**Project Report**

# Data and Model Analytics

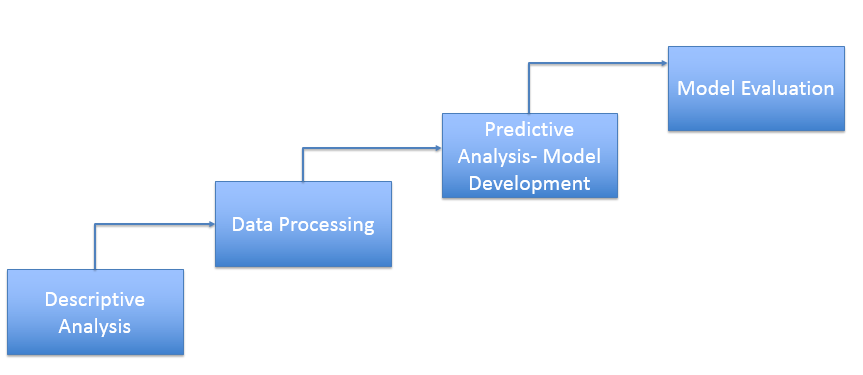
# Dataset: Expedia Sessions Data of Users

# Fall 2016

**Objective**

Predict whether or not a user is going to make a booking by analysing the behaviour of users on the website. Perform analysis on data tracked online for site through SAS Enterprise Miner. This prediction can enable company to share more personalized content, decrease the average time to first booking, and better forecast demand.

**Process Flow**

****

**Descriptive Analysis**

**Techniques Used:**

* Understanding what is available
* Data distribution and missing value analysis
* Exploratory graph analysis

**Understanding**

**what is available**

* Data is site-centric
* Data set is combination of below information
  + Customer specific information
  + Session related information
  + Booking related information

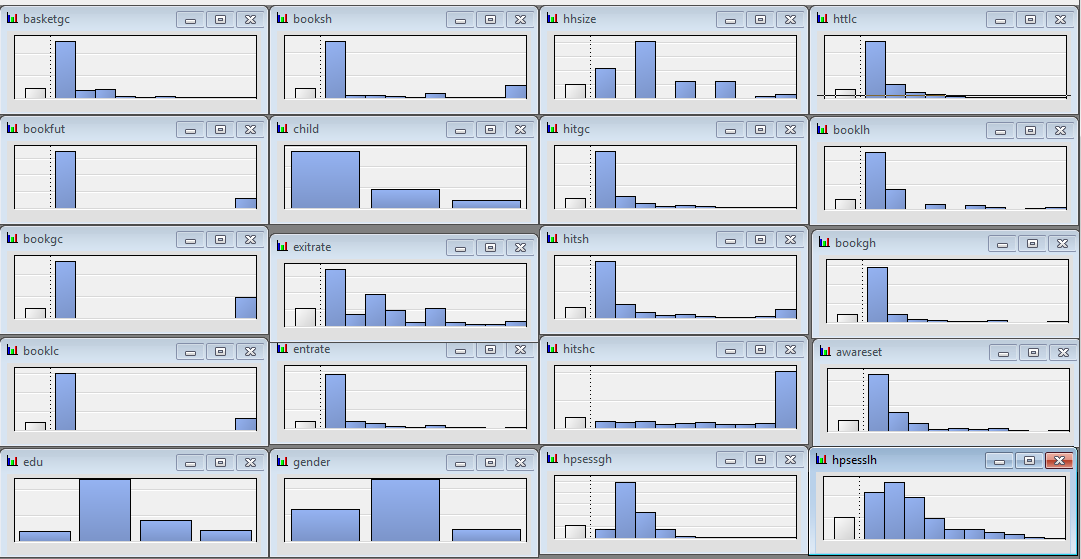
**Data distribution and missing value analysis**

We analysed distribution of each variable to get below details

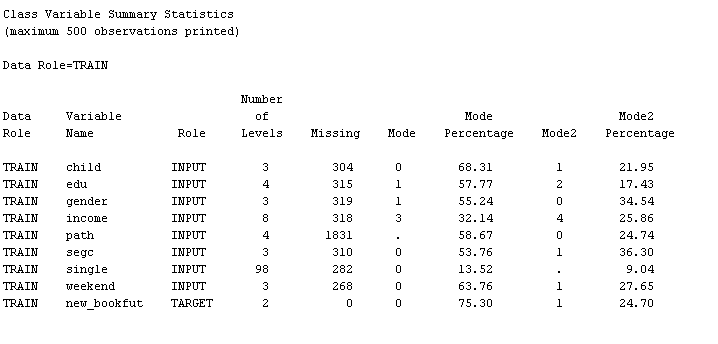
* Identify most representative information of variables
* Identify outliers
* Identify missing values of each variable

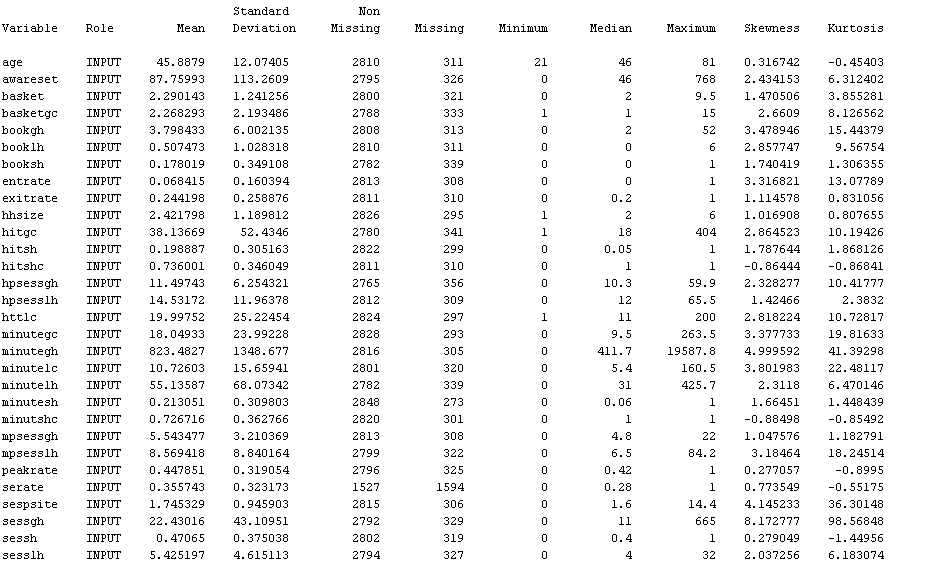
From the frequency distribution graphs from each variable, we concluded,

* Most of the session variables of type interval are right skewed
* Variables don’t have outliers which resides far away from general distribution of variable

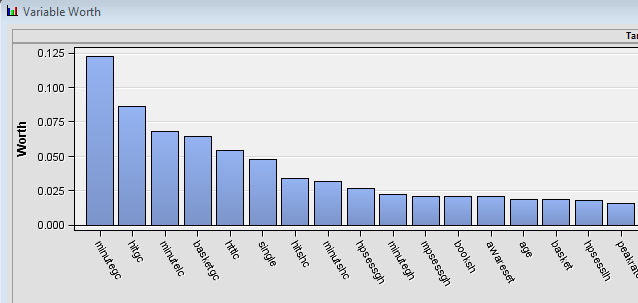


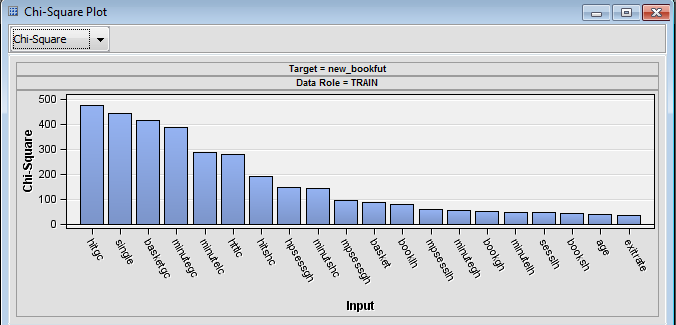
Data set has missing values at variable level as shown in below report





**Variable worth plot**



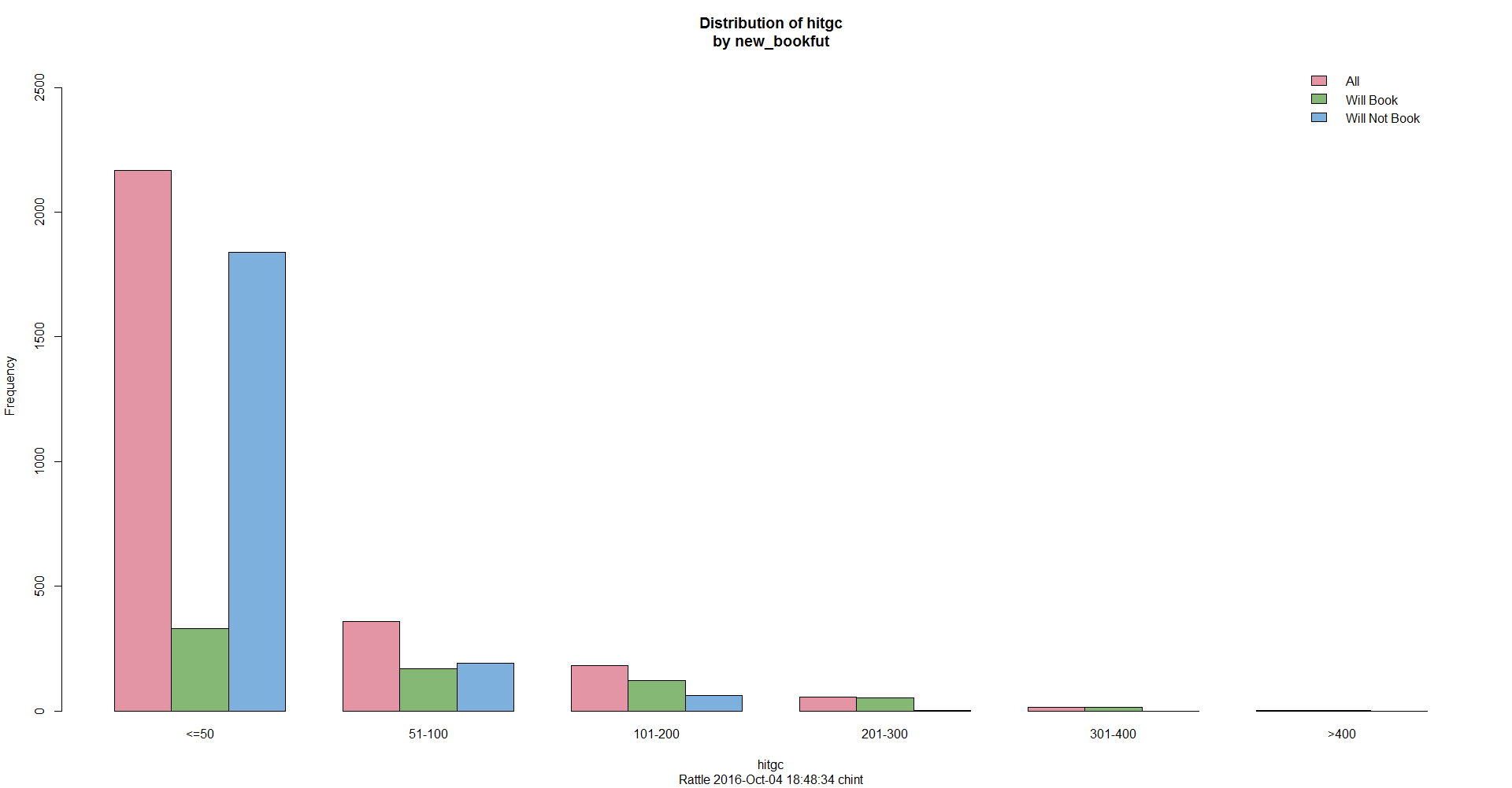


INSIGHT – From variable worth graph, we can conclude that demographic and customer specific attributes have less impact on final booking.

**Exploratory Graph Analysis**

We analysed the distributions of the variables that have the maximum worth or impact on the target variable bookfut. This helps us gauge the high level relation and business impact of the variables on the target.

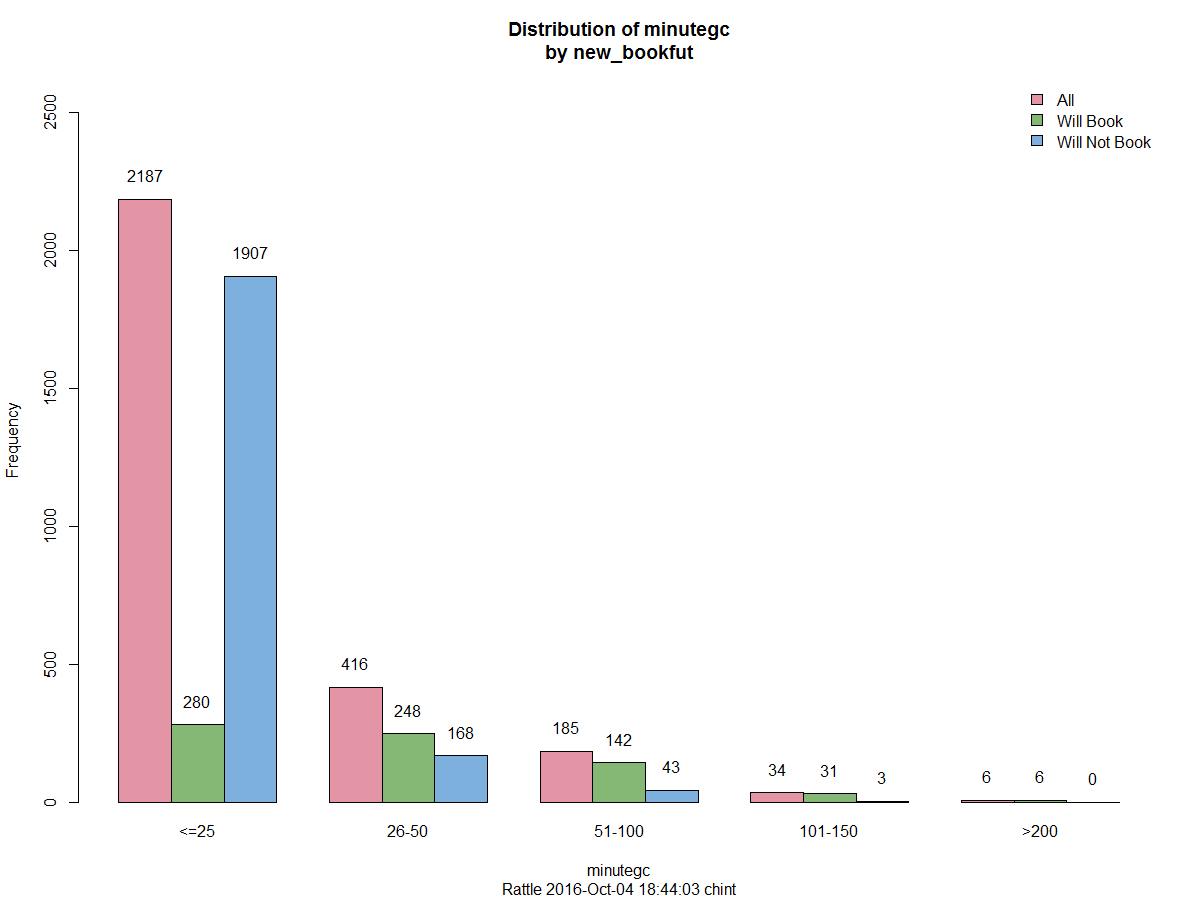
1. Distribution of hitgc vs bookfut



This shows that when the hits in the current session exceeds 50 the probability of a customer booking on the site increases from 15% to 67% and to about 95% if the hits exceed 200.

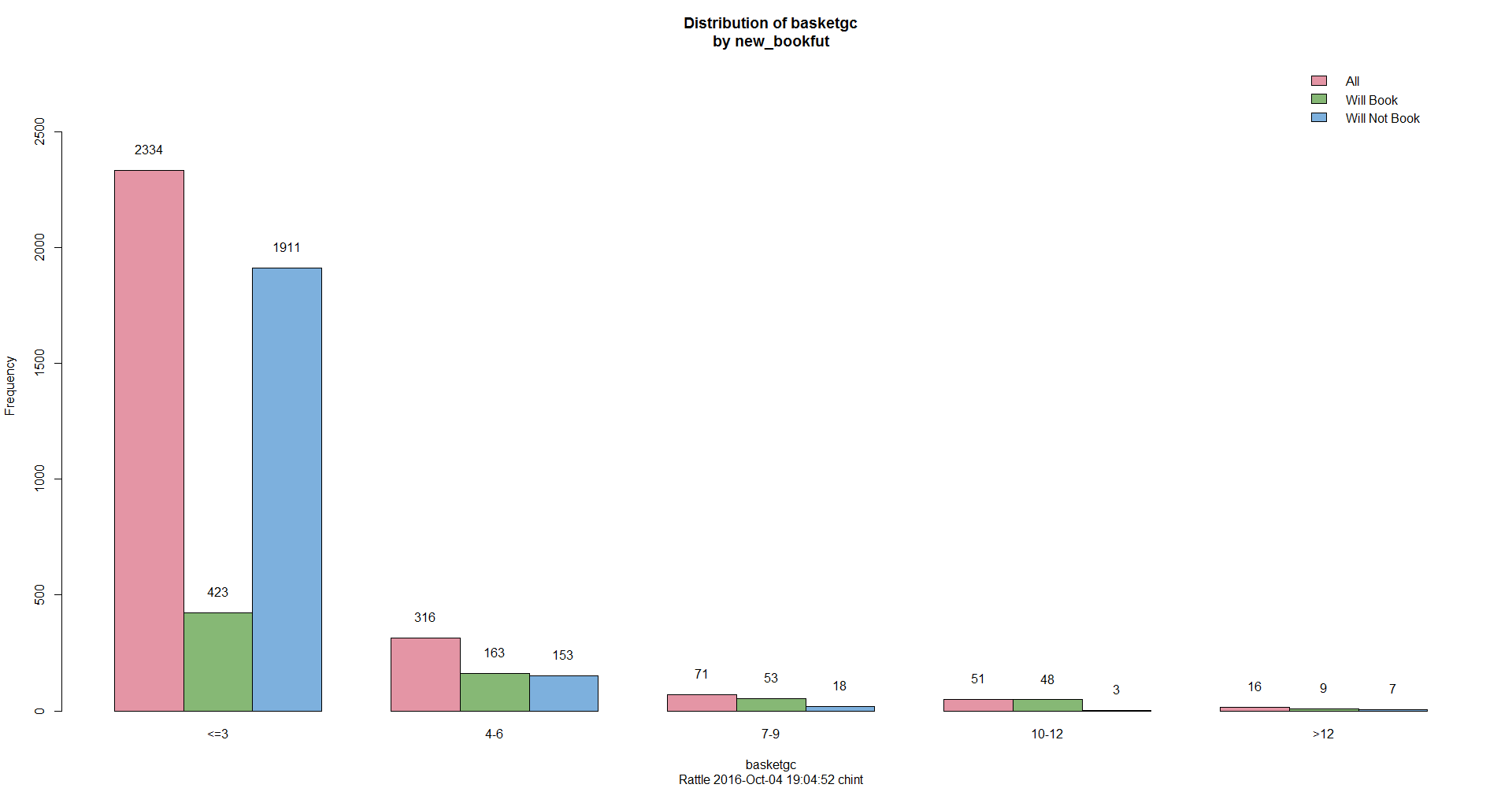
This shows that with the increase in the hits the probability of a customer booking increases.

1. Distribution of minutegc vs bookfut



We can see here that if the customer spends less than 25 minutes in the session than there is about 12.8% chance that he/she will book on the site. But this increases drastically to 59.6% if the time spent is 26-50 minutes and to about 77% for 51-100 minutes.

1. Distribution of basketgc vs bookfut



The distribution shows that with the with the increase in basketgc (number of shopping sites in the session) the probability of booking increases from about 18% (sites<=3) to 55% (sites>3).

**Data Processing**

**Objective:** Fit the data to model for better prediction

**Techniques Used:**

* Removal of redundant target variables
* Variable Transformation
* Data Partition
* Missing value handling by Imputation
* Variable Selection

**Target Variable Modification**

In data set we have variables,

booklc: Indicates if the user has booked at this site up to this point in the current session

bookfut: Indicates if this user is going to book in the remainder of the session

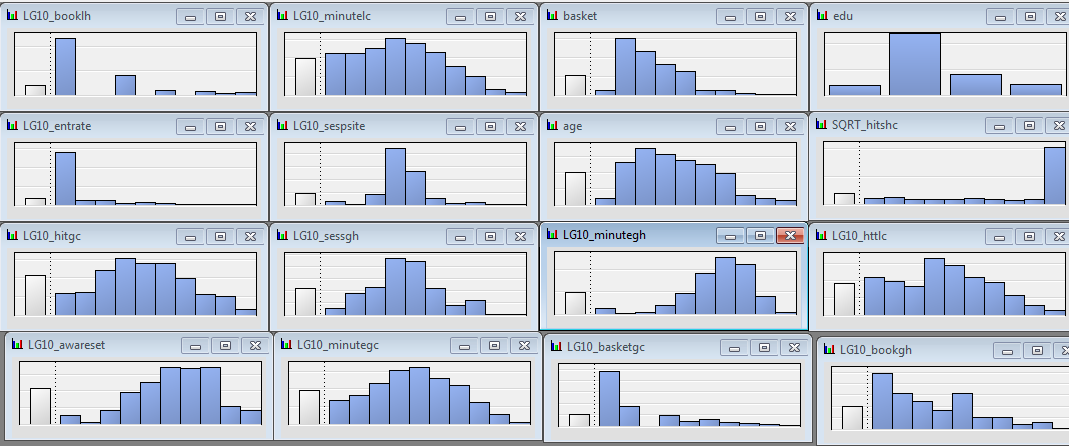
bookgc: Indicates if this user has booked at any sites up to this point in the current session

The decision tree analysis gave a 100% purity split for bookgc and booklc variables. This proves that these variables are dummy variables or a proxy for the final booking target variable bookfut.

booklc and bookgc are kind of redundant variables for our prediction because these are variables for the quick bookers. bookfut shows end results. We are interested in end result analysis. We should reject redundant variables. However, we should be cautious when rejecting them: They are almost the same as the target variable bookfut. We first need to change the targets from 0 to 1 for the observations with booklc=1 or bookgc=1, and then reject them. So we made new column new\_bookfut using base SAS programming.

**Variable Transformation**

From descriptive analysis of data set we have concluded that most of the session variables of interval type are right skewed. Predictive models like regression works on assumption of normal distribution. So to reduce complexity of model, we have used transformation. We have made settings to consider missing value as a separate level.

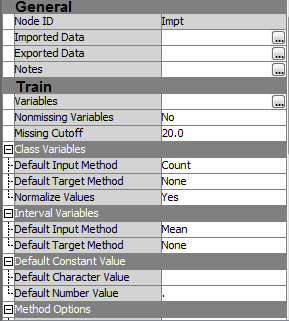


**Data Partition**

The training data set is used for preliminary model fitting. The validation data set is used to monitor and tune the model weights during estimation and is also used for model assessment. So we have used 55% of data for training and 45% for validation.

**Data Imputation**

We have transformed data using variable transform and made overall variable distribution normal. So, we chose most representative method for imputation. For class variables it is count and interval variables it is mean.



**Variable Selection**

The Variable Selection node is useful to evaluate the importance of input variables in predicting or classifying the target variable. The variables that are not related to the target are set to a status of rejected. Although rejected variables are passed to subsequent nodes in the process flow diagram, these variables are not used as model inputs by more detailed modelling.

Variable Selection has significantly improved the prediction outcome of neural network and regression. However, the variable selection had no impact on the performance of Decision tree. Feature selection and evaluation has been an important criterion in reducing the misclassification rate of the model. The Variable Selection node was used to identify the most important variables using a tree based algorithm. 15 variables out of the 41 were selected and fed into the Regression, AutoNeural and Neural Network algorithms and the results before and after the feature selection were compared. It showed that there is a very good improvement in the Neural Network models post the data pre-treatment and variable selection.

We have also tried variable selection using Principle Component Analysis, but it turned out that variable selection node helps to improve performance of model and PCA is not much effective for data set.

**Predictive Analysis- Model Development**

**Objective:** With predictive analysis, make a reliable model to predict whether a user is going to book or not

**Techniques Used:**

* Used techniques like decision tree, neural network, regression and boosting
* Initially we applied most common techniques, i.e. decision tree, neural network and regression to data without performing the data processing in order to benchmark the results. Three models are being applied in order to find the target of whether the user in the current session is going to place a booking or not
* We also tried Interactive decision tree, which is manual development of decision tree by changing split rules. We have not developed full decision tree using this technique due to time constraint, but we have provided analysis for one split.
* When further refinement of the models is ignored, Auto neural network turns out to be the algorithm in the model.

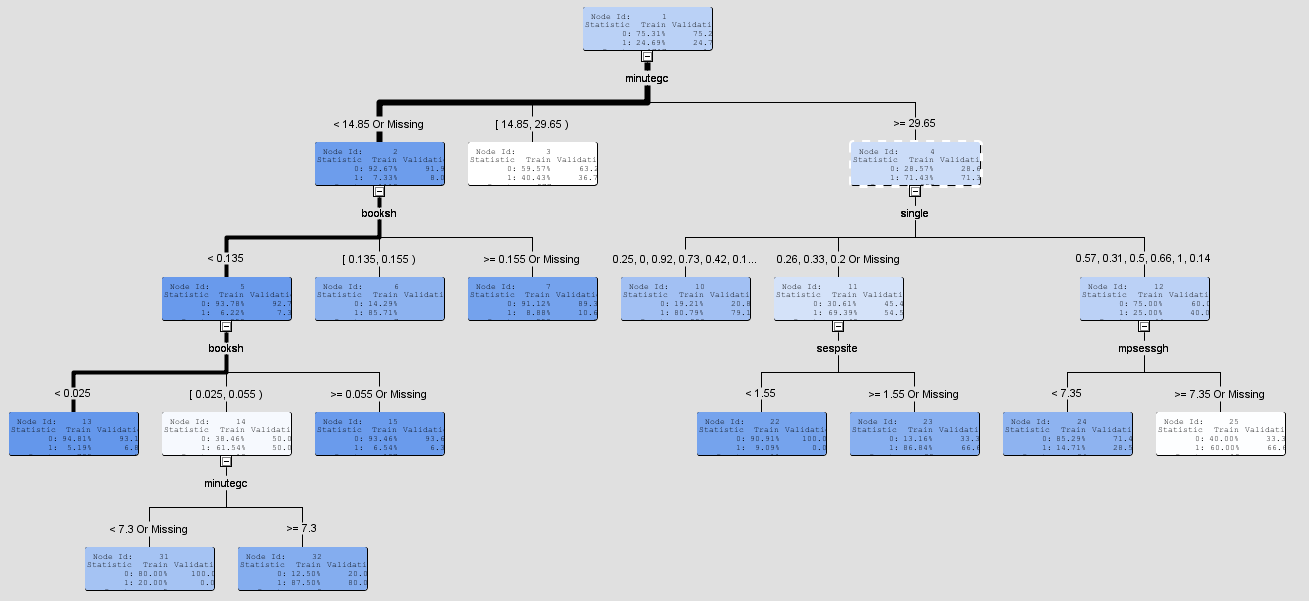
We have reported detailed analysis of all techniques used for model development as below

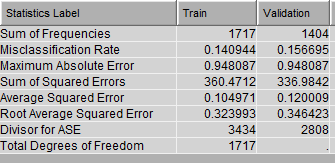
**Decision Tree**

Decision tree algorithm in itself is a variable selection algorithm and is not sensitive to missing variables or skewness. Decision tree is capable of dealing with missing values through surrogate splitting rules. The fact that the misclassification rate of decision tree is getting worse after variable selection implies that the surrogate splitting rules are more powerful than the traditional variable selection techniques for decision trees.

* **Without Data Pre-processing:**

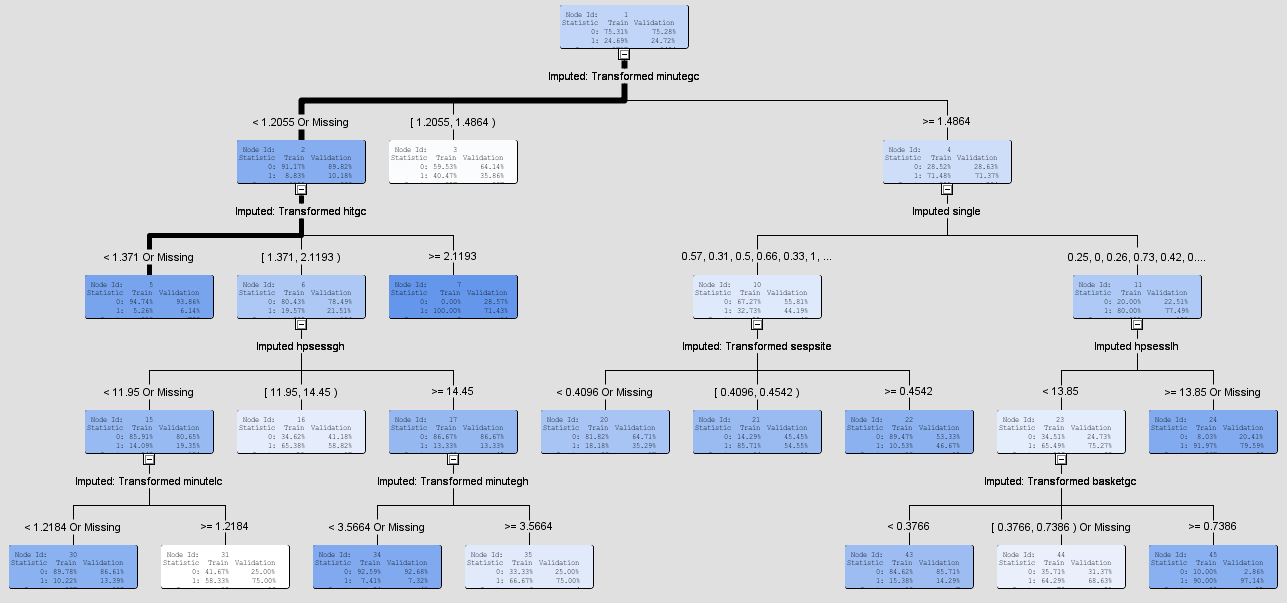
The decision tree was configured with number of surrogate levels to be 7 to automatically compensate for the data pre-processing. The tree obtained had a misclassification rate of about 15%. This is because the tree also makes extensive use of the demographic variables which are of least importance in order the classify the target.

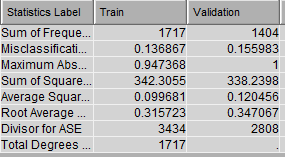




* **With Data Pre-processing:**

With data pre-processing tree is grown better and fit statistics also improved

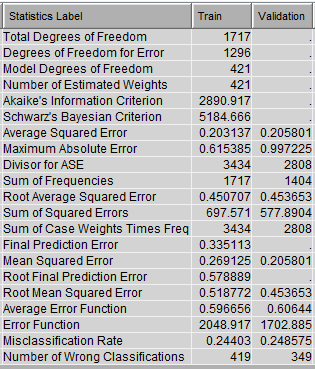




**Neural Network**

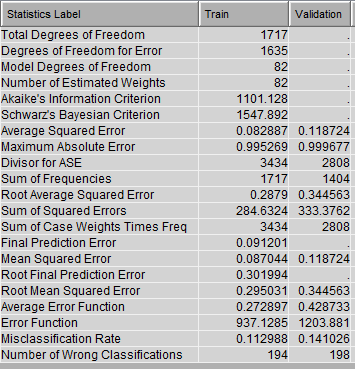
* **Without Data Pre-processing and Variable Selection:**

The neural network without data pre-processing seems to perform poorly having the misclassification rate to be around 24.8% which is greater than the prior probability itself. So clearly Neural network does not give good results without data pre-processing.



* **With Data Pre-processing and Variable Selection:**

With data pre-processing and variable selection, fit statistics improved.

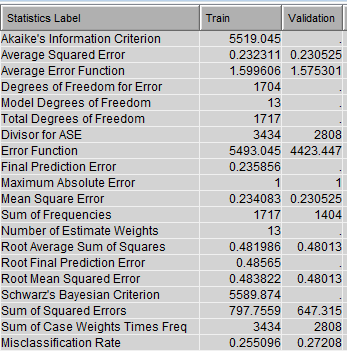


**Logistic Regression**

We have selected regression type logistic and our target variable is binary.

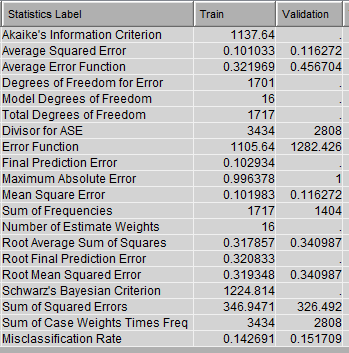
* **Without Data Pre-processing and Variable Selection:**

The misclassification rate is even more greater than the prior probabilities for the classifier. The misclassification rate is around 27% for logistic regression, so without data pre-processing this classifier performance is the weakest of all the other classifiers.

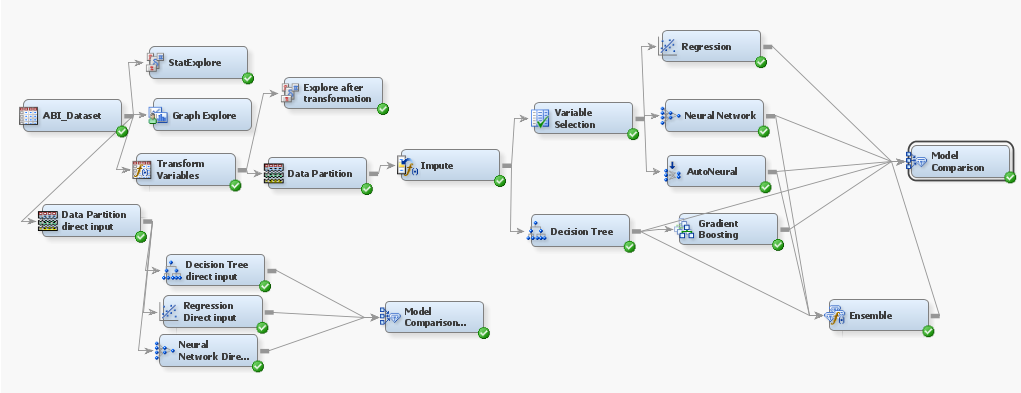


* **With Data Pre-processing and Variable Selection:**

With data pre-processing and variable selection, fit statistics improved.



**Complete Predictive Model**



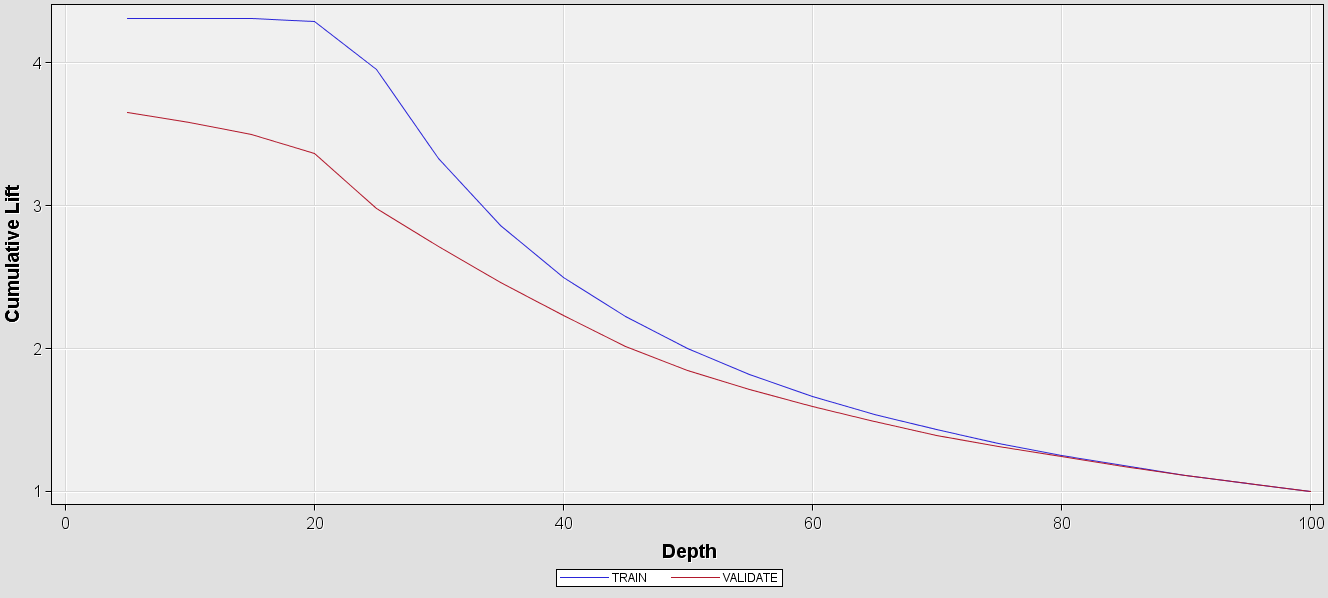
**Model Evaluation- Model Comparison & Improvement**

**Model Improvement through Ensemble Techniques**

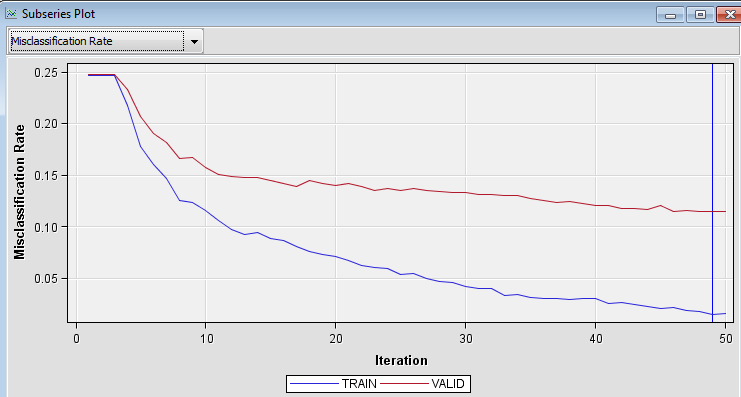
After comparing the two models before and after feature engineering, we attempted to improve the prediction even further using ensemble techniques. We chose a tree based technique called gradient boosting based on the decision tree performance in the preceding sections. As expected, the gradient boosting gave the maximum improvement in the model performance. All the default settings of the model were retained except the following:

1. The maximum depth was increased to 30. This was carefully tuned to avoid overfitting of data
2. Leaf fraction was set to 0.01. Gives the proportion of minimum samples required at the terminal leaf node. In a class imbalance problem, usually low leaf fraction values are preferred
3. Split size is increased to 20
4. Number of surrogate rules were adjusted to 6 to handle missing values in the trees

Cumulative lift of Gradient Boosting



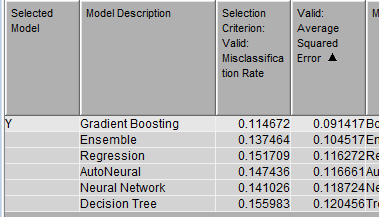
Subseries plot comparing the misclassification rates:



From the plots above, we can observe a good consistency in the model as there aren’t too many fluctuations in the misclassification rate over several iterations but has rather smooth reduction in error rates. One could argue that the training data appears to be over fitted, but the validation does not support that argument. It is also important to note that the model has not flat-lined on the error rate. Hence further increase in the number of iterations could lead to slight improvements in the prediction rates.

**Model Comparison**

We compared our models using model comparison node. We have used misclassification rate as selection statistics. SAS has chosen gradient boosting model as a selected model with 11.46% misclassification rate and 9.14% average squared error of validation data.



**References**

1. SAS Documentation on High Performance Procedures
2. Decision Trees for Predictive Modeling Padraic G. Neville SAS Institute Inc. 4 August 1999
3. Using Surrogates to Improve Datasets with Missing Values by Salford Systems
4. Feature Selection with Neural Networks Philippe LERAY and Patrick GALLINARI

**Tools used:**

1. SAS Enterprise Miner
2. SAS Base Programming
3. Rattle (R Studio)