*MIS 6334.501 Advanced BA*

**Project 1:**

**Data and Model Analytics using SAS Enterprise Miner**

**Final Report**

Instructor: Xianjun Geng

**Group 6**

Hua Guo

Yanwei Jia

Fei Cen

Pengfei Liu

Yi Liu

Lu Chen

[1. Data Preprocess 3](#_Toc463427026)

[1.1 Read Data Description 3](#_Toc463427027)

[1.2 Understand All the Variables 3](#_Toc463427028)

[1.3 Check Abnormal/Rejected Variables 3](#_Toc463427029)

[1.4 Check Statistic Results 4](#_Toc463427030)

[1.4.1 Verify variable distribution by distribution plot 5](#_Toc463427031)

[1.4.2 Check outliers by boxplot 6](#_Toc463427032)

[1.5 Variables Selection According to above Analysis 6](#_Toc463427033)

[2. Advanced Data Engineering Techniques 7](#_Toc463427034)

[2.1 Target Leakage 7](#_Toc463427035)

[2.2 Highly Correlated Variable Removal 7](#_Toc463427036)

[Identify correlated variables 7](#_Toc463427037)

[Find relationships among variables 7](#_Toc463427038)

[Delete redundancy variables 7](#_Toc463427039)

[2.3 Zero- and Near Zero-Variance Variables Removal 8](#_Toc463427040)

[2.4 Variable transform according to skewness 9](#_Toc463427041)

[2.5 Imputation 11](#_Toc463427042)

[Combined method with Distribution and Tukey’s Biweight 11](#_Toc463427043)

[K-nearest neighbors (KNN) method 11](#_Toc463427044)

[Bagged trees method 11](#_Toc463427045)

[3. Modeling 12](#_Toc463427046)

[3.1 Neural Network 12](#_Toc463427047)

[3.2 Decision tree 13](#_Toc463427048)

[4. Summary 13](#_Toc463427049)

[4.1 Diagram 13](#_Toc463427050)

[4.2 What we learned 14](#_Toc463427051)

[5. Appendix 14](#_Toc463427052)

[5.1 Appendix\_Diagram 14](#_Toc463427053)

[5.2 Appendix\_Results 17](#_Toc463427054)

# 1. Data Preprocess

## 1.1 Read Data Description

Before beginning we read the data description carefully and understood the scenario to know well about the Expedia project.

## 1.2 Understand All the Variables

Read data description carefully, make sure understand each variable’s data type. Then check the data type created by SAS Enterprise Miner. Right click on the data→Edit variables. Check the “Level” column to see whether the Level matches what we think.

In this step, we adjusted a few levels, such as change x3, x5 to ordinal, x7, x35 to interval, and 38 to nominal. The adjusted variables level please see below.

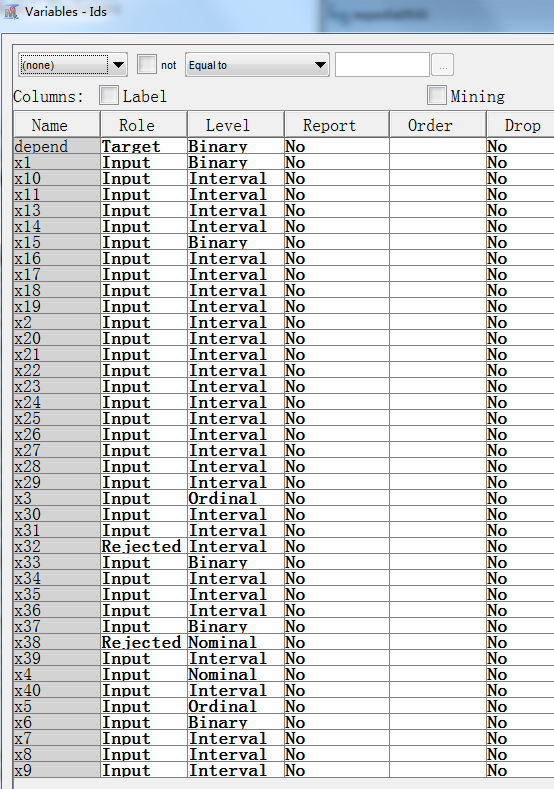
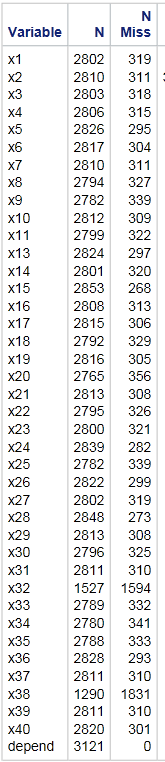
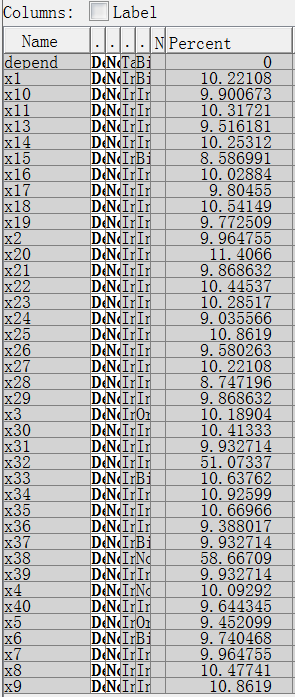


Figure 1 Adjusted variables level

## 1.3 Check Abnormal/Rejected Variables

In the above variable table, we can see almost every variable’s role is Input, except two variables. X32 and x38 were rejected by SASEM. So, the next step we checked the reason for the rejection of x32 and x38.

We explored the statistics summary for Expdeia form SASEM(left below picture), and mean procedure from SAS(right below picture).



(StatExplore → Edit Variables → Statistics)

Figure 2 Missing value stats

We found that the number of missing value of x32 and x38 is 1594 and 1831, respectively. Relative to the total number 3121, the missing rate of x32 and x38 all over 50%. We thought this is the reason SASEM rejected x32 and x38. We may put more concerns on these 2 variables in the modeling step.

## 1.4 Check Statistic Results

We checked mean procedure of every variable by using SAS. We found that x9, x13, x14, x16, x18, x19, x22, x28, x34, x35, and x36 are not normal distributed. According to central limit theorem, all the other variables can treat as normal distributed. The means procedure table please see below:

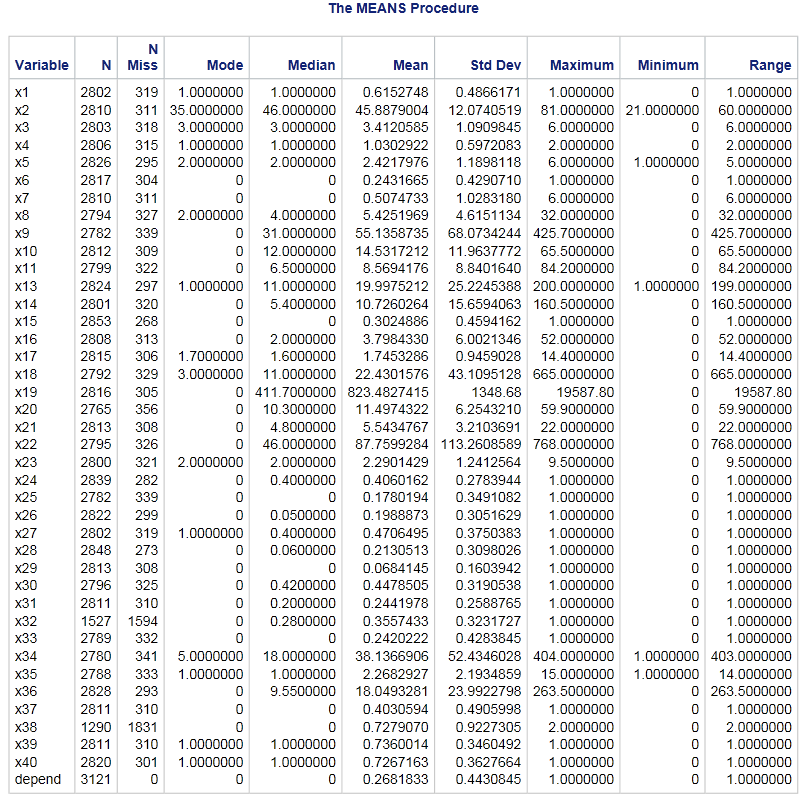


Figure 3 Mean Procedure

### 1.4.1 Verify variable distribution by distribution plot

To verify the distributions of above variables’. We also visual checked those variables distribution plots. Here we’ll use x19(left) and x22 (right) as examples. Those distribution plots verified that x9, x13, x14, x16, x18, x19, x22, x28, x34, x35, and x36 are not normal distribution. We should put more care about those not normal distributed variables in the modeling procedure.

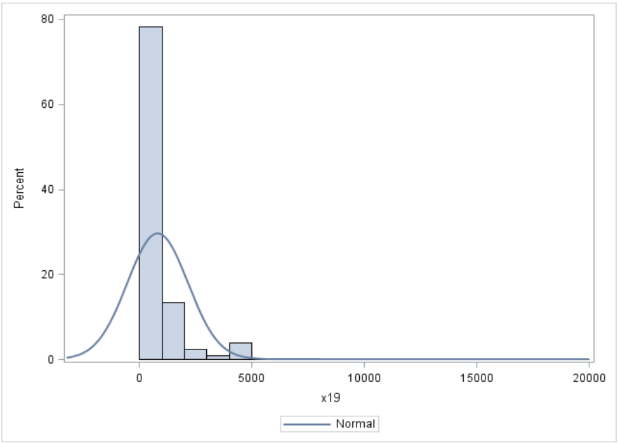
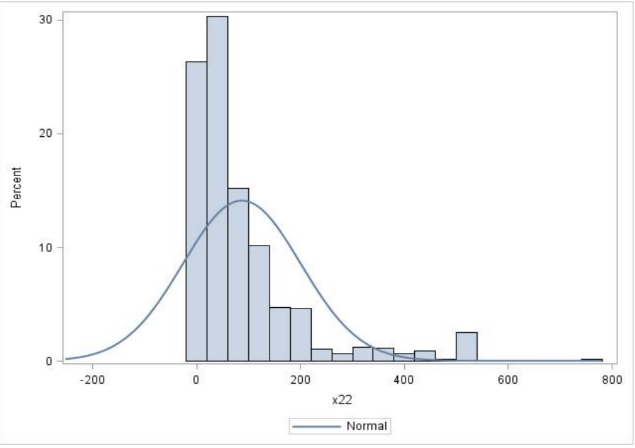


Figure 4 Distribution plot of variable x19 and x22

### 1.4.2 Check outliers by boxplot

We used boxplot for checking outliers. From boxplots of every variables, we found that:

No outliers: x2, x24, x27, x30, x31, x32, x39, x40

Has outliers: x10, x11, x16, x17, x18, x20, x23, x25, x35

Has too many outliers: x13, x14, x19, x22, x26, x28, x29, x34, x36

We should pay more attention to these outliers in later. Here, we use x19 (left) and x22(right) as examples. We prefer keep all the outliers, but different models might treat outliers in distinct method.

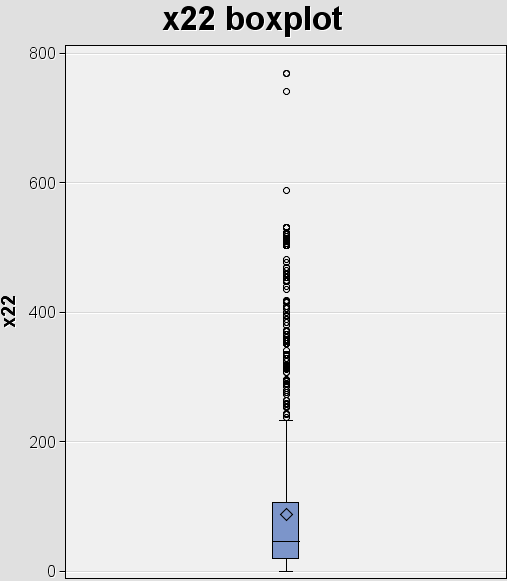
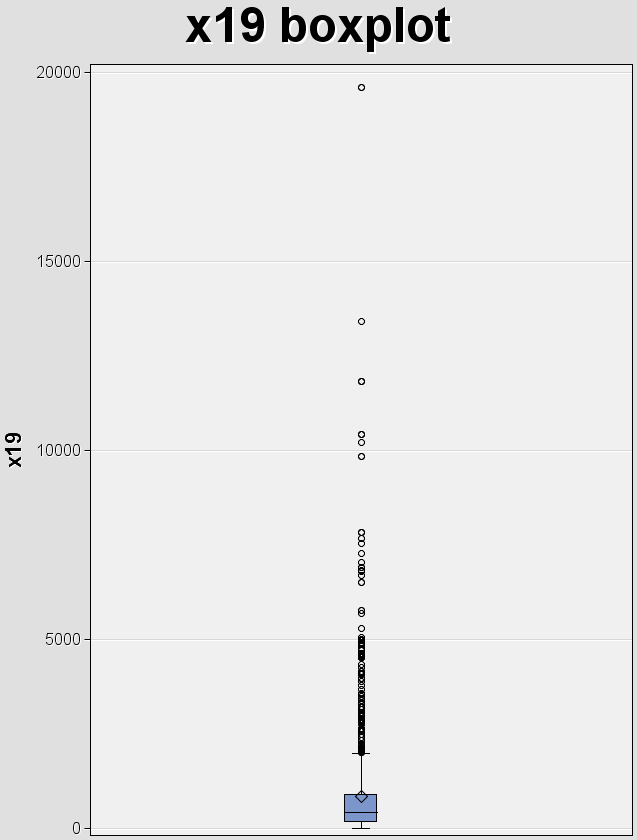


Figure 5 Boxplot of variable x19 and x22

## 1.5 Variables Selection According to above Analysis

The StatExplore helped us select worth variables. We may take top 12 variables as a try. The top 12 variables are: x33, x14, x25, x7, x36, x11, x13, x9, x20, x22, x21. We may pay more concerns on these variables in the modeling step.

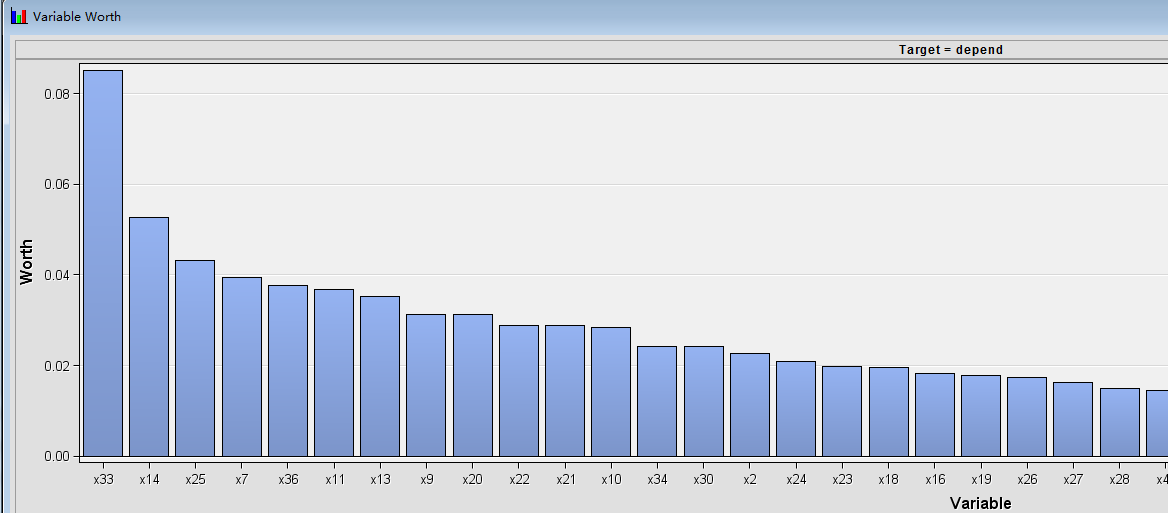


Figure 6 Variable worth chart from SAS Enterprise Miner

# 2. Advanced Data Engineering Techniques

## 2.1 Target Leakage

Our professor told us variable 12 (x12) is a very unique one, and remaindered us to treat it carefully. At every beginning, we made a mistake that we treated x12 as a normal feature to predict the target. After talk with other students who got information from our professor, we found that x12 is just recorded the part time of one session, we need to combine x12 with target.

Integrate variable 12 and target:

1) Compare x12 and target we found that a rule exits between x12 and target. The rule show below:

2) if x12 = ., then target = target;

else target = target + x12

3) use SAS combine the two variables, and drop x12

## 2.2 Highly Correlated Variable Removal

### Identify correlated variables

We checked the correlation among variables to help us find out problematic variables. We also used correlation results help us figure out what we need to pay more concerns in the modeling process. Because some models that thrive on correlated variables (such as regression), while other models my benefit from reducing the level of correlation between the variables. To get a more robust prediction result, different model may take distinct methods.

### Find relationships among variables

After read the data description and check the correlation results, we found some mathematical relationships among variables. The result list below:

X26 = x28

X18/x19 = C

X9/x19 = x28

X8/x18 = x27

X9/x8 = x11

x39 = x40

x13/x34 = x39

x7/x16 = x25

Based on these relationships, we calculated the missing values. During the calculating process, we found that there are some wrong records exist in the Expedia dataset. X25, x27, and x28 are “percentage”, so these variables should not greater than 1. But if we use x7/x16, for example, the result sometimes greater than 1. It’s not reasonable. We also found that the original dataset set a threshold, 1, for the variable. So, when we found any calculate result of x25, x27 and x28 greater than 1, we set it as 1.

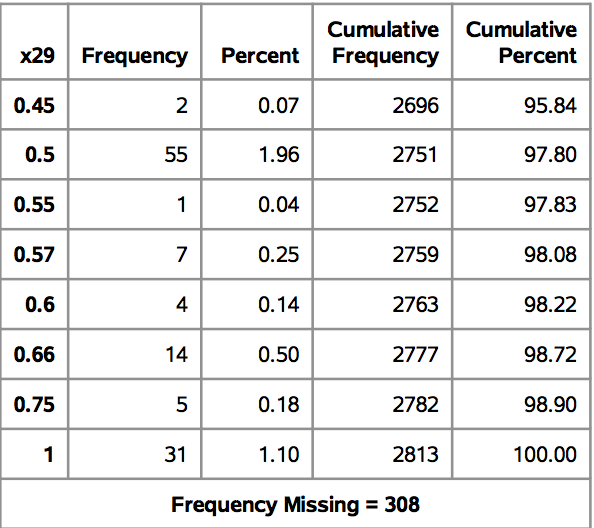
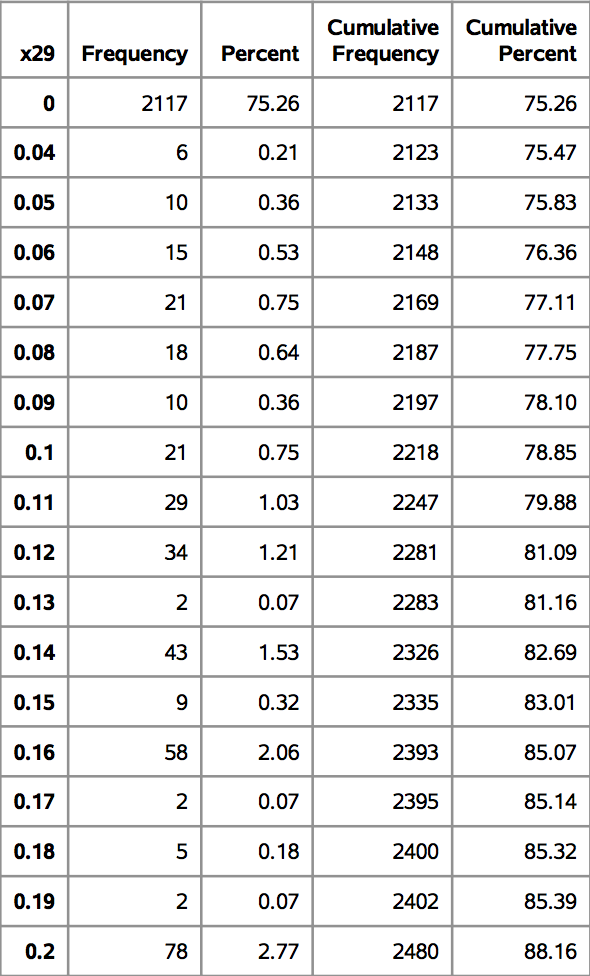
### Delete redundancy variables

After the previous step. We deleted redundancy variables: x26, x19, x27, x11, x39, x25.

## 2.3 Zero- and Near Zero-Variance Variables Removal

In some situations, the data can have variables that only have a single unique value (i.e. a “zero-variance variable”). For many model, this may cause the model to crash or the fit to be unstable.

Similarly, variables might have only a handful of unique values that occur with very low frequencies. For example, in the Expedia dataset, the x29 descriptor (No. of sessions start with this site/total sessions of this site) data have a few unique numeric values that are highly unbalanced:

Table 1 Top3 frequency of variable x29

|  |  |  |
| --- | --- | --- |
|  | X29 | Freq |
| 1 | 0 | 2117 |
| 2 | 0.2 | 78 |
| 3 | 0.25 | 61 |

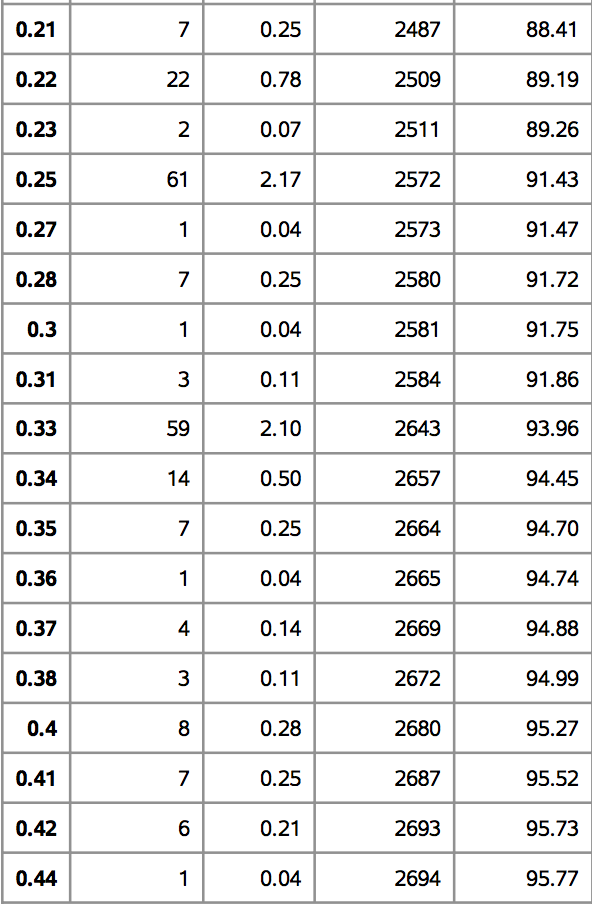


Figure 7 Frequency of variable x29 from SAS

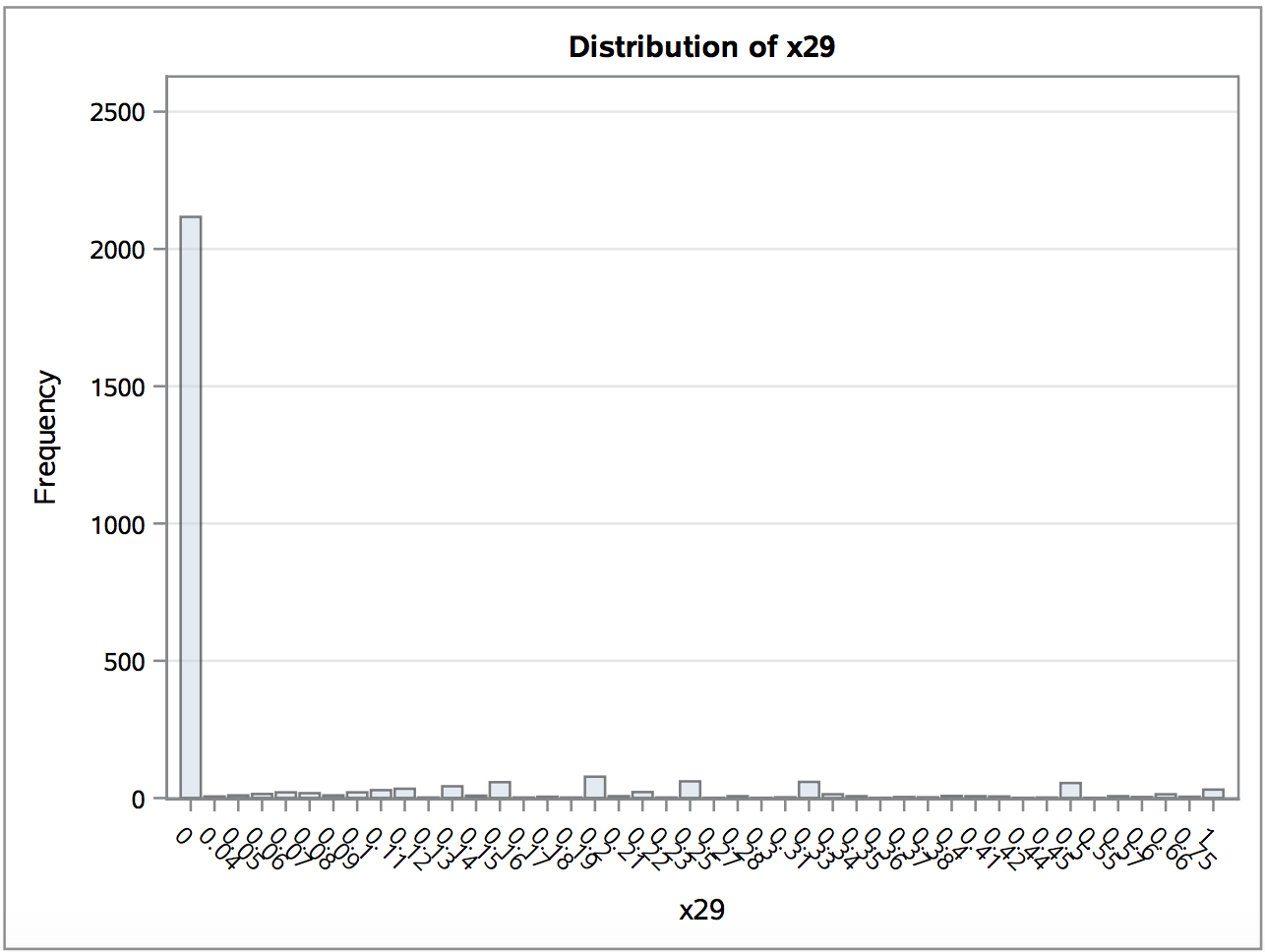


Figure 8 Distribution of variable x29

These variables may become zero-variance variables when the data are split into cross-validation/bootstrap sub-samples or some samples may have an undue influence on the model. These “near-zero-variance” variables may need to be identified and eliminated prior to modeling.

To identify these kind of variables, we calculated the following 2 metrics:

1. Frequency ratio =The frequency of the most prevalent value / The second most frequent value, which would be near one for well-behaved variables and very large for unbalanced data and

i.e. Frequency ratio of x29 = 2117/78=55.8

1. Percent of unique values = (The number of unique values / The total number of samples) \*100 that approaches zero as the granularity of the data increases

i.e. Percent of unique values = (44/3132) \*100 = 1.408

If the frequency ratio is greater than a pre-specified threshold and the unique value is less than a threshold, we might consider a variable to be near zero-variance.

After zero- and Near Zero-Variance Variables checking, we consider variables x29 should be removed.

## 2.4 Variable transform according to skewness

We checked the skewness (Figure 9), and distribution plot. When skewness of a variable is greater than 2.0, we think that variable needs to be transformed by log/log10 to normal distribution. We use log for variables ranging from 0 to 1, otherwise we adopted log 10 as the transform formula. By this strategy, we found that x8, x9, x13, x14, x16, x17, x18, x22, x28, x34, x36 should be transformed. Among them, x28 is transformed by log, and others are transformed by log 10.

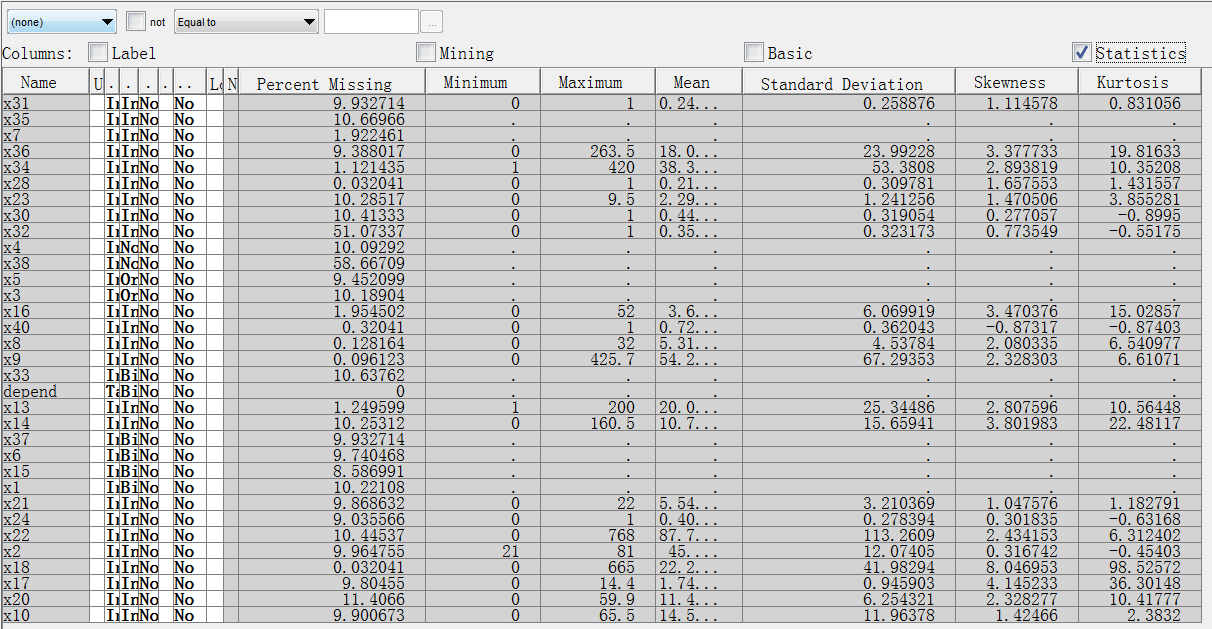


Figure 9 Skewness of all variables

Transform settings shows below:

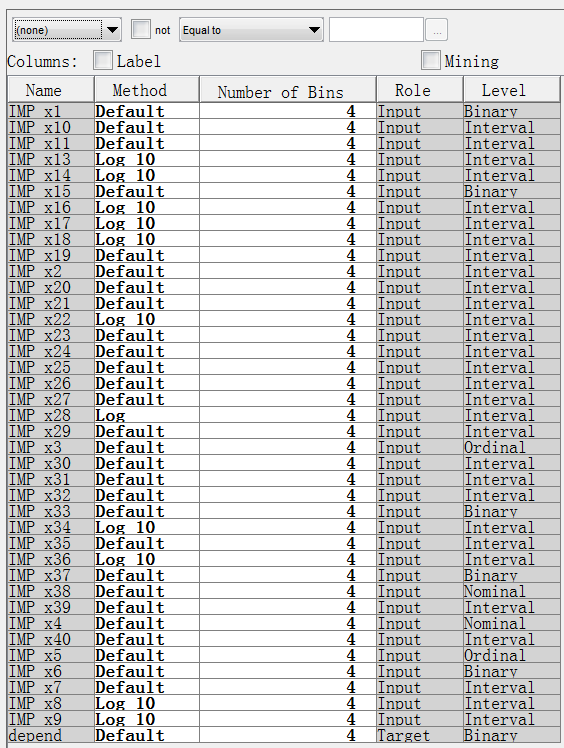


Figure 10 Transform settings

But this log and log10 transformation strategy didn’t perform well. So we adopted Best transformation method as our transformation method for variables as below:

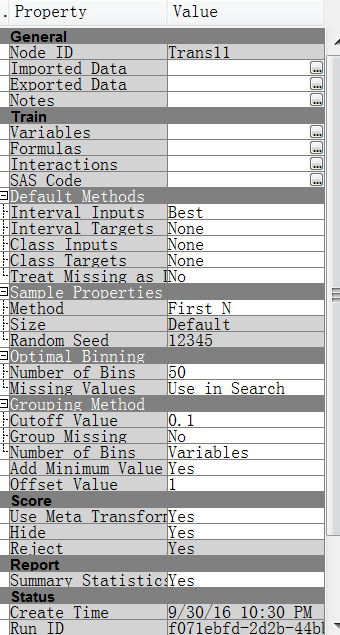


Figure 11 Transform settings we adopted

## 2.5 Imputation

We adopted 3 methods to impute all the interval variables’ missing values:

### Combined method with Distribution and Tukey’s Biweight

The first method we chose is a combined method with both Distribution and Tukey’s Biweight. This method is the most powerful one among SAS Enterprise Miner provided we tried. All the non-interval variables we selected distribution method. The details of imputation method list below:

Table 2 Imputation method for all variables

|  |  |
| --- | --- |
| Distribution | Tukey’s Biweight |
| All the nominal,  ordinal, and binary variables | All the other interval variables |

### K-nearest neighbors (KNN) method

For doing imputation with KNN method, Centering and Scaling is pre-required. We used R centering and scaling variables, and then we imputed missing values with KNN method.

### Bagged trees method

We also adopted bagged trees to impute. In theory, this is a more powerful method of imputing.

We may compare those 3 methods in the later.

# 3. Modeling

First, we set decision weights according project instruction. Please see the picture below.

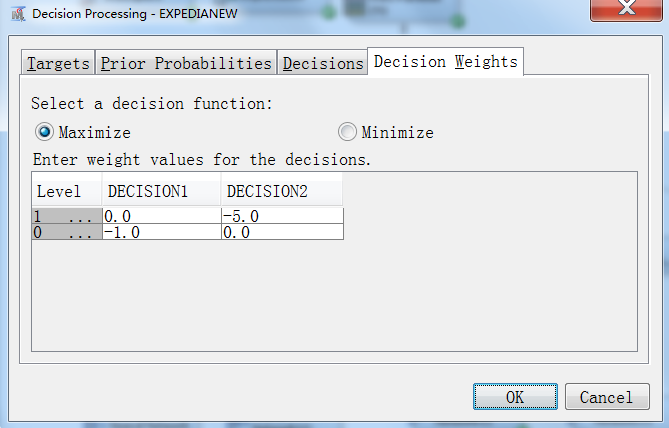


Figure 12 Decision weight setting in SAS Enterprise Miner

We first use Data Partition in SASEM to cut the whole data to 2 parts. We used 55% as training dataset, while used 45% as validation dataset.

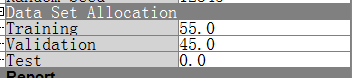


Figure 13 Dataset partition

## 3.1 Neural Network

Here, we want to share a more helpful experience which we found during the neural network modeling process.

First, we create the nodes as below:

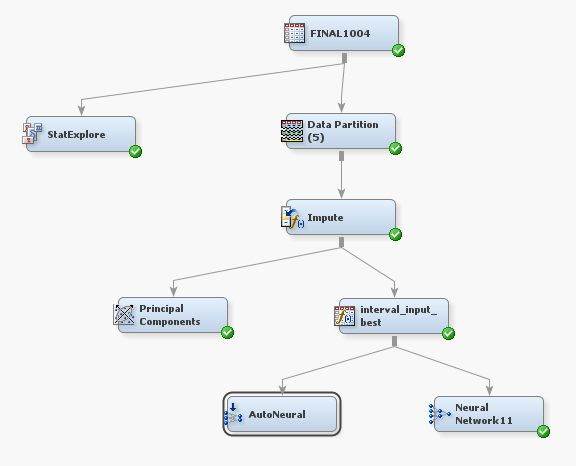


Figure 14 Modeling diagram for neural network

Click the transform node → Property → Default Methods → Interval Targets → Best, and then click run. After that, look through the results to find which formula the system used. And then adjust that parameter by yourself. In our project, the system used Optimal bin as the best formula, so we tried several bin settings, finally we found the 50 can give us the best result. We also tried Huber, Andrew’s wave, Mid-Mimum, Tree surrogate, Tree, Distribution, Midrage, Median, Minimum, Maximum, and Mean, but Tukey’s Biweight is the most help one. So, we used 50 as bin’s parameter in neural network modeling.

Similarly, for the non-interval variables, SAS Enterprise Miner adopted distribution method. We also reviewed the distribution plots, we thought impute the missing values by distribution is make sense. So, we took distribution method for non-interval variables for modeling.

We also tried PCA, but no help here.

We tried the original dataset, removed redundancy variables (cleaned) dataset, imputed by 2 groups (depend=0, and depend = 1), bagged imputed dataset, and KNN imputed dataset.

All the train and rate misclassification rate and NN results addressed below:

Table 3 Final results comparison of different neural network models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Auto Neural Network | | Neural Network | |
|  | Train | Valid | Train | Valid |
| Original dataset | 0.011655 | 0.071 | 0.011655 | 0.071 |
| Cleaned dataset | 0.0169 | 0.08 | 0.038 | 0.092 |
| Bagged tree imputation dataset |  |  | 0.030 | 0.092 |
| KNN imputation dataset | 0.011 | 0.099 | 0.039 | 0.098 |

## 3.2 Decision tree

1. Data Imputation

We have done a lot of dealing with most of the missing data in pre-processing of data. But to improve the prediction, we still use SAS EM to do the imputation of the lest missing values.

Setting properties:

Missing cutoff🡪60

Class Variables--Default Input Method🡪tree surrogate

Default Target Method🡪none (if choose any except none, there will be no profit comparison in the final result)

Interval Variables—Default Input Method🡪Tree Surrogate

Default Target Method🡪Turkey’s Biweight (Does not impact much on the final result)

The order of data partition and data imputation, in decision tree, first data partition and then imputation give a better result.

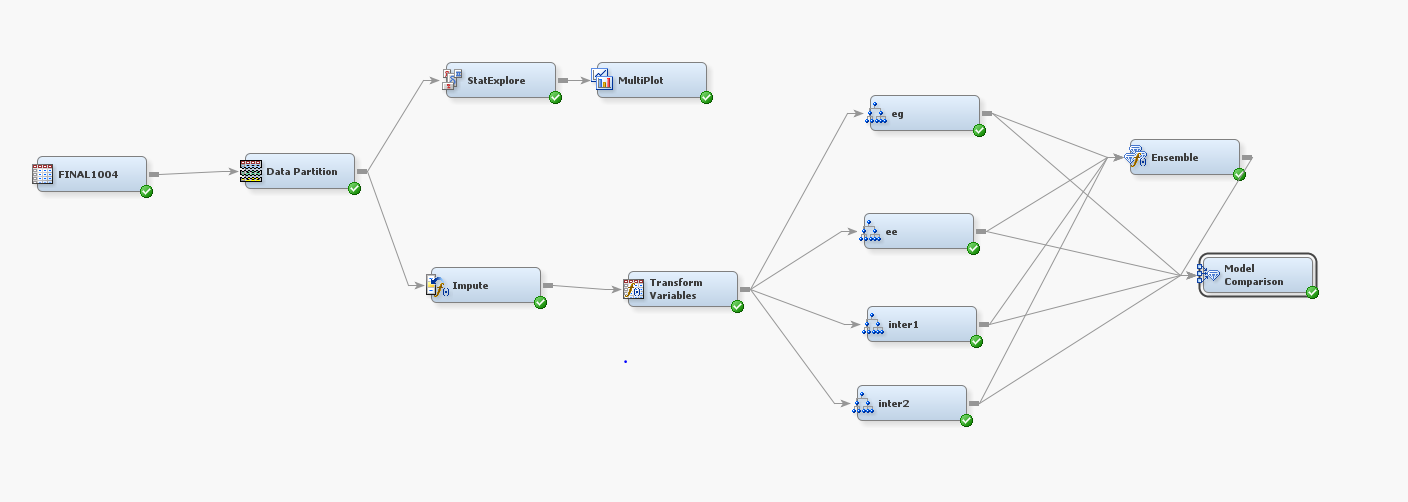
1. Data Transform

To deal with skewed data, we use data transform to make it better normalized.After analyzing the stat explore, the following variables are not normalized distribute, so we made log and log10 transformation. X34 X16 X13 X14 X18 X22 X36 X9 🡪LOG10, X28 X40 X7 X39🡪LOG

1. Variable selection

Variable selection is a big deal on every model, if we can accurately choose the top n important variables, we will get good results compared with the result from the total 41 variables. But when we manually choose the variables we thought matters combined with the outliers plot and correlation rate, we did not improve the prediction on decision tree. In contrast, the result become bad.

1. No sampling is needed because we only have 3121 records.
2. Model Improvement



**Decision Tree**

The best model Setting Properties(profit:-0.15231, mis rate: 0.10107)

Normal Target Criterion🡪Entropy

Ordinal Target Criterion🡪Gini/ Entropy

Missing values🡪use in search

Maximum Branch🡪5(less or more gives bad result)

Use Input Once🡪No

Maximum Depth🡪10

Minimum Categorical Size🡪2

Leaf size🡪6(less or more give bad result)

Number of rules& Surrogate rules🡪5(4 or 5 does not matter)

Assessment measure🡪decision (profit consideration)

Compare with all tried decision trees, maximum branch , leaf size, and target criterion is the most critical settings.

**Interactive DT**

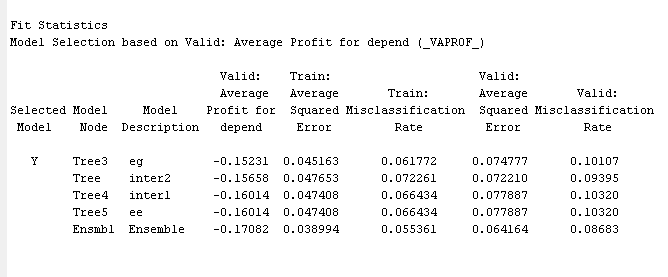
Comparing the worth rate of variables from stat explore, the importance rate of variables from autoed decision tree, and the logworh of variables, we choose from X20 X21 X30 X31 X19 X33 X7 X2 to do the interactive decision tree prediction.

After trying some interactive trees, we found out the most 2 important rules is maximum branch and leaf size, the result of maximum branch is 3 and leaf size is 5 gives a good profit and mis rate than the tree of maximum branch is 5 and leaf size is 6.

In interactive DT, if we can try as many times of pruning and splitting, we may get the good results, but is time costing and need more research and analysis on nodes selection.

1. Ensemble learning

Decision is not robust, so we use ensemble learning to integrate 4 trees and get a new model, but we found out the mis rate improved to 8% but the profit does not improve.



1. Conclusion

Decision tree is not the best model in our project, but during the process, we found out decision tree gives a good hint of variable or feature selection, may be benefit to other models. And during the pre-processing od data, our data source changed a lot times, but we found out once we made good settings of the properties, the result will not affected that much even some variables are rejected. EM will automatically choose the best nodes.

# 4. Summary

## 4.1 Diagram

The Enterprise diagram for the 5 datasets list below:

Original dataset diagram and result shows below. All the other four diagrams please see appendix\_diagrams.

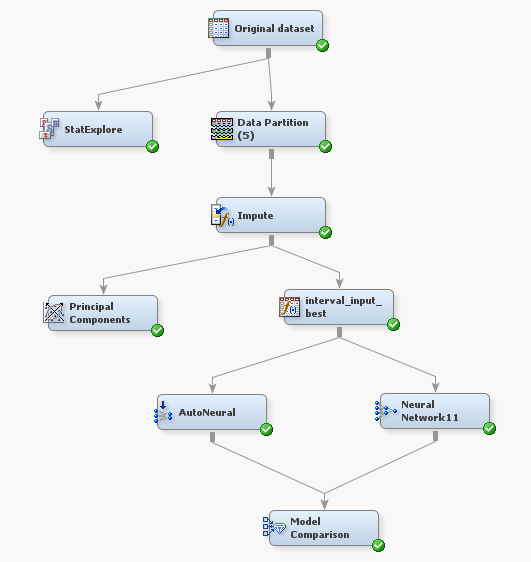


Figure 15 Original dataset diagram

## 4.2 What we learned

Good data paved good model training, so data preprocess and data engineering step plays an important role in Business Intelligence. We should spend more time and energy on the preprocessing step to get a more robust result. At the very beginning, we just did raw data preprocessing and imputation by default methods provided in SAS Enterprise Miner. No matter what training model we employed (we even use ensemble methods), the final training accuracy is around 90%, and the validation accuracy is less than 90%. However, after our exploring the variable relations between each other by visualization and reading data description carefully, we find some hints for our next-step data engineering work.

The second lesson we learned is the more specific method adopted, the better performance will be. Of course good data preprocessing is time consuming, however, it deserves. After we put so much time into imputation, we successfully improved our performance from 90% to 98% with simple training models. It is a huge improvement. We also did imputation with the help of R language, which is more flexible in data processing.

# 5. Appendix

## 5.1 Appendix\_Diagram

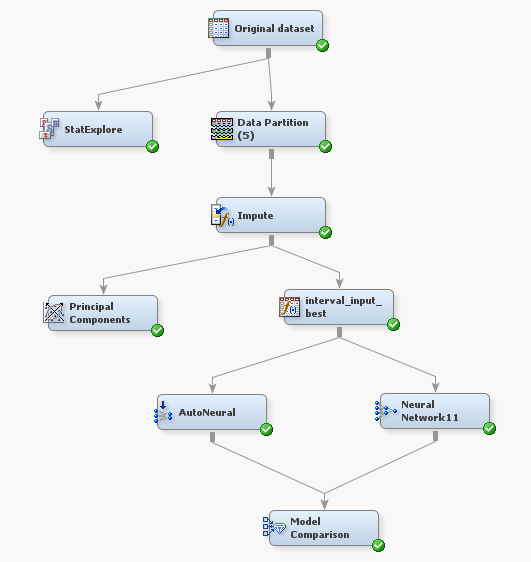


Figure 16 Original dataset diagram

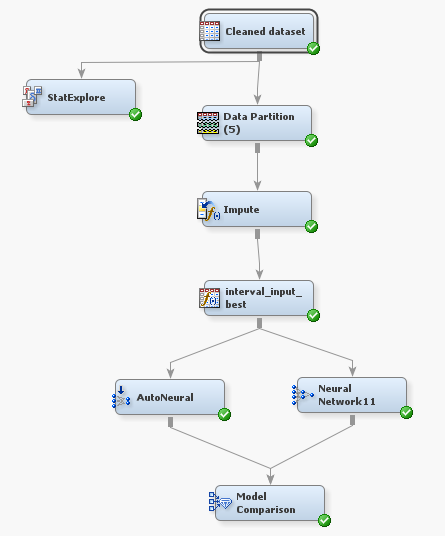


Figure 17 Cleaned dataset diagram

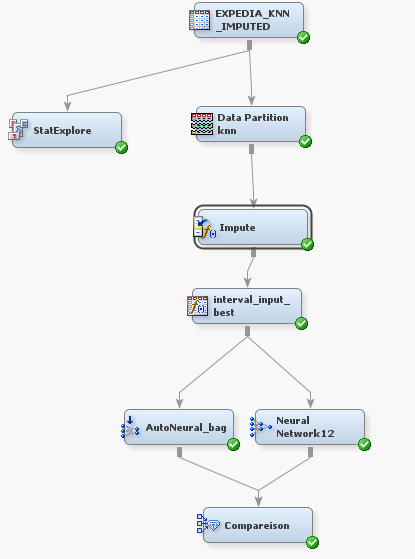


Figure 18 KNN imputation diagram

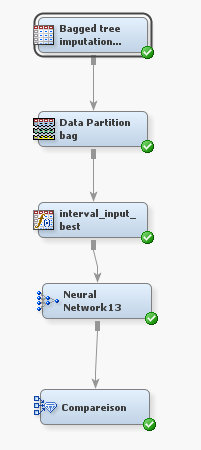


Figure 19 Bagged tree imputation diagram

## 5.2 Appendix\_Results

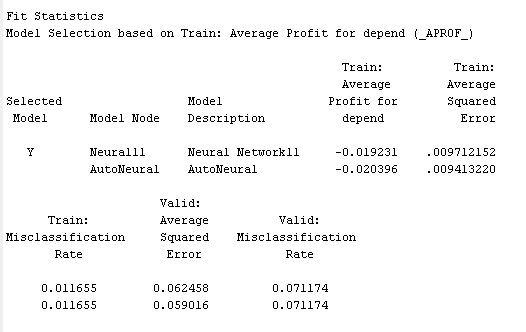


Figure 20 Training and validation results for original dataset

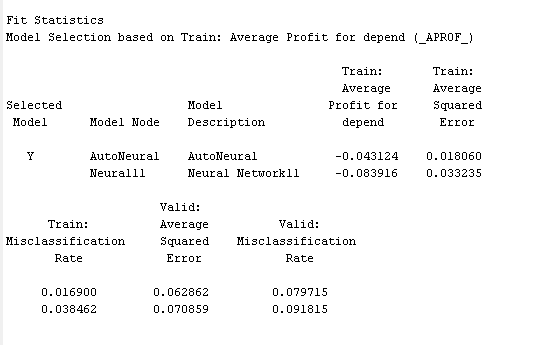


Figure 21 Training and validation results for cleaned dataset

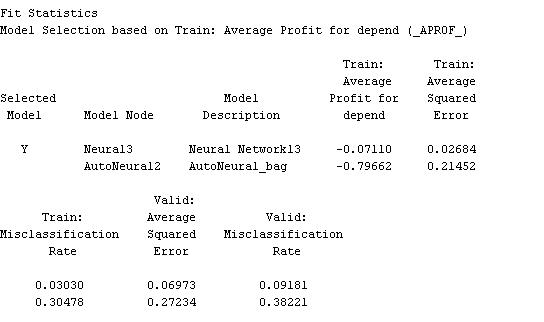


Figure 22 Training and validation results for bagged tree imputation dataset

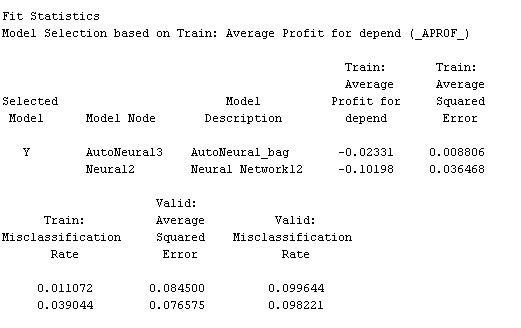


Figure 23 Training and validation results for KNN imputation dataset