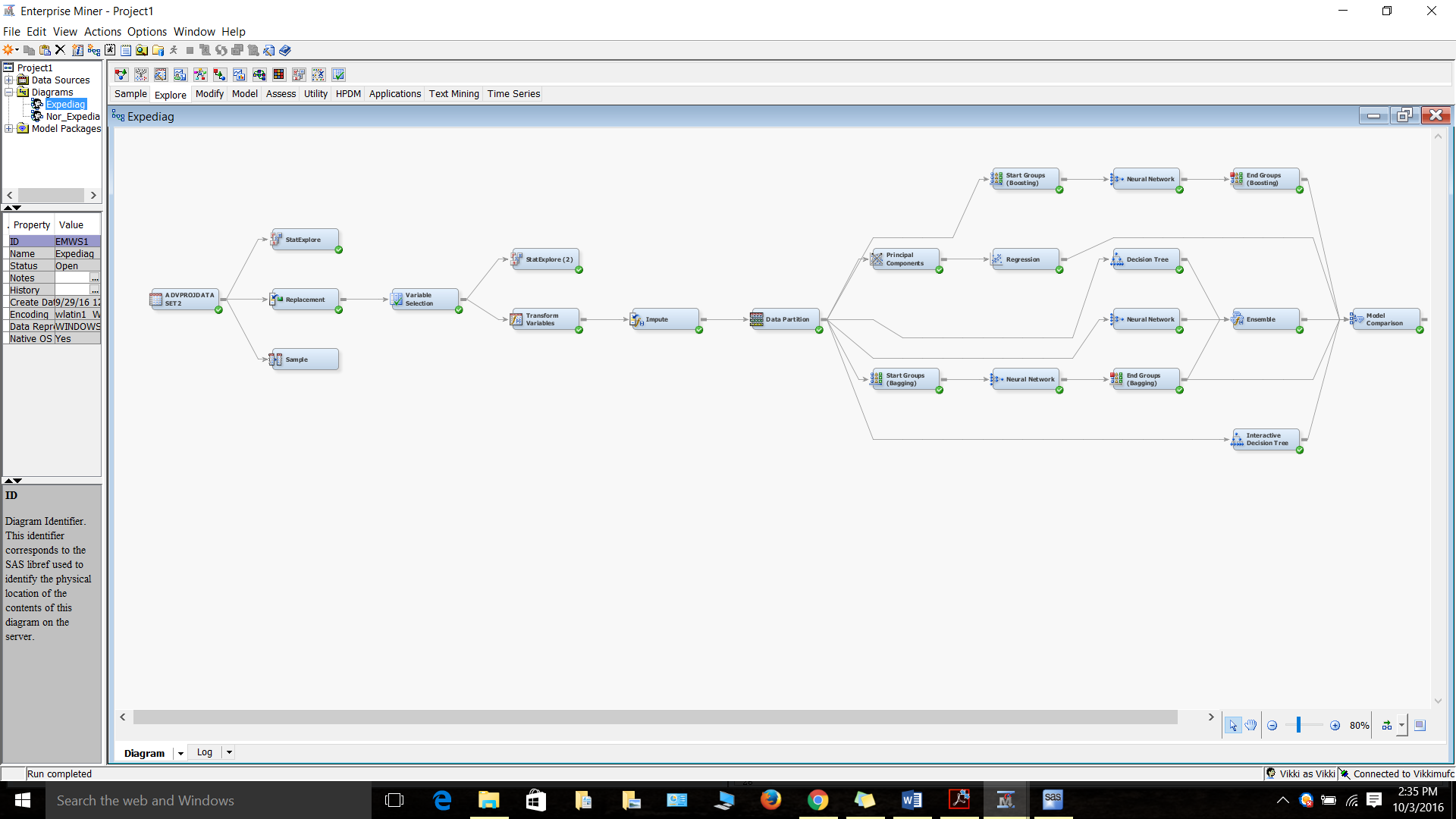
# **Summary:**

We used 7 different models to predict the target and they are listed below:

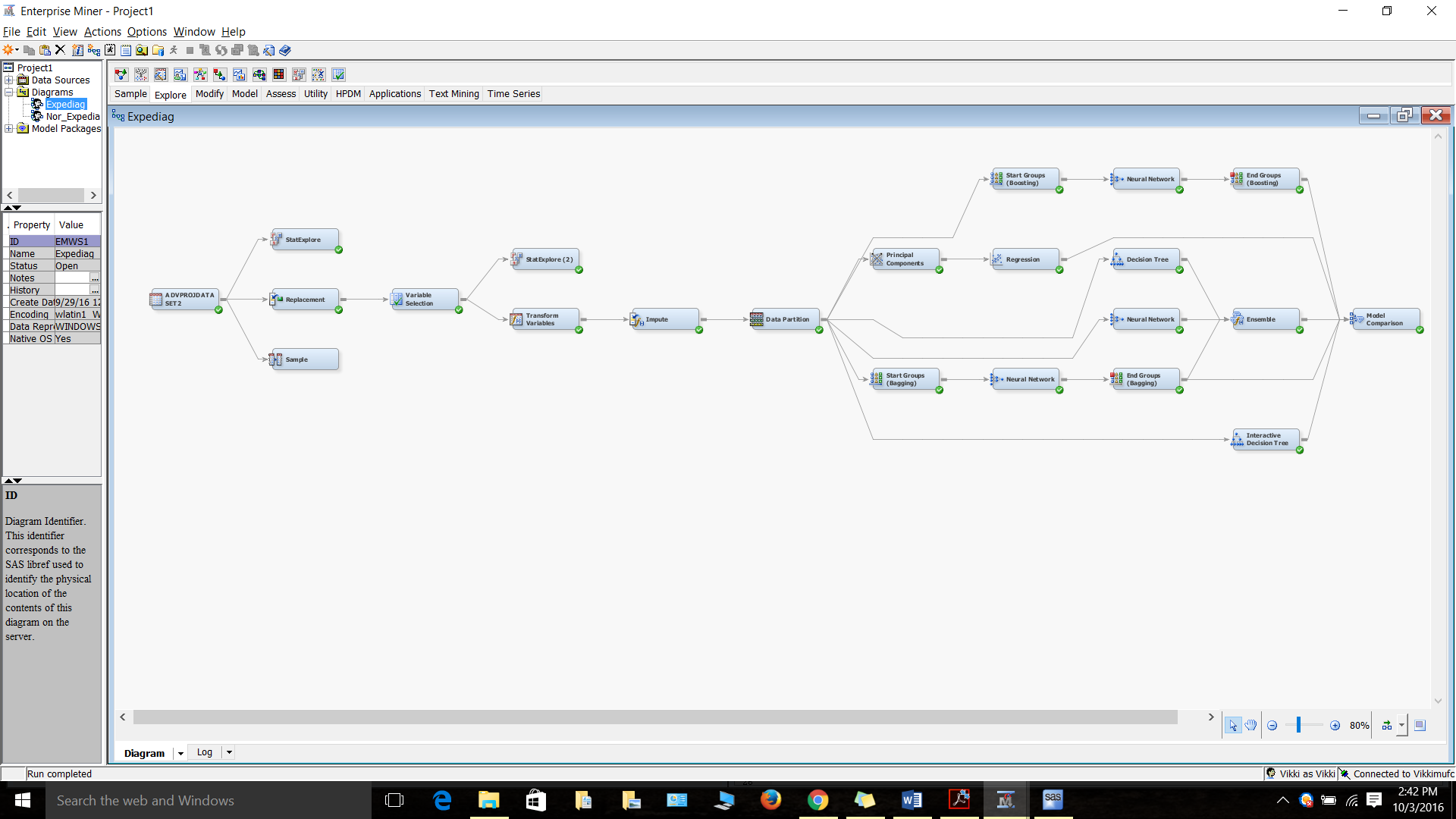
* Decision Tree
* Regression
* Neural Networks
* Neural Networks with Bagging
* Neural Networks with Boosting
* Interactive Decision Tree
* Ensemble

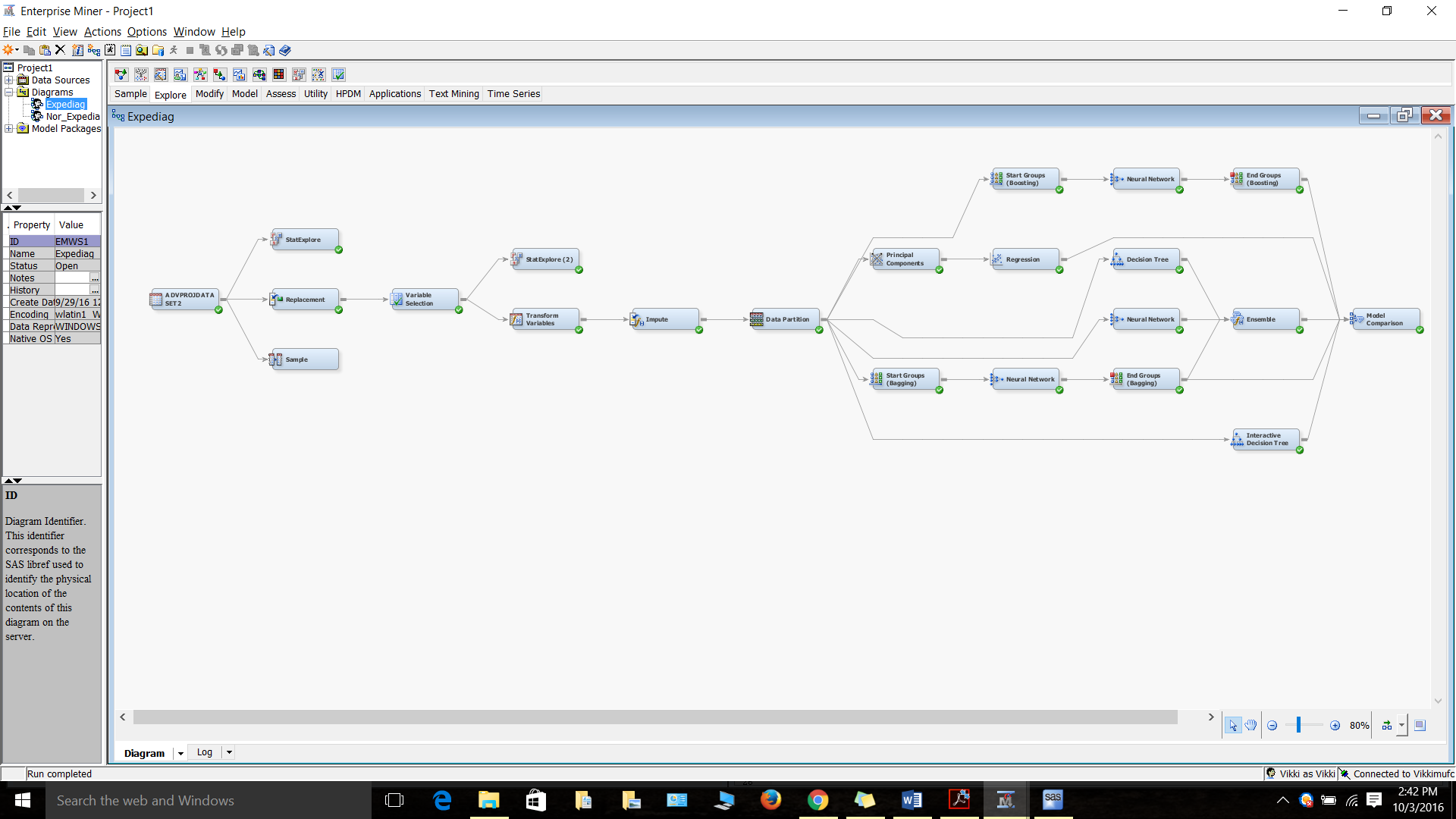
Comparing all these models, Ensemble gave the result which we intended for and hence we chose that as the best model.

The complete diagram is given below:



To give a more clear and brief picture of it, we divided the final diagram into 2:





## **Learnings from the project:**

* We used three models Neural Networks, Regression and Decision Tree in our MIS 6324 class but through this project we learnt more about bagging and boosting concepts.
* Use of Ensemble node which is by far the best model which we have come across.
* We also learnt more on impute node which we did not focus on our previous class. Changing the properties gave us different insights which in turn gave us better results.
* Interactive decision tree was something new and we tried to make use of it to the fullest.
* Use of Principal Components. Though we did not get the desired result we learnt more about this node.