–

By - Adagarby Bharath, Arora Prince, Jain Akash, Marathe Nikhil

IS7036-001-Group01

Optimizing telemarketing

of

term deposit subscriptions

through CRISP-DM

Abstract

Due to the global financial crisis, credit on international markets became more restricted for banks, turning attention to internal clients and their deposits to gather funds to expand their business. This driver led to a demand for knowledge about client’s behavior towards deposits and specially their response to telemarketing campaigns.

This project uses datamining approach to extract valuable knowledge from recent Portuguese bank telemarketing campaign data. This data can be used by managers who optimize telemarketing costs. They can use this knowledge to identify the customers willing to make term subscription a priori to making a call. Such approach was guided by the CRISP-DM methodology and the data analysis and model building was conducted using SPSS Modeler 18 and R Studio. Four classification models were tested viz. CHAID, Logistic Regression, Artificial Neural Networks and Support Vector Machine and are compared based on Specificity, Sensitivity, Precision, Accuracy, AUC and F-ratio. Overall, the **Ensemble ANN with Bagging** **Model** obtained the best results and final analysis was done on this model to gain insights, make recommendations and identify customers.

Contents

[Introduction 6](#_Toc475730217)

[1. Business Problem – BUSINESS UNDERSTANDING (CRISP - DM METHODOLOGY) 8](#_Toc475730218)

[2. DATA UNDERSTANDING (CRISP – DM Methodology): 8](#_Toc475730219)

[2.1 BoxPlots (Outlier Analysis) 10](#_Toc475730220)

[2.2 Histogram (Distribution analysis) 11](#_Toc475730221)

[3. DATA PREPARATION (CRISP – DM Methodology): 14](#_Toc475730222)

[4. MODELING (CRISP – DM METHODOLOGY): 16](#_Toc475730223)

[4.1 CHAID – Chi – Square Automatic Interaction Detector 16](#_Toc475730224)

[4.1.1 Background 16](#_Toc475730225)

[4.1.2 Sample the dataset – we use stratified sampling instead of random sampling. 17](#_Toc475730226)

[4.1.3 Create the model on the training dataset. Then test this trained model on the testing set. 18](#_Toc475730227)

[4.1.4 Enhance the model by using different parameters in SPSS. Perform model comparison. 20](#_Toc475730228)

[4.1.4.1 The model parameters that we will be using for model evaluation are as follows: 22](#_Toc475730229)

[4.1.5 Running the trained model on the testing data and the entire data to check how well the model fits the data. 23](#_Toc475730230)

[4.1.5.1 Testing Model 1: BASIC CHAID Model 24](#_Toc475730231)

[4.1.5.2 Testing Model 1: BOOSTING CHAID Model 24](#_Toc475730232)

[4.1.5.3 Testing Model 1: BAGGING CHAID Model 25](#_Toc475730233)

[4.2 Artificial Neural Network as a Classifier 26](#_Toc475730234)

[4.2.1 Background 26](#_Toc475730235)

[4.2.2 Structure of a Neuron 27](#_Toc475730236)

[4.2.3 Architecture of Neural Networks 27](#_Toc475730237)

[4.2.4 Learning Methods 28](#_Toc475730238)

[4.2.5 Ensemble Methods 28](#_Toc475730239)

[4.2.6 Artificial Network Performance on Bank Marketing Data 29](#_Toc475730240)

[4.2.6.1 Model Training 29](#_Toc475730241)

[4.2.6.2 Basic MLP Neural Network with Automatic Hidden Layer Computation 30](#_Toc475730242)

[4.2.6.3 Basic MLP Neural Network with two hidden layers of 5 Neurons each 31](#_Toc475730243)

[4.2.6.4 Ensemble MLP Neural Network with Automatic Layer Computation – Bagging Approach 32](#_Toc475730244)

[4.2.6.5 Ensemble MLP Neural Network with Automatic Layer Computation – Boosting Approach 35](#_Toc475730245)

[4.2.6.6 Basic RBF Neural Network with Automatic Hidden layers 38](#_Toc475730246)

[4.2.7 Model Testing 40](#_Toc475730247)

[4.2.7.1 Confusion Matrix of models 40](#_Toc475730248)

[4.3 Support Vector Machine as a Classifier 43](#_Toc475730249)

[4.3.1 Background 43](#_Toc475730250)

[4.3.2 Create the model on the training dataset. Then test this trained model on the testing set. 43](#_Toc475730251)

[4.3.2.1 SVM with Linear Kernel Function – 43](#_Toc475730252)

[4.3.2.2 SVM with Radial Basis function – 44](#_Toc475730253)

[4.3.2.3 SVM with Polynomial kernel function – 45](#_Toc475730254)

[4.3.2.4 SVM with Sigmoid kernel function – 46](#_Toc475730255)

[4.3.3 The model parameters that we will be using for model evaluation are as follows: 47](#_Toc475730256)

[4.3.4 Running the trained model on the testing data and the entire data to check how well the model fits the data. 47](#_Toc475730257)

[4.3.4.1 Testing Model 1: SVM with Linear kernel function 48](#_Toc475730258)

[4.3.4.2 Testing Model 2: SVM with RBF kernel function 49](#_Toc475730259)

[4.3.4.3 Testing Model 3: SVM with Polynomial kernel function 50](#_Toc475730260)

[4.3.4.4 Testing Model : SVM with Sigmoid kernel function 50](#_Toc475730261)

[4.3.5 Performance Metrics of trained models on testing data – 51](#_Toc475730262)

[4.4 Logistic regression 51](#_Toc475730263)

[4.4.1 Background 51](#_Toc475730264)

[4.4.2 Model building 52](#_Toc475730265)

[4.4.2.1 Forwards logistic regression: 52](#_Toc475730266)

[4.4.2.2 Backwards logistic regression: 53](#_Toc475730267)

[4.4.3 Analysis metrics: 53](#_Toc475730268)

[4.4.4 Training Model Analysis: 54](#_Toc475730269)

[4.4.4.1 Stepwise Logistic Model 55](#_Toc475730270)

[4.4.4.2 Forwards Logistic Model 56](#_Toc475730271)

[4.4.4.3 Backwards Logistic Model 58](#_Toc475730272)

[4.4.4.4 Backwards Stepwise Logistic Model 59](#_Toc475730273)

[4.4.4.5 Training Results Comparison: 60](#_Toc475730274)

[4.4.5 Testing Model Analysis: 61](#_Toc475730275)

[4.4.6 Stepwise Logistic Model 61](#_Toc475730276)

[4.4.7 Forwards Logistic Model 62](#_Toc475730277)

[4.4.8 Backwards Logistic Model 62](#_Toc475730278)

[4.4.9 Backwards Stepwise Logistic Model 63](#_Toc475730279)

[4.4.10 Testing Results Comparison: 64](#_Toc475730280)

[4.4.11 ROC Comparison 65](#_Toc475730281)

[4.4.12 Comparison of Testing and Training results: 65](#_Toc475730282)

[5. EVALUATION-(CRISP – DM Methodology) Recommendations 67](#_Toc475730283)

[Conclusion: 69](#_Toc475730284)

[REFERENCES: 69](#_Toc475730285)

Table of Figures

[Figure 1: CRISP-DM Model 7](#_Toc475723744)

[Figure 2: Boxplots 10](#_Toc475723745)

[Figure 3: Histograms 11](#_Toc475723746)

[Figure 4: Histograms\_1 12](#_Toc475723747)

[Figure 5: Histograms\_2 13](#_Toc475723748)

[Figure 6: BASIC CHAID MODEL ANALYSIS 19](#_Toc475723749)

[Figure 7: BOOSTING\_Analysis 21](#_Toc475723750)

[Figure 8: BAGGING\_Analysis 21](#_Toc475723751)

[Figure 9: BAGGING\_Analysis 22](#_Toc475723752)

[Figure 10: Model Evaluation 23](#_Toc475723753)

[Figure 11: BASIC\_Testing\_Analysis 24](#_Toc475723754)

[Figure 12: BOOSTING\_Testing\_Analysis 24](#_Toc475723755)

[Figure 13: BAGGING\_Testing\_Analysis 25](#_Toc475723756)

[Figure 14: Accuracy\_Model 1\_ANN 30](#_Toc475723757)

[Figure 15: Classification\_Model\_1\_ANN 31](#_Toc475723758)

[Figure 16: Accuracy\_Model\_2\_ANN 32](#_Toc475723759)

[Figure 17: Classification\_Model\_2\_ANN 32](#_Toc475723760)

[Figure 18: Accuracy\_Model\_3\_ANN 33](#_Toc475723761)

[Figure 19: Predictor\_Frequency\_Model\_3\_ANN 34](#_Toc475723762)

[Figure 20: Component\_Details\_Model\_3\_ANN 34](#_Toc475723763)

[Figure 21: Accuracy\_Model\_4\_ANN 36](#_Toc475723764)

[Figure 22: Predictor\_Frequency\_Model\_4\_ANN 36](#_Toc475723765)

[Figure 23: Component\_Details\_Model\_4\_ANN 37](#_Toc475723766)

[Figure 24: Ensemble\_Accuracy\_Model\_4\_ANN 38](#_Toc475723767)

[Figure 25: Accuracy\_Model\_5\_ANN 39](#_Toc475723768)

[Figure 26: Classification\_Model\_5\_ANN 39](#_Toc475723769)

[Figure 27: ROC Curves Comparison 42](#_Toc475723770)

[Figure 28: Training analysis on SVM - Linear 44](#_Toc475723771)

[Figure 29:Training analysis on SVM - RBF 45](#_Toc475723772)

[Figure 30: Training analysis on SVM - Polynomial 46](#_Toc475723773)

[Figure 31: Training analysis on SVM - Sigmoid 47](#_Toc475723774)

[Figure 32: ROC curve of different models on testing data 48](#_Toc475723775)

[Figure 33: Testing data analysis on SVM - Linear 49](#_Toc475723776)

[Figure 34: Testing data analysis on SVM - RBF 49](#_Toc475723777)

[Figure 35: Testing data analysis on SVM - Polynomial 50](#_Toc475723778)

[Figure 36: Testing data analysis on SVM - Sigmoid 50](#_Toc475723779)

[Figure 37: Stepwise\_Logical\_Model\_Analysis 55](#_Toc475723780)

[Figure 38: Forwards Logistic Model\_Analysis 56](#_Toc475723781)

[Figure 39: Backwards Logistic\_Analysis 58](#_Toc475723782)

[Figure 40: Backwards\_Stepwise\_Logistic\_Analysis 59](#_Toc475723783)

[Figure 41: Stepwise\_Logical\_Analysis 61](#_Toc475723784)

[Figure 42: Forwards\_Logistic\_Analysis 62](#_Toc475723785)

[Figure 43: Backwards\_Logistic\_Analysis 63](#_Toc475723786)

[Figure 44: Backwards\_Stepwise\_Logistic\_Analysis 64](#_Toc475723787)

[Figure 45: ROC Curve Comparison 65](#_Toc475723788)

[Figure 46: ROC Curve Comparison for Recommendation 68](#_Toc475723789)

Table of Tables

[Table 1 : Data Dictionary 10](#_Toc475723954)

[Table 2: Model Comparison 23](#_Toc475723955)

[Table 3: Model Comparison\_Testing 25](#_Toc475723956)

[Table 4: Confusion\_Matrix\_Model\_3\_ANN 35](#_Toc475723957)

[Table 5: Confusion\_Matrix\_Model\_4\_ANN 38](#_Toc475723958)

[Table 6: Confusion\_Matrix\_Comparison 41](#_Toc475723959)

[Table 7: Model Evaluation\_Criteria\_ANN 41](#_Toc475723960)

[Table 8: Model Evaluation Criteria\_AUC 42](#_Toc475723961)

[Table 9:Model Comparison (training data) 47](#_Toc475723962)

[Table 10: Model Comparison (testing data) 51](#_Toc475723963)

[Table 11: Step Summary 55](#_Toc475723964)

[Table 12: Model Fitting Information 56](#_Toc475723965)

[Table 13: Step\_Summary 57](#_Toc475723966)

[Table 14: Model Fitting Information 57](#_Toc475723967)

[Table 15: Model Fitting Information 58](#_Toc475723968)

[Table 16: Backwards\_Stepwise\_Logistic\_Summary 59](#_Toc475723969)

[Table 17: Training\_Results\_Comparison 60](#_Toc475723970)

[Table 18: Testing\_Results\_Comparison 64](#_Toc475723971)

[Table 19: Comparison of Training and Testing Results 66](#_Toc475723972)

[Table 20: Recommendation\_Analysis 67](#_Toc475723973)

# **Introduction**

In recent times, financial crisis has been prevalent all around the world and distrust on banks becomes an important issue to consider for these institutions. Skepticism leads to withdraws, latent investment wait for more optimistic scenarios, and credit becomes restricted not only to individual clients and companies, but also between financial institutions.

Such a context constitutes by its own a huge driver that potentiates a pursuit for efficiency. The global economic crisis that emerged in 2007 in the United States and spread world-wide, affecting Europe is paving its way by triggering new ideas about financial management and new thoughts and points of view (Hodgson, 2009).

Considering the recent effects of the crisis on Europe and US, one consequence for the banks is the credit restriction, which led to competition for **client’s deposits**. These drivers led retail banks to invest in lucrative products and campaigns to gather and retain financial assets by selling long-term deposits and prime and sub-prime mortgages to clients by taking advantage of their profiles

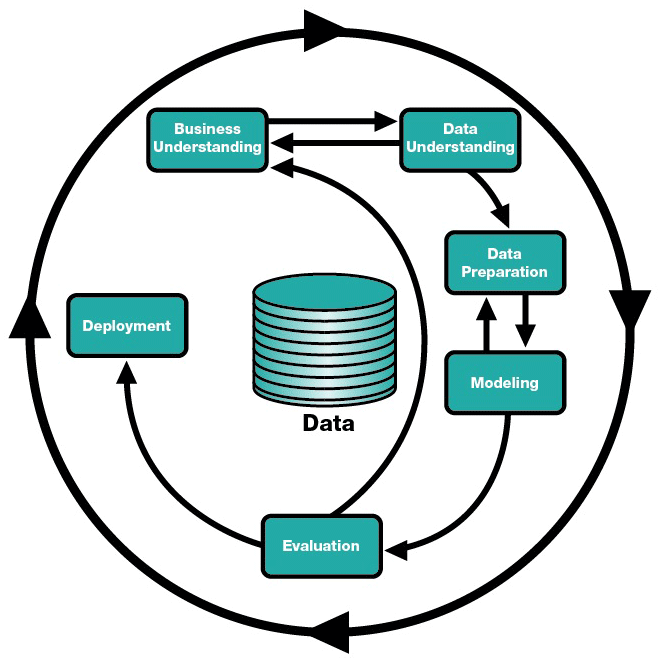
The main approach for banks to promote products/services to clients, are through mass campaigns or direct campaigns, targeting a specific set of clients. But the positive responses to these campaigns has been very less and which leads to ineffective use of resources and workforce.

Telemarketing (sometimes known as inside sales, or telesales in the UK and Ireland) is a method of direct marketing in which a salesperson solicits prospective customers to buy products or services, either over the phone or through a subsequent face to face or Web conferencing appointment scheduled during the call. Telemarketing can also include recorded sales pitches programmed to be played over the phone via automatic dialing (Wikipedia).

With the automation of telemarketing through the integration of computers and analytics in Telephony, it became quite common and easy to generate a wide variety of reports from other marketing campaigns and gaining insights by adding other types of information available to organizations about the prospective clients.

One effective way to identify patterns in this huge amount of data, to gain knowledge and use this knowledge in business to gain wisdom, is through Data mining and Business Intelligence techniques. These techniques build data-driven models to extract useful patterns which then can be transformed to knowledge and eventually to wisdom.

A DM project encompasses all the steps needed to extract useful knowledge and apply it to produce benefit by improving the ROI (Return on Investment). We have used Cross-Industry Standard Process for data mining (CRISP-DM Methodology) which defines a non-rigid sequence of six phases, which allows the building and implementation of a data mining model to be used in a real environment, helping to support business decisions. It defines the project as a cyclic process, where several iterations are conducted to turn Data Mining towards business goal.



*The CRISP-DM Process Model*

Figure : CRISP-DM Model

# **Business Problem – BUSINESS UNDERSTANDING (CRISP - DM METHODOLOGY)**

A Portuguese retail bank uses its own contact-center to do direct marketing campaigns, mainly through phone calls (telemarketing). Each campaign is managed in an integrated fashion and the results for all calls and clients within the campaign are gathered together, in an excel file report concerning only the data used to do the phone call. The agents were all human, thus no automatic calls through Interactive Voice Response (IVR) or Voice Response Unit (VRU) were performed.

The primary goal is to find previously undiscovered valuable knowledge to redirect managers’ efforts to improve campaign results. In other words, the objective was, on the one hand, decrease the number of phone calls (efficiency dimension – cost reduction) and, on the other hand, increase the total number of term deposits subscriptions.

# **DATA UNDERSTANDING (CRISP – DM Methodology):**

The data source that we are using for this project is taken from the UCI Machine Learning Library. This dataset is publicly available for performing classification studies.

**“Citation:**

[Moro et al., 2011] S. Moro, R. Laureano and P. Cortez. Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology.

In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference - ESM'2011, pp. 117-121, Guimarães, Portugal, October, 2011. EUROSIS”.

The data is basically related directly to the telemarketing campaign of a banking institution in portugal. The marketing campaing was based mostly on phone calls to the clients, in order to convince them to opt for term deposits. The dataset also gives us information on whether more than one contact was required with a particular client, in order to convince the client to opt for the term deposit. The purpose of this study is to mainly analyze the factors concerning the client that are considered when he/she accepts the term deposit. Also, classification analysis on this dataset can help us to identify the mistakes that the bank is making in terms of their telemarketing camapaign, and also, the kind of clients and the form of communication on which to focus, in order to have a higher rate of conversions to accept the term deposit.

In order to perform the analysis, it is essential to understand the fields that we have in the dataset.

Following is a brief description on the fields that are used as a part of this study.

|  |  |
| --- | --- |
| **Field** | **Description** |
| age(numeric) | Age of the client contacted |
| marital(categorical) | |  | | --- | | Type of job -(categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student", "blue-collar","self-employed","retired","technician","services") | | |
| Job(categorical) | Marital Status - ("married","divorced","single"; note: "divorced" means divorced or widowed) |
| education(categorical) | Highest Education - ("unknown","secondary","primary","tertiary") |
| default(categorical) | if the client has credit in default? ( "yes","no") |
| balance(numeric) | average yearly balance, in euros |
| housing(categorical) | if the client has housing loan - ("yes","no") |
| loan(categorical) | if the client has personal loan - ("yes","no") |
| contact(categorical) | channel through which contact was communication - ("unknown","telephone","cellular") |
| day(numeric) | last contacted day of the month |
| month(categorical) | last contacted month of year - ("jan", "feb", "mar", ..., "nov", "dec") |
| duration(numeric) | last contact duration, in seconds |
| campaign(numeric) | number of contacts performed during this campaign and for this client - (includes last contact) |
| pday(numeric) | number of days that passed by after the client was last contacted from a previous campaign - (1 means client was not previously contacted) |
| previous(numeric) | number of contacts performed before this campaign and for this client |
| poutcome(categorical) | outcome of the previous marketing campaign - ("unknown","other","failure","success") |

Table : Data Dictionary

The next step in the CRISP – DM Methodology after Business Understanding is the Data Understanding part.

Here we try to understand the dataset that we are using, as this data understanding can help us in getting ready for the data preparation stage where in we initially clean the data for further analysis.

The best method for Data Understanding is performing the EDA(Exploratory Data Analysis) on the given data. Following are some of the methods that we will be using for performing EDA.

## **BoxPlots (Outlier Analysis)**

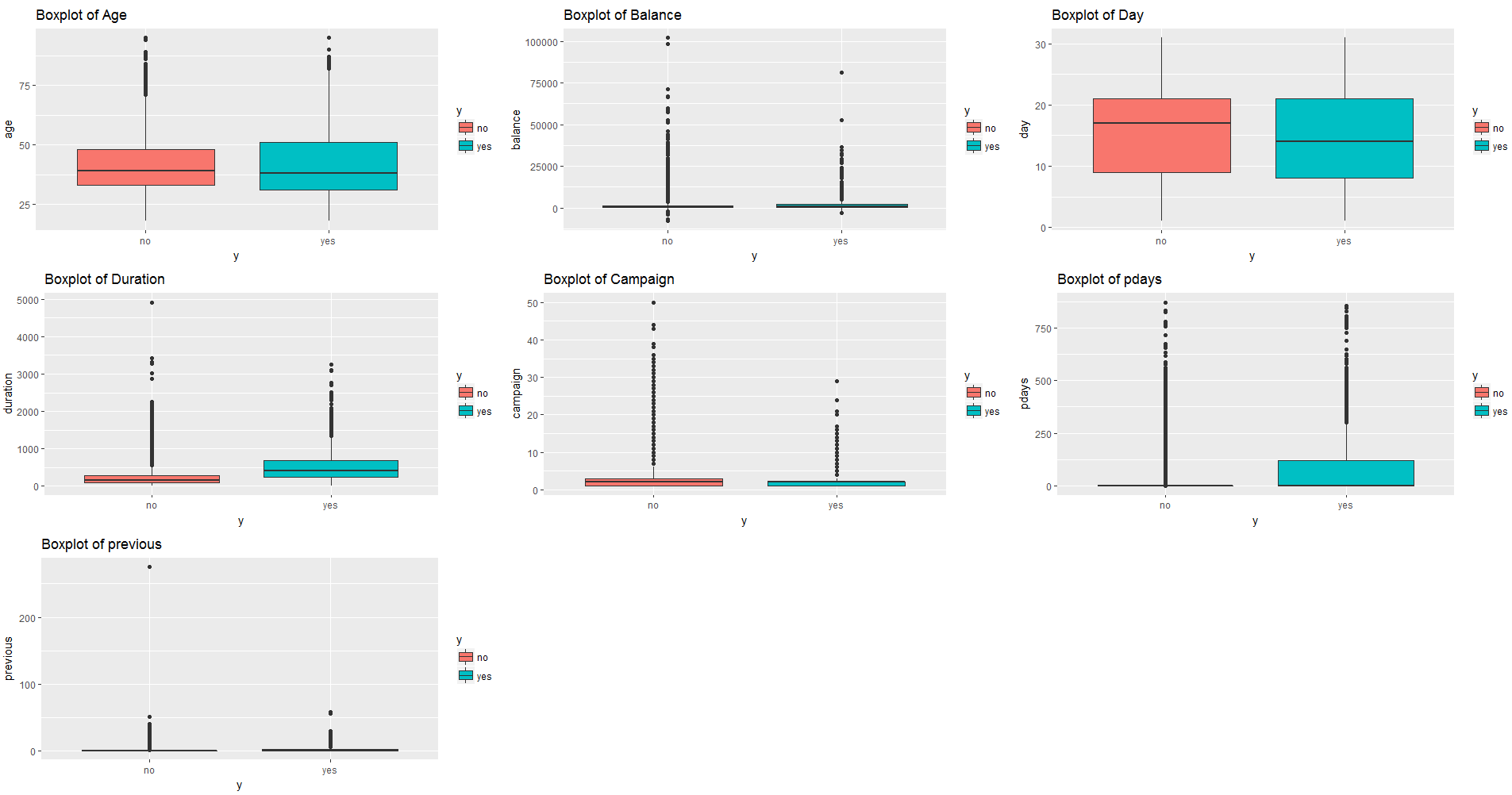
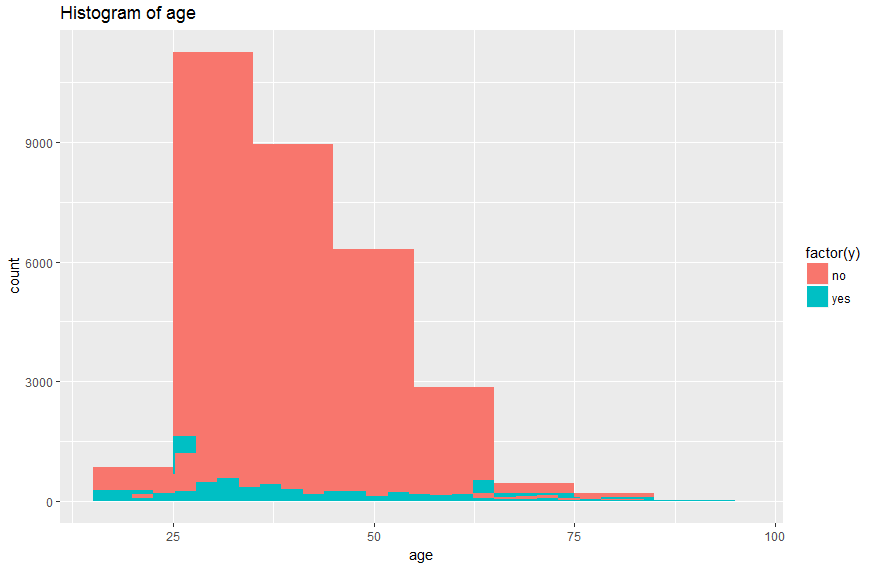
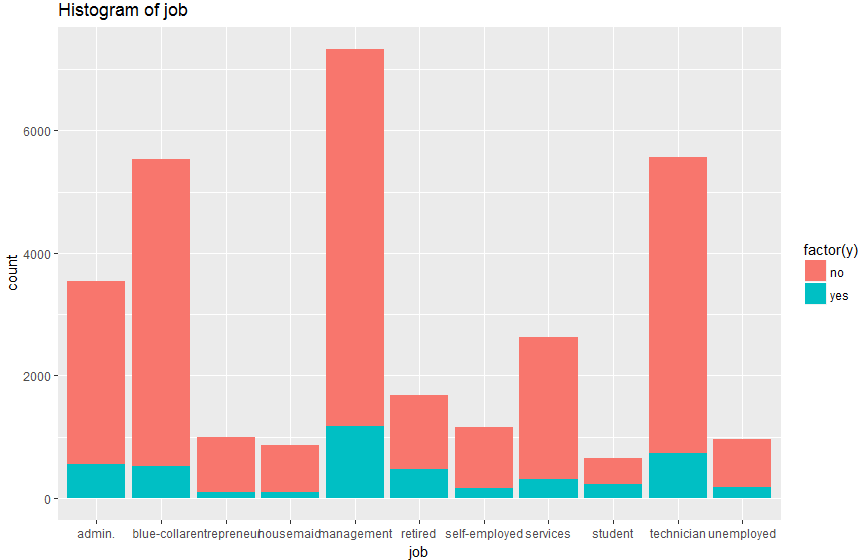
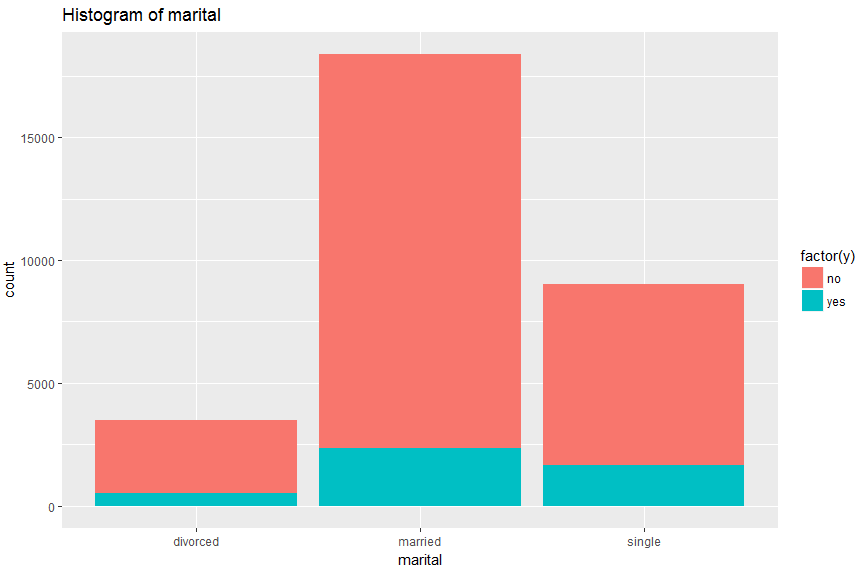
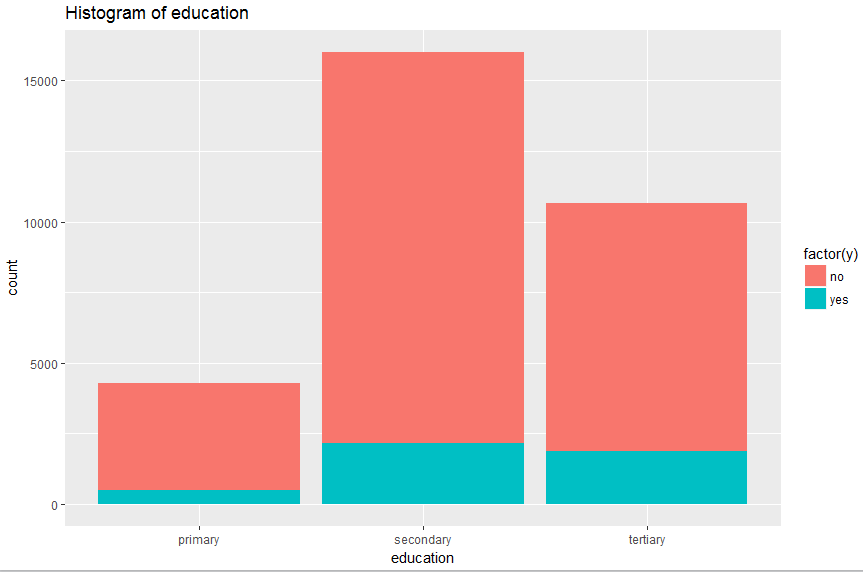


Figure : Boxplots

From the above boxplots, we can see that there are a few outliers in the given data. Such outliers can play a significant part in affecting the classification analysis that we will be performing later. The pdays and balance variable have the most number of outliers, as we can see from the boxplots for the variables.

## **Histogram (Distribution analysis)**

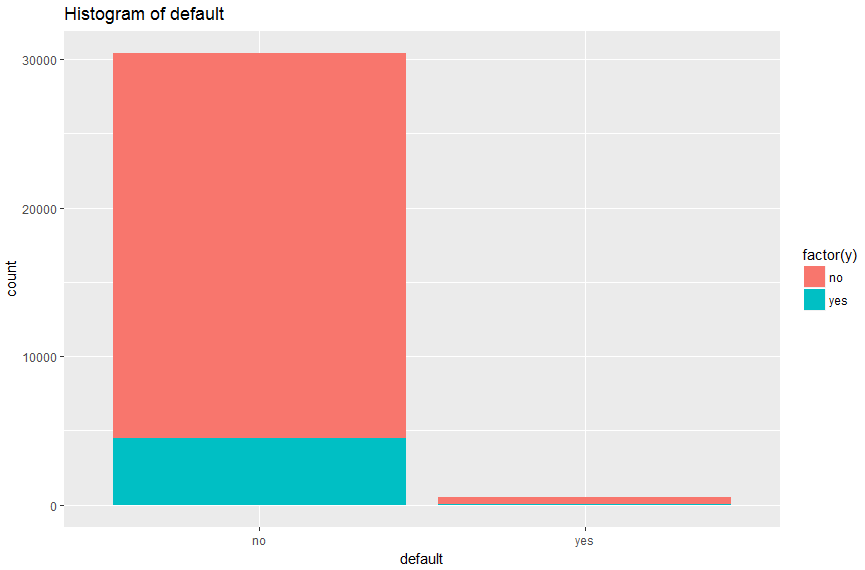
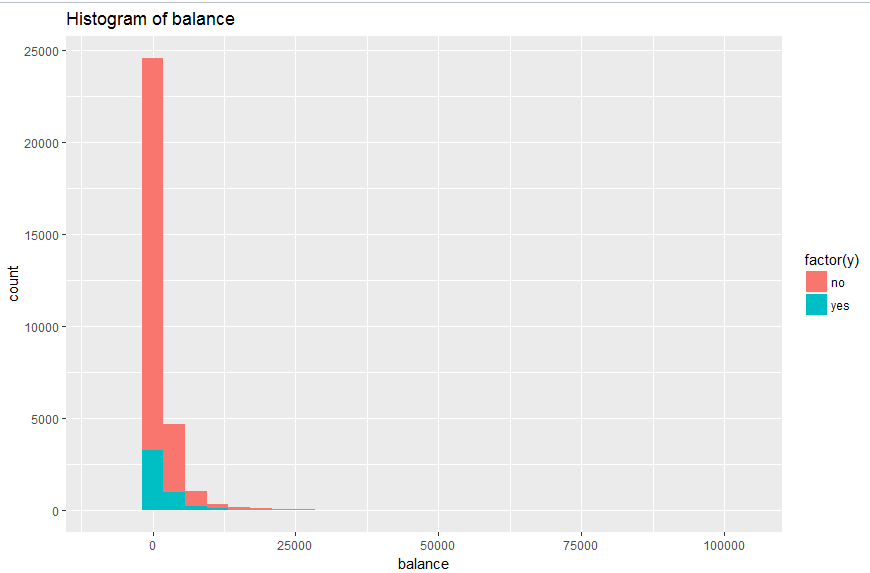
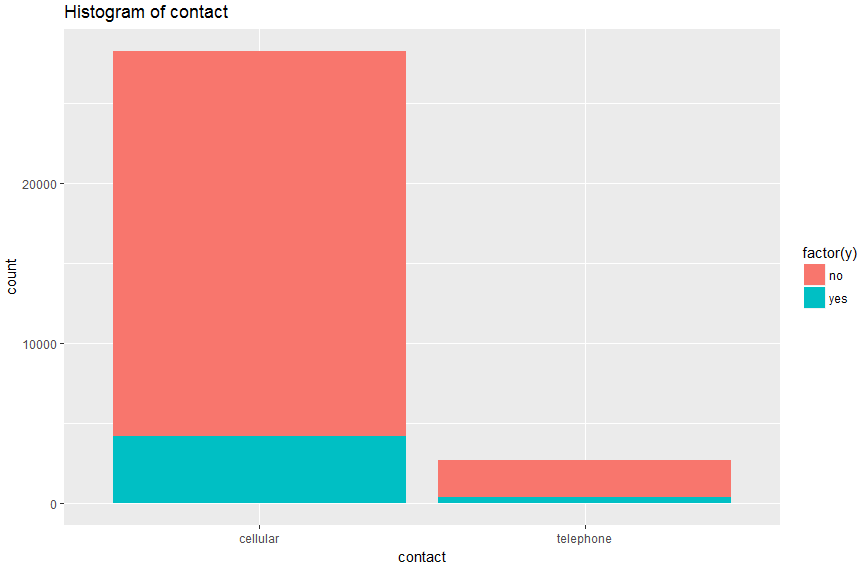
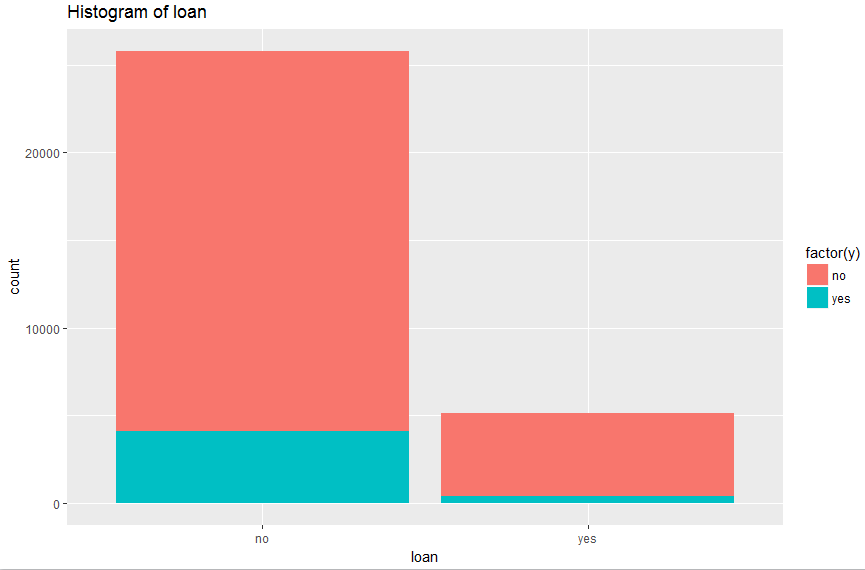
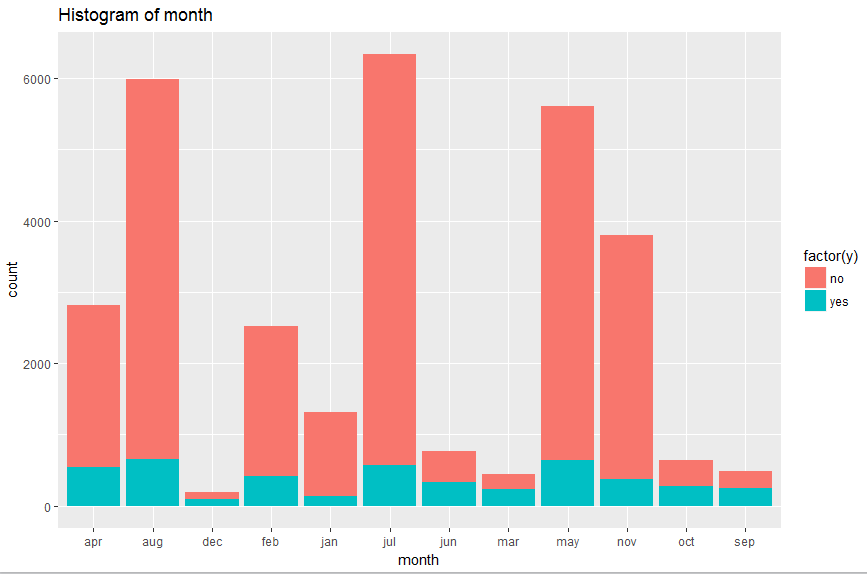
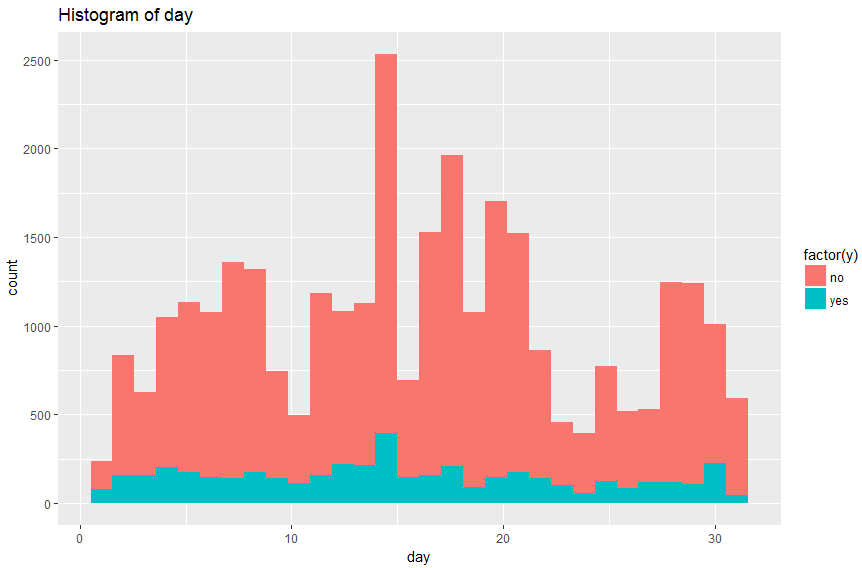
 

Figure : Histograms

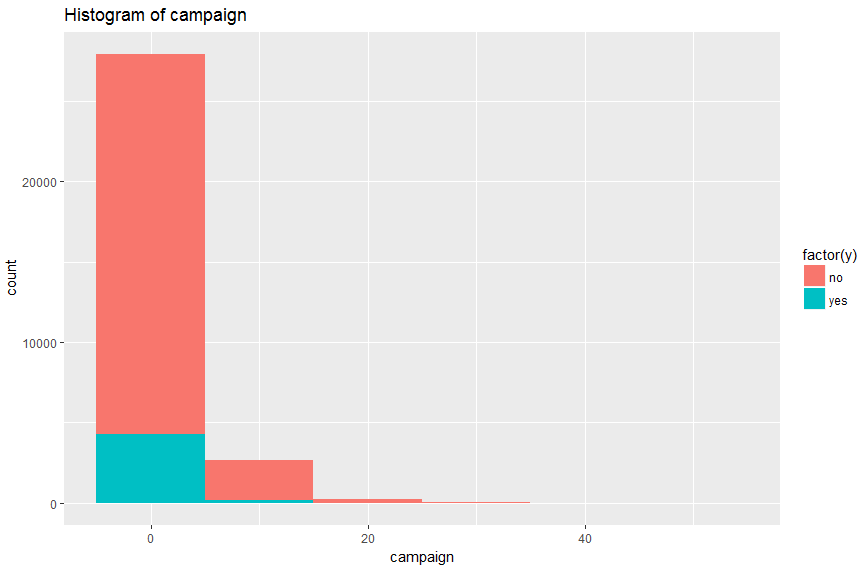
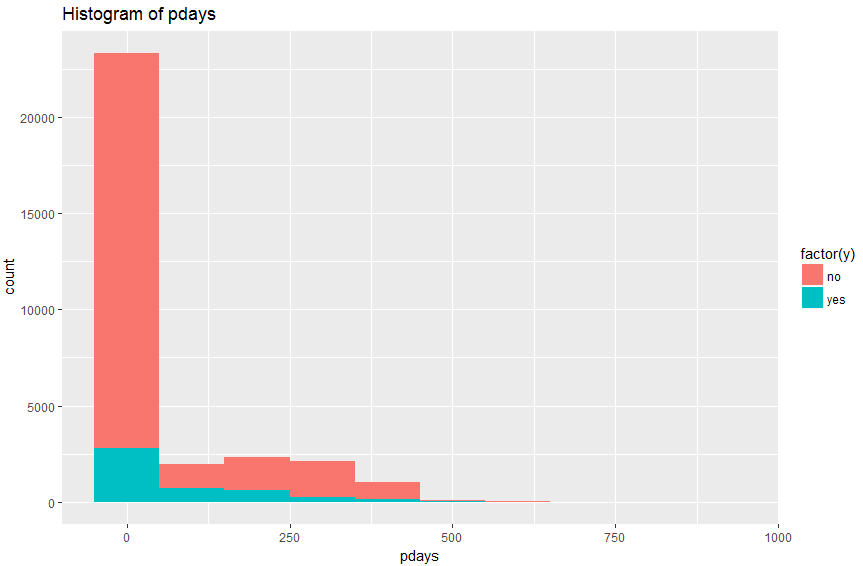
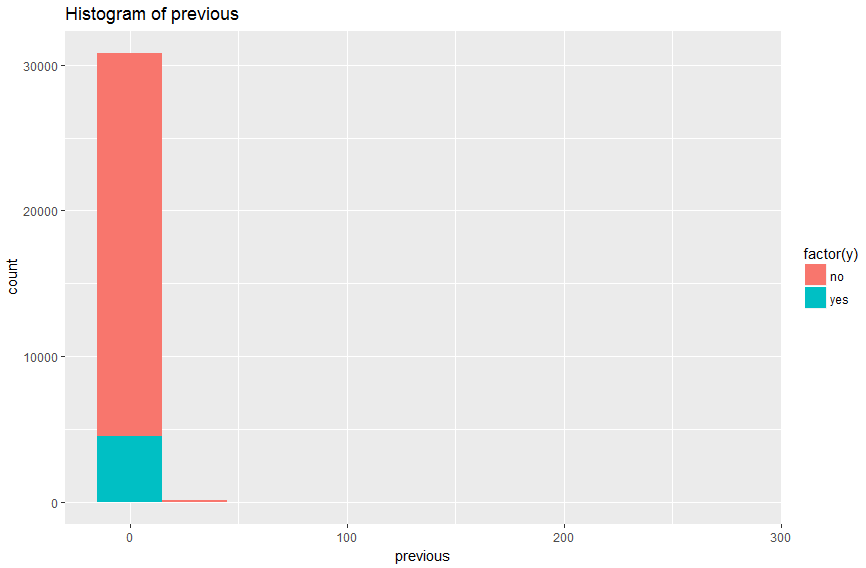
 

Figure : Histograms\_1

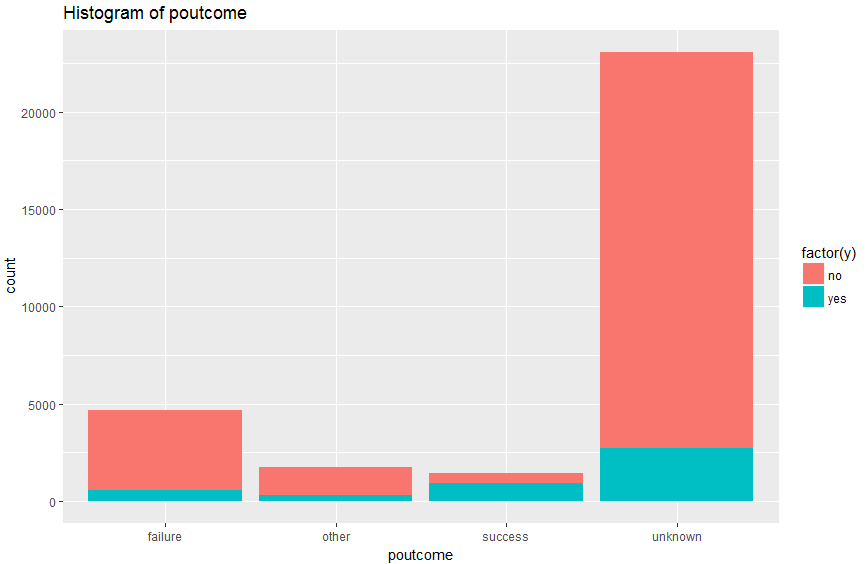


Figure : Histograms\_2

* The above Histograms for all the variables show the distribution of the variable values against the parameter ‘y’ – our response variable for this study. Looking at the histograms above in depth, we can see that there is not one single variable which is significantly affecting the outcome of the response variable. thus, when we will be building our models for final analysis, we will mostly have to include all the variables in the models.
* One variable that stands out in this case is the ‘duration’ field. The outcome of our study is to predict whether a client will opt for the term deposit or not. For this purpose, we will not necessarily know the duration of the call that the telemarketer would be making to the client. Some records have a duration value as 0, and so the response variable is ‘no’. We will not consider the duration field for the model building because of this reason.
* The number of records which have response variable as ‘no’ is almost 85% of the total number of records, and as such we will use stratified sampling for model creation.

# **DATA PREPARATION (CRISP – DM Methodology):**

The main step in the entire CRISP – DM Methodology, is the data preparation. An efficiently prepared dataset can help in easier model creation and analysis. Data preparation takes the most time in terms of the entire process of the CRISP – DM Methodology.

Since we are using this dataset from the UCI Machine Learning repository, this dataset is mostly clean, but still we are performing some data preparation techniques on this dataset. Since this is a telemarketing data, one of the main variables that we must analyze initially is the contact variable, which gives details about the form of communication used to connect with the client. There are 3 different categories for this variable, “telephone”, “cellular” and “unknown”. However, we are not using the “unknown” records for this analysis, as it does not make sense to include those records for which the contact information is unknown. We should reach a conclusion which says which form of communication is preferred or is more likely to turn the response of the client to the opting of a term deposit to ‘yes’. As result, we have cleaned or removed all the records from the dataset for which we have contact information as unknown. Thus, the final count of records that we will be using for analysis is around the 31000 mark.

From the EDA in the earlier step, we can see that almost 85% of the records in the dataset have the value of the response variable ‘y’ as ‘NO’. We normally do the analysis by segregating the dataset into two parts, ‘training’ and ‘testing’ parts. However, if we do opt for this method of partitioning, there is a high probability that in the 70% randomly sampled training set that we get, we might end up having almost all the records with the response variable as ‘no’. Or for that matter, if we consider any random sampling for training, the maximum percentage of response variable values as ‘yes’ in this training set is going to be around the 15% mark.

As a result, we cannot perform random sampling on the dataset, and we will be performing **stratified sampling** on the given dataset. Stratified sampling involves the division of a population into smaller groups known as strata. This method of sampling is like the concept of a weighted average, in which the sampling produces characteristics that are proportional to the overall population. We then consider training and testing partitions from the stratified sampled dataset, and then we extend these built models to the original dataset to see how well this build model works on the entire dataset. Stratified sampling helps us in assigning equal weights to the different categories in the dataset, in cases where the categories for a particular response variable are right or left skewed to the extreme.

In the next stage of the CRISP – DM Methodology of Model Building, we are building 4 different models based on different algorithms as given below:

* CHAID Algorithm
* Artificial Neural Network (ANN)
* Support Vector Machine (SVM)
* Logistic Regression

The model building and analysis is described in the next part of the CRISP-DM Methodology, **‘Modeling’**.

# **MODELING (CRISP – DM METHODOLOGY):**

After completing the first three steps of the CRISP-DM Methodology, we will now proceed to the next step of MODELING.

For this study, we will create 4 models based on different algorithms as follows:

* CHAID Algorithm
* Artificial Neural Network (ANN)
* Support Vector Machine (SVM)
* Logistic Regression

The creation and analysis of these models is explained as follows:

## **CHAID – Chi – Square Automatic Interaction Detector**

## **Background**

The algorithm that we will be considering for this part of the study is the CHAID Algorithm in SPSS Modeler. CHAID – Chi-Square Automatic Integration Detector, is a classification method for building decision trees by using Chi-Square statistics to identify optimal splits. The Chi-Square statistics, instead of the impurity measures such as GINI or Entropy, is used by this algorithm to determine the optimal splits for the decision trees. The one advantage that CHAID has over other decision tree models is that it can generate non-binary trees. This helps in the analysis as we do not necessarily have to group more than 2 categories of a variable into a single partition. Each category is split differently which is a more effective way of predicting the accuracy of a model and by extension, the dataset in general. The algorithm categorizes all the variables that are used as the input for the algorithm, be it continuous, nominal or ordinal. The way CHAID works is that it tries to find the best explanatory variable and the best merger of categories at each node. CHAID starts the classification by creating the first “layer” of branches by displaying values of the dependent variable. CHAID determines this split automatically and how many categories to split this variable into. There are more than 2 categories for some variables in the dataset andso CHAID is a good choice for the decision trees.

As the name suggests, CHAID uses the chi-square statistic for classification as opposed to GINI or ENTROPY for CART or C4.5 respectively. A chi square (X2) statistic is used to investigate whether distributions of categorical variables differ from one another. There are basically two types of random variables and they yield two types of data: numerical and categorical. Basically, categorical variable yield data in the categories and numerical variables yield data in numerical form. If responses to questions are say “yes” or “no” or different categories, then such variables are known as categorical variables. In contrast, if responses to questions are pure numbers, for. E.g height or the distance travelled, then such variables are numerical. Numerical data can be either discrete or continuous. The Chi Square statistic compares the tallies or counts of categorical responses between two (or more) independent groups. Chi square tests can only be used on actual numbers and not on percentages, proportions etc.

**The analysis is done as follows:**

1. **Sample the dataset – we use stratified sampling instead of random sampling.**
2. **Create the model on the training dataset. Then test this trained model on the testing set.**
3. **Enhance the model by using different parameters in SPSS. Perform model comparison.**
4. **Use this most optimum model and run it on the test data and original dataset (non sampled data) and see how the model performs on this dataset.**

## **Sample the dataset – we use stratified sampling instead of random sampling.**

From our initial EDA, we observed that there is no one specific variable that is significantly determining the outcome or response variable. Also, we observed that from the given dataset that there are only around 85-90% records which have the response variable ‘y’ as ‘yes’. Because of this discrepancy, random sampling is not a good idea as if we do consider a random sample with 70% and 30% training and testing data respectively, we might end up getting all the records with response variable as ‘no’ and as such model might be the best example of under or over fitting. To overcome this problem, we perform stratified sampling, wherein we divide the dataset into different equivalent stratas and then separate them into 70% and 30% training and testing datasets and perform model analysis. (**IS7036-001-Group01\_ProjectReport\_CHAID.str – Part 1**).

## **Create the model on the training dataset. Then test this trained model on the testing set.**

* **Basic Setting:**
* Maximum Tree Depth: Default (5)
* Stopping Rules: Use of Percentage%
  + Min Records in Parent Branch – 5
  + Min Records in Child Branch – 2
* Ensembles
  + Default combining rule for categorical targets: Voting
  + Default combining rule for continuous targets: Mean

The training set, generated from the sampled data above, is then passed through the CHAID Algorithm for classification. For the initial analysis, we will be using the default settings for the CHAID algorithm.The summary analysis and the tree viewer for the BASIC CHAID Algorithm suggests that the root node used for the initial analysis is the poutcome node. The summary for the model also gives an indication as to which is the predictor variable that is affecting the classification the most. From the summary, we can also see the effect of the other predictors in the dataset **(Appendix\_Figure 1)**.

As we can see from the nodes in the tree, the poutcome variable is split into multiple child nodes, and not just two, as is the case with the binary tree classification. This tree gives the classification for the response variable based on the poutcome node which is the root node. This suggests that, the main factor affecting the response variable over here, that is, whether a client will opt for the term deposit depends on the outcome for this particular client from a previous similar telemarketing campaign **(Appendix\_Figure 2)**. The analysis node, applied to the model created, gives the misclassification matrix, giving us the idea about how many records have been misclassified as yes or no. The accuracy that we got for this model is about 70%, which is a good enough measure of how fit the generated model is. **(Figure 6)**

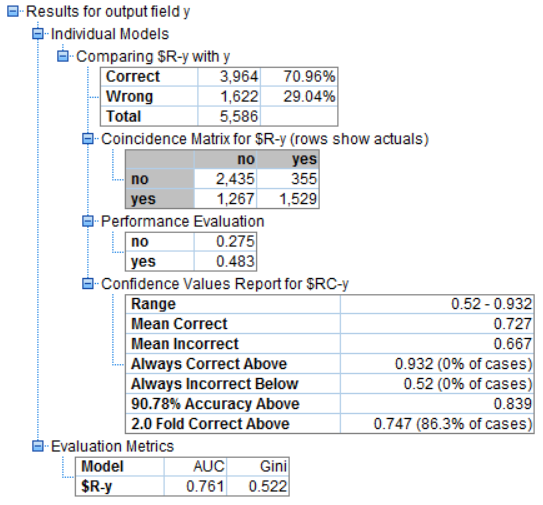


Figure : BASIC CHAID MODEL ANALYSIS

From this matrix, we can also see the sensitivity and the specificity measures for the generated model. The model also provides the AUC values, as this is a classification tree. **The AUC is obtained by plotting the FPR(x-axis) vs TPR(y-axis) graph**. **The True Positive Rate(TPR), also known as Sensitivity or Recall or hit rate** is the proportion of yes points that are correctly considered as yes, with respect to all the yes points. The higher this rate, the fewer positive data points we will miss. **The False Positive Rate(FPR) also known as (1 – Specificity)** is the proportion of negative data points that are mistakenly considered as positive with respect to the all the negative points, with respect to all the negative data points. The higher the FPR, the more negative points we will misclassify. From the **TPR, FPR and Accuracy**, we can say that the generated model is a good fit for the given training dataset. However, this is just the BASIC CHAID model that we are testing. We will be enhancing the model further and checking the variations in the classification because of this enhancement.

## **Enhance the model by using different parameters in SPSS. Perform model comparison.**

* **Basic Setting:**
* Enhance model accuracy: Boosting / Bagging
* Maximum Tree Depth: Default (5)
* Stopping Rules: Use of Percentage%
  + Min Records in Parent Branch – 2
  + Min Records in Child Branch – 1
* Ensembles
  + Default combining rule for categorical targets: Voting
  + Default combining rule for continuous targets: Mean

Bootstrap aggregating (Bagging) and boosting are algorithms used to improve model stability and accuracy. Bagging works well for unstable base models and can reduce variance in predictions. Boosting can be used with any type of model and can reduce variance and bias in predictions. These are just model enhancements methods, to see if the bootstrapping can have any impact on the overall model performance. They combine multiple models into one usually more accurate than the best of its components.

We have built two different models for boosting and bagging as can be seen in (**IS7036-001-Group01\_ProjectReport\_CHAID.str – Part 2**). We are still creating our models on the training data, and then we will apply all these models, the BASIC and the bootstrapped models on the testing data to see if we are facing the issues of overfitting or underfitting. Ensemble combines multiple models into one usually more accurate than the best of its components. The ensemble model displays the accuracy of the final model, compared to a reference model and a naïve model. The best model is supposed to have the best accuracy. Since we are doing a classification on a categorical variable, the accuracy is simply the % of records for which the predicted value matches the observed data. When we consider booting ensemble, the standard model that is built on the whole training part is the reference model. For bagging ensembles, the first component model that is built is the reference model. If no model were built, then this accuracy is given by the naïve model. All these models can be seen in the graph **(Appendix\_Figure 3, 4, 5)**. The analysis for the boosting model is as can be seen in the figure below. **(Figure 7)**

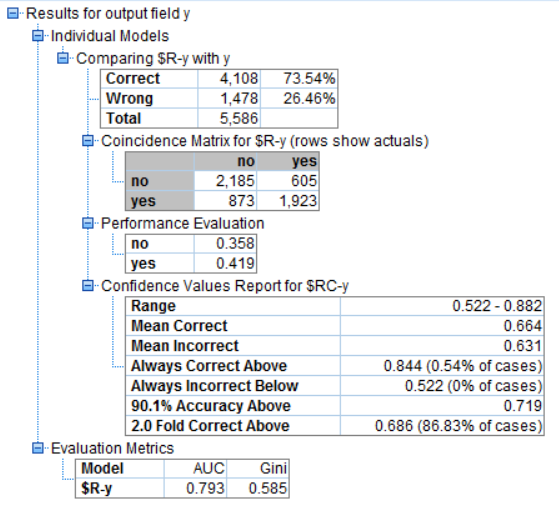


Figure : BOOSTING\_Analysis

Similarly, we have built the bagging model **(Appendix\_Figure 6 and 7)**. The analysis for the bagging model is as seen in the figure below (Figure 8)**.**

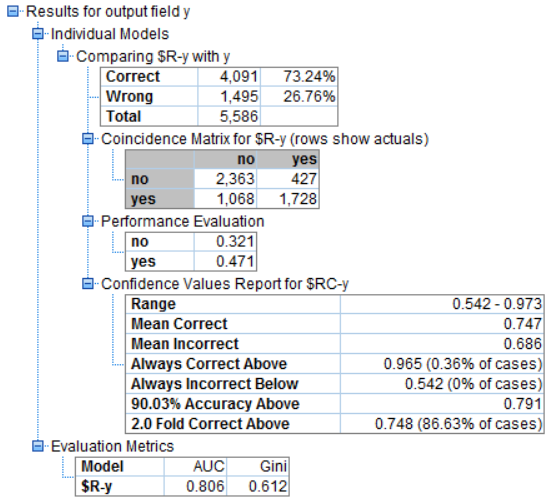


Figure 8: BAGGING\_Analysis

We have also run the exhaustive CHAID model for this analysis. However, from the output for this model, we can see that it is exactly same as the BASIC CHAID model. **(Figure 9)**

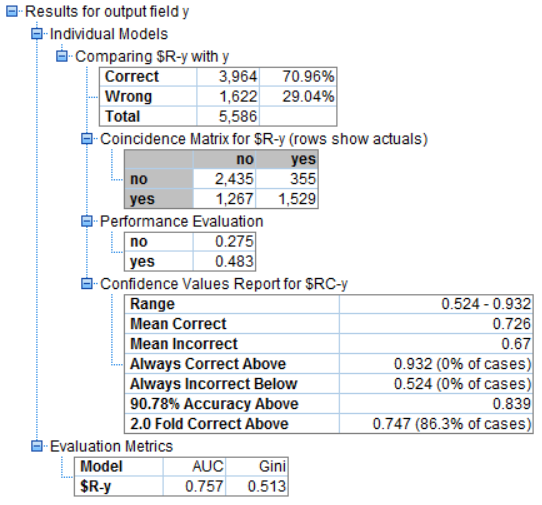


Figure 9: BAGGING\_Analysis

So, we won’t be considering this model on the training set. The analysis for all the three models are compared in the table shown. **(Table 2: Model Comparison).**

So, to summarize the steps till now, we have built three different models, basic and enhanced models, making use of the CHAID Algorithm. The comparison table tells us the best model as ‘ ‘ which models the training data set. However, we still should check how this trained model works on our testing data and on the dataset in the end. (**IS7036-001-Group01\_ProjectReport\_CHAID.str – Part 2**).

### **The model parameters that we will be using for model evaluation are as follows:**

* Sensitivity/Recall
* Specificity
* Precision
* Area under the ROC curve(AUC)
* Accuracy
* F-ratio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Sensitivity** | **Specificity** | **Precision** | **Accuracy** | **AUC** | **F-Ratio** |
| **BASIC** | 54.68526 | 87.27599 | 81.15711 | 70.96312 | 0.761 | 65.34188 |
| **BOOSTING** | 68.77682 | 78.31541 | 76.06804 | 73.541 | 0.793 | 72.23892 |
| **BAGGING** | 61.80258 | 84.69534 | 80.18561 | 73.23666 | 0.806 | 69.80408 |
| **Exhaustive\_CHAID** | 54.68526 | 87.27599 | 81.15711 | 70.96312 | 0.761 | 65.34188 |

Table : Model Comparison

## **Running the trained model on the testing data and the entire data to check how well the model fits the data.**

We will now run all the three models created in the above step, to run them on the test data set to see how well the trained model fits the testing data. The parameters that we will check for this purpose are the accuracy, sensitivity, specificity, F-ratio, Precision and AUC (Area Under the Curve). A comprehensive comparison among all the three models is also shown as part of this study.

A table of comparison for all the three models can help us analyze how well the trained models fit the test data **(Table 3)**. All these models are then passed through a comparative test to check which model is the best fit, and then we use that to analyze how well this ‘best chosen’ model fits the overall data**.** The comparative analysis for all the three models in the end give us a comparison for the ROC Curves for all the three models and subsequently their respective AUC’s **(Figure 10)**. (**IS7036-001-Group01\_ProjectReport\_CHAID.str – Part 3**).

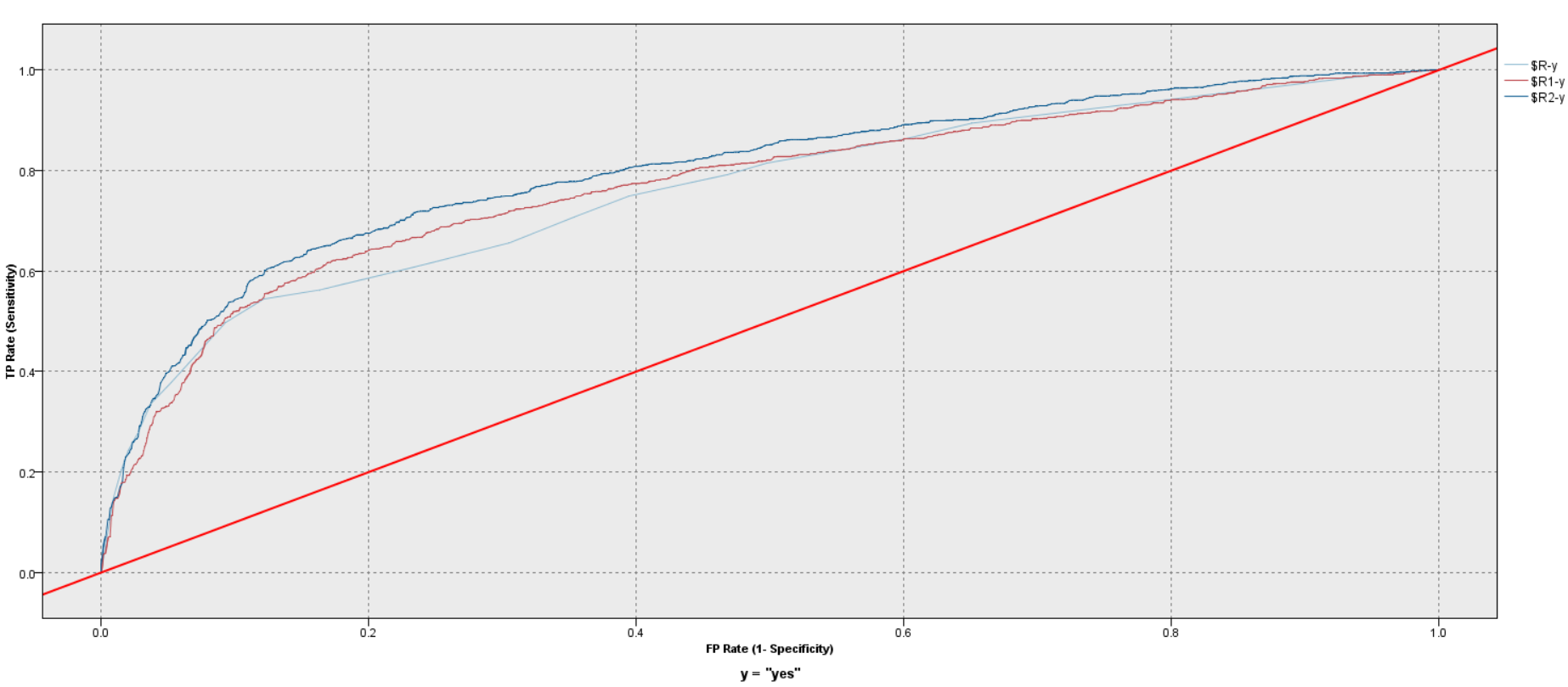


Figure : Model Evaluation

### **Testing Model 1: BASIC CHAID Model**

The analysis after running the BASIC CHAID model on the training set is as follows: **(Figure: 11)**

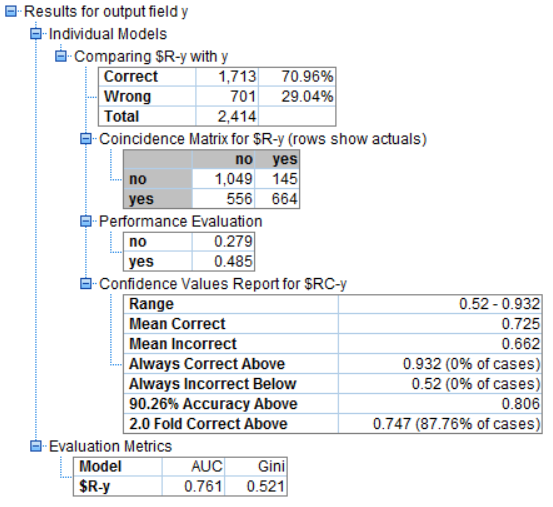


Figure : BASIC\_Testing\_Analysis

### **Testing Model 1: BOOSTING CHAID Model**

The analysis after running the BOOSTING CHAID model on the training set is as follows: **(Figure: 12)**

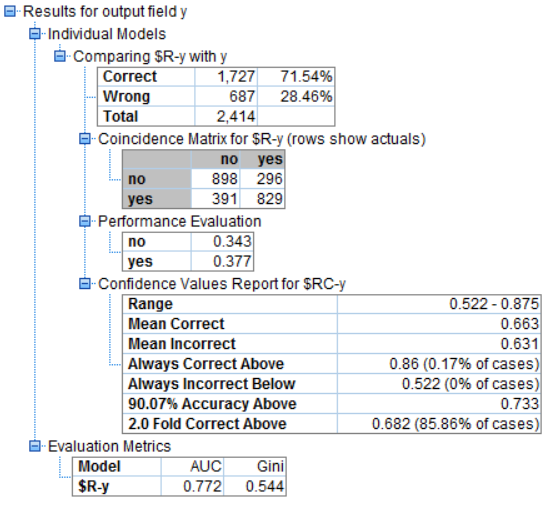


Figure : BOOSTING\_Testing\_Analysis

### **Testing Model 1: BAGGING CHAID Model**

The analysis after running the BAGGING CHAID model on the training set is as follows: **(Figure: 13)**

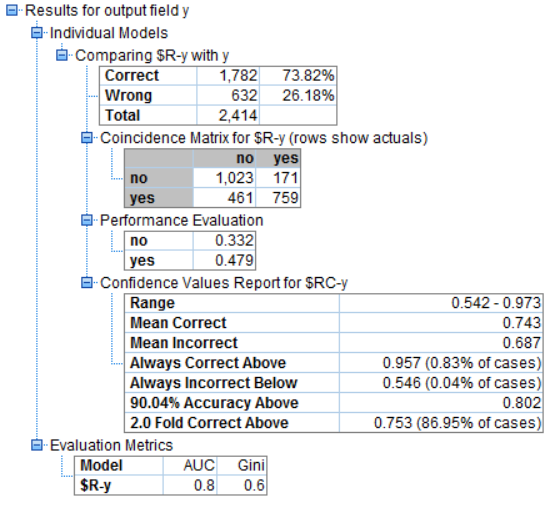


Figure : BAGGING\_Testing\_Analysis

As a final step, we apply the three trained models on the original dataset and observe the analysis and the model evaluation. **(Table 3)** (**IS7036-001-Group01\_ProjectReport\_CHAID.str – Part 4**)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Sensitivity** | **Specificity** | **Precision** | **AUC** | **Accuracy** | **F-Ratio** |
| **BASIC** | 54.42623 | 87.85595 | 82.07664 | 0.761 | 70.96106 | 65.45096 |
| **BOOSTING** | 67.95082 | 71.54101 | 73.68889 | 0.772 | 83.84424 | 70.70362 |
| **BAGGING** | 62.21311 | 85.67839 | 81.6129 | 0.8 | 73.81939 | 70.60465 |

Table : Model Comparison\_Testing

**Final Analysis based on the different parameters:**

* In terms of accuracy, Bagging and Boosting models performs the best for testing data, 73% and 83% respectively. These models identify the willing and unwilling customers the most.
* BOOSTING CHAID is the best at identifying the people willing to opt for the term deposit at 68% while bagging\_CHAID algorithm ranks second at 62%.
* For the identification of unwilling customers, Basic CHAID is the best model while, bagging algorithms ranks second trailing by only 2%.
* Precision of BASIC CHAID is also the best as it misclassifies only 18% of the unwilling customers as willing customers
* F-ratio for the BOOSTING CHAID model is the best i.e. it performs in synergy with Precision and Sensitivity.

**Conclusion:**

If we take AUC into consideration, the ENSEMBLE BAGGING model is the best of all the three models and hence will be used for further analysis, and to see how well this model fits the whole dataset, as we have considered stratified sampling for the initial training and testing analysis.

## **Artificial Neural Network as a Classifier**

## **Background**

An Artificial Neural Network is an information processing paradigm that is inspired by the way biological nervous systems process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of many interconnected processing elements, called Neurons, working in unison to solve specific problems. This structure gives ANN an advantage to exploit parallel processing, in turn giving a better and faster result. ANN learn by example and is configured for a specific application, such as pattern recognition, forecasting, prediction and classification, through the learning process.

## **Structure of a Neuron**

An artificial neuron can have many inputs and many outputs, these inputs and outputs have weights associated. Typically, Neurons has two modes of operation; the training mode and the deployment mode. In the training mode, the neuron can be trained for the input patterns. In the deployment mode, the neuron looks for the trained input patterns and its associated output is fired out. If the input doesn’t belong to the list of taught patterns, the firing rules are used to govern whether to fire the output or not.

Apart from the sets of input and output, a neural neuron is composed of two elements, a summation function and a transfer function. Summation functions performs a weighted sum of all the inputs and provide the result to the transfer function which then transforms the weighted sum to the required output.

The transfer functions are basically of tree types:

1. Linear: The final output is proportional to the total weighed output from the summation function
2. Sigmoid: The output varies continuously but not linearly as the input changes. Range of output is [0,1]
3. Tangent Hyperbolic: The output varies continuously but not linearly as the input changes. Range of output is [-1,1]

## **Architecture of Neural Networks**

A neural network is composed of certain inputs, outputs and a hidden layer of neurons. The interconnection between these structural elements gives a variety of Neural Networks. Some of the architecture of Neural Networks are as follows:

1. Feed-Forward Networks: Allows signals to travel one way only, from input to output. There is no feedback and hence the output of any layer doesn’t affect the input of that layer.
2. Feedback Networks: Feedback networks can have signals travelling in both directions by introducing loops in the network.

## **Learning Methods**

Information is stored in a neural network in the form of the weights associated with the inputs and output of neurons, input layer and output layer. Based on the weights, networks can be classified into two major categories:

1. Fixed Networks: Weights are fixed a priori based on the problem to be solved
2. Adaptive Networks: Weights are dynamic in nature and hence can change.

The interest of our analysis is the Adaptive Networks trained using supervised learning using a back-propagation algorithm. In back-propagation algorithms, the network is training by adjusting the weights of each unit in such a way that the error between the desired output and the actual output is the lowest.

## **Ensemble Methods**

Ensemble method are motivated by the fact that different classifiers (like Neural Networks, SVN etc.) may make different predictions on the test instances due to the specific characteristics of the classifier, or because of their sensitivity to the random observations in the training data. Ensemble method is an approach to increase the prediction accuracy by combining results from different classifiers. There are two types of ensembles namely Data-Centered ensembles and Model-centered ensembles. We will be using Data-centered ensembles in our analysis and apply them on Artificial Neural networks.

Ensembles are used to reduce the Bias introduced by the model, Variance in the prediction because of different training and testing sets and the Noise (can’t be reduced as it is induced in the actual data). We will be using **Bagging and Boosting** ensembles with our Artificial Neural Networks

Bagging is used to reduce the variance of the model by uniformly sampling data points from the original data with replacement and the sampled data is approximately of the size of original data with duplicated observations. The models are repeatedly build using these different samples and then an average of the all models in used to predict the actual output.

Boosting has weights associated with each instance of training instance and the classifier is trained with the use of these weights. These weights are modified iteratively based on the classifier performance i.e. the performance of the future models are dependent on the results from previous models. The basic idea is to focus on the incorrectly classified instances in future iterations by increasing the relative weight of these instances. The misclassification is basically introduced by the bias in the classifier and hence increasing the instance weight of misclassified instance will result in a new classifier that corrects the bias on these instances. Boosting helps in reducing both the bias and variance for the model.

## **Artificial Network Performance on Bank Marketing Data**

### **Model Training**

We first started off with the training of Network with all the variables on the training dataset. We used a lot different neural networks on different setting and these are the 5 best that are under consideration. The Neural Networks trained are as follows:

1. Basic MLP Neural Network with Automatic Hidden layers computation.
2. Basic MLP Neural Network with 2 Hidden Layers of 5 neurons each.
3. Ensemble MLP Neural Network with Automatic Hidden layers computation. Bagging Approach.
4. Ensemble MLP Neural Network with Automatic Hidden layers computation. Boosting Approach.
5. Basic RBF Neural Network with Automatic Hidden layers

The criteria for evaluation of these models will be based on the following parameters in order:

1. Sensitivity/Recall
2. Specificity
3. Precision
4. Area under the ROC curve(AUC)
5. Accuracy
6. F-ratio

The above criteria will be used in evaluating the models on testing data.

### **Basic MLP Neural Network with Automatic Hidden Layer Computation**

* **Basic Setting:**
  + - Objective: Standard Model
    - Model: Multilayer Perceptron
    - Hidden Layers: Auto calculated
    - Over-fit: Prevention set: 30%
    - Predictors Used: Age, Job, Marital Status, Education, Default, Balance, Housing, Loan, Contact, Day, Month, Campaign, Pdays, Previous, Poutcome
* **Fitted Neural Network: Refer the (Appendix\_Figure 9)**
* **Model Structure and Statistics:**

1. Network Structure

The model has 1 hidden layer with 8 neurons.

1. Predictor Importance

The predictors classified as most influential towards the output in order of decreasing importance are: Poutcome, Month, Pdays, Balance, Previous, Campaign, Age, Job, Day, Housing. **Refer (Appendix\_Figure 8) for software output.**

1. Accuracy

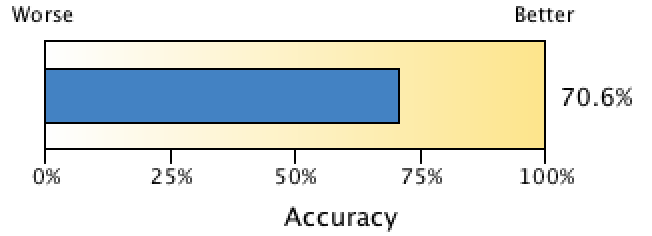


Figure 14: Accuracy\_Model 1\_ANN

The model is able to accurately classify 70.6% of the observations.

1. Confusion Matrix

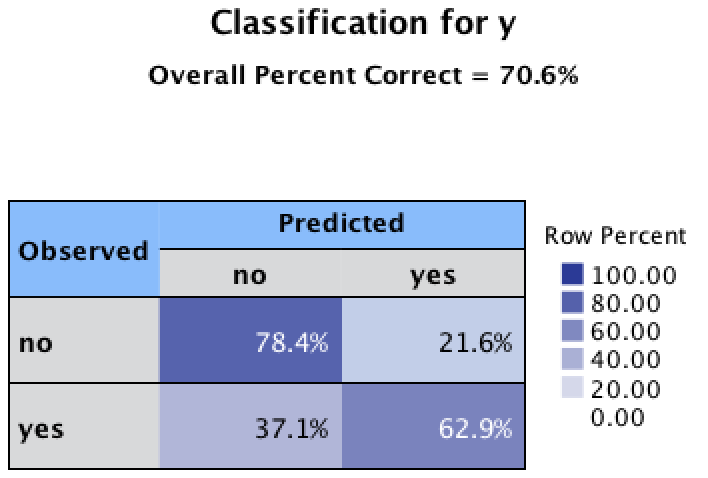


Figure 15: Classification\_Model\_1\_ANN

The model identified 78.4% of unwilling customers and 62.9% willing customers correctly. These statistics looks satisfactory as the model is able to identify the huge chunk of unwilling and willing customers correctly and hence will save us a lot of money and time in contacting the unwilling customers and will be able to help the business in generating a lot of money by contacting the customers willing to take term subscription. But still this **model will be losing 37% of the potential customers which in this case are many customers.**

### **Basic MLP Neural Network with two hidden layers of 5 Neurons each**

* **Basic Setting:**
  + - Objective: Standard Model
    - Model: Multilayer Perceptron
    - Hidden Layers: 2 Layers
    - Over-fit: Prevention set: 30%
    - Predictors Used: Age, Job, Marital Status, Education, Default, Balance, Housing, Loan, Contact, Day, Month, Campaign, Pdays, Previous, Poutcome
* **Refer (Appendix\_Figure 10) for software output.**
* **Fitted Neural Network: Refer the (Appendix\_Figure 11)**
* **Model Structure and Statistics:**

1. Network Structure

The model has 2 hidden layer with 5 neurons.

1. Predictor Importance

The predictors classified as most influential towards the output in order of decreasing importance are: Poutcome, Month, Previous, Campaign, Pdays, Age, Job, Balance, Housing, Marital

1. Accuracy

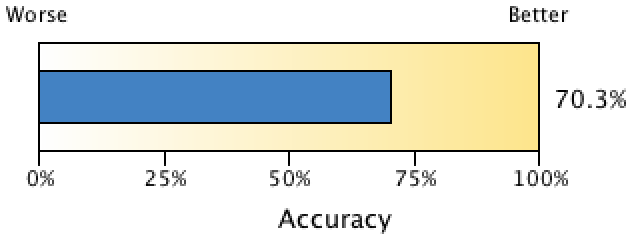


Figure 16: Accuracy\_Model\_2\_ANN

The model is able to accurately classify 70.3% of the observations.

1. Confusion Matrix

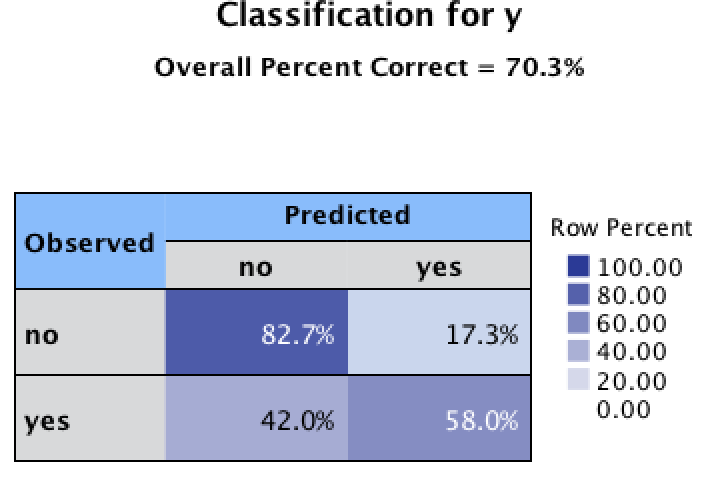


Figure 17: Classification\_Model\_2\_ANN

The model identified 82.7% of unwilling customers and 58% willing customers correctly. These statistics looks satisfactory as the model is able to identify the huge chunk of unwilling customers correctly and hence will save us a lot of money and time in contacting the unwilling customers but is **incorrectly identifying 42% of the potential customers and hence will not be able to help the business in generating a lot of money by contacting the customers willing to take term subscription.**

### **Ensemble MLP Neural Network with Automatic Layer Computation – Bagging Approach**

* **Basic Setting:**
  + - Objective: Standard Model
    - Model: Multilayer Perceptron
    - Hidden Layers: Auto calculated
    - Over-fit: Prevention set: 30%
    - Predictors Used: Age, Job, Marital Status, Education, Default, Balance, Housing, Loan, Contact, Day, Month, Campaign, Pdays, Previous, Poutcome
* **Refer (Appendix\_Figure 12) for software output.**
* **Fitted Neural Network: Refer the (Appendix\_Figure 15)**
* **Model Structure and Statistics:**

1. Predictor Importance

The predictors classified as most influential towards the output in order of decreasing importance are: Month, Poutcome, Housing, Job, Campaign, Previous, Education, Loan, Pdays, Marital

1. Accuracy

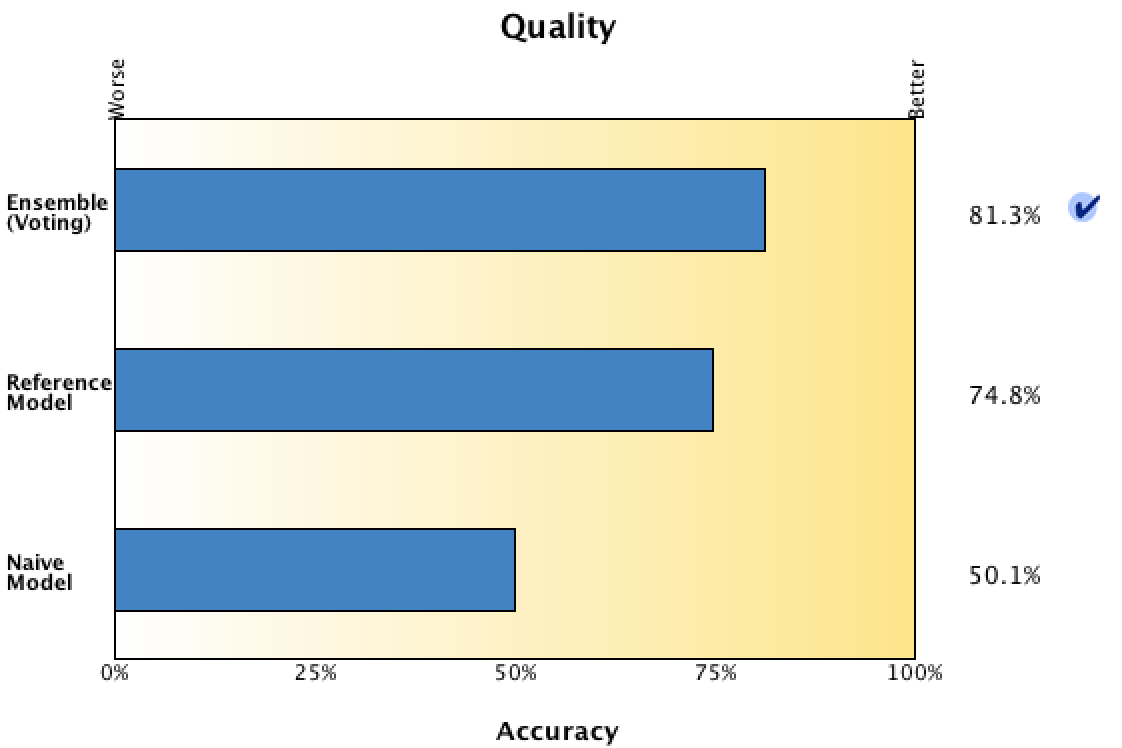


Figure 18: Accuracy\_Model\_3\_ANN

The Ensemble model is able to accurately classify 81% of the observations. The Ensemble model is being selected via voting which means that it selects the final category that has the highest probability, most often across the base model. The Ensemble model has better accuracy than the Reference model Naïve Model. Here reference model is the one which has been build using all the observations without any repetition. The Naïve model represents the accuracy of the case when no model is built, and assigns all the records to the modal category.

1. Predictor Frequency

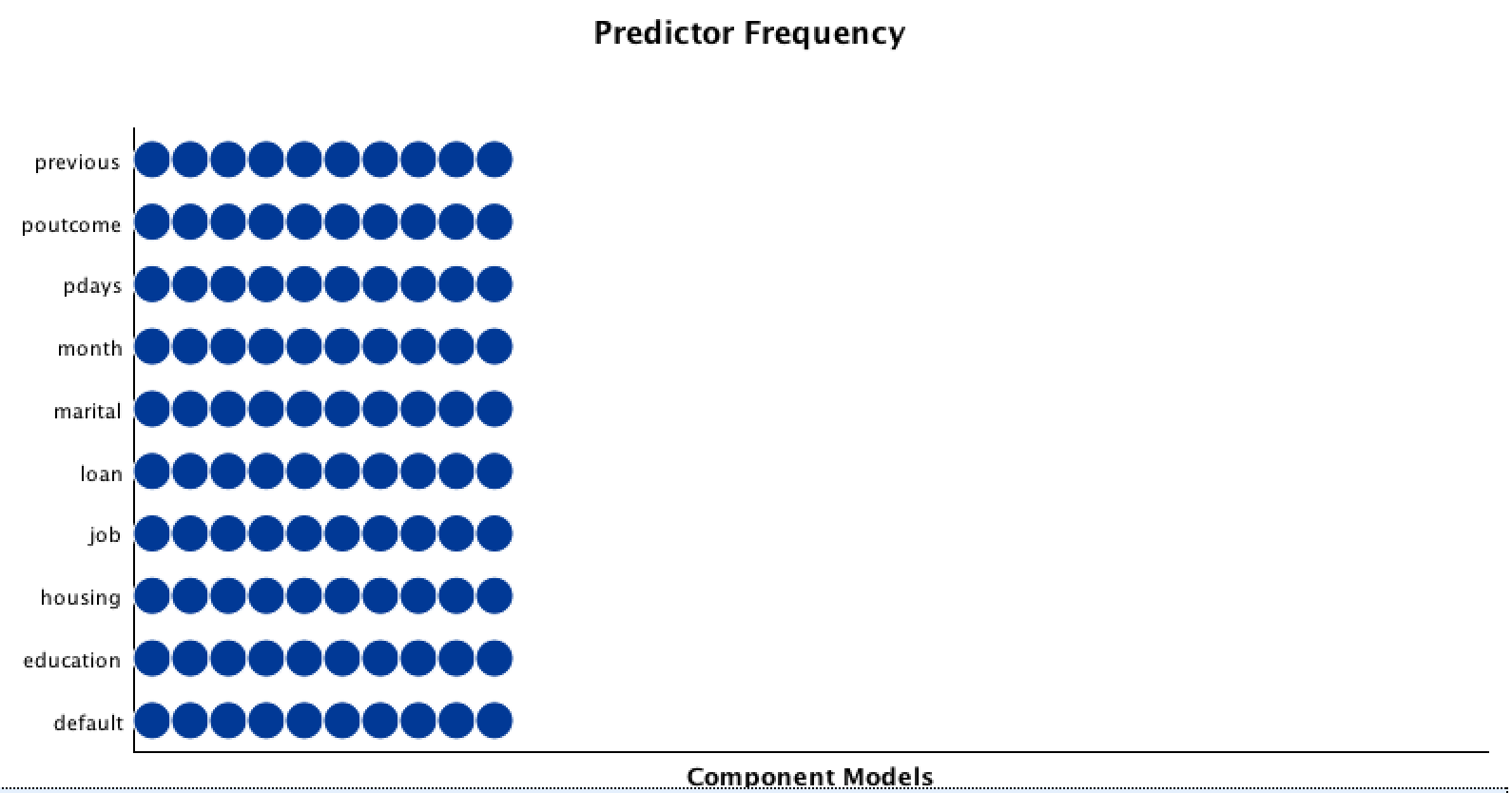


Figure 19: Predictor\_Frequency\_Model\_3\_ANN

This plot shows the distribution of predictors across all the 10 component models of the ensemble. From the plot we can see that Previous has been the most frequent predictor used in the component models and Month which is the most important predictor has also been used in a lot of component models but it ranks 4th in the frequency which means that though it has been used not quite a lot in building ensembles, the model is still sensitive to its input and hence it can be said that it is an important classifier. Same can be said about Poutcome. All the predictors have been used in all the 10 models.

1. Component Model Details

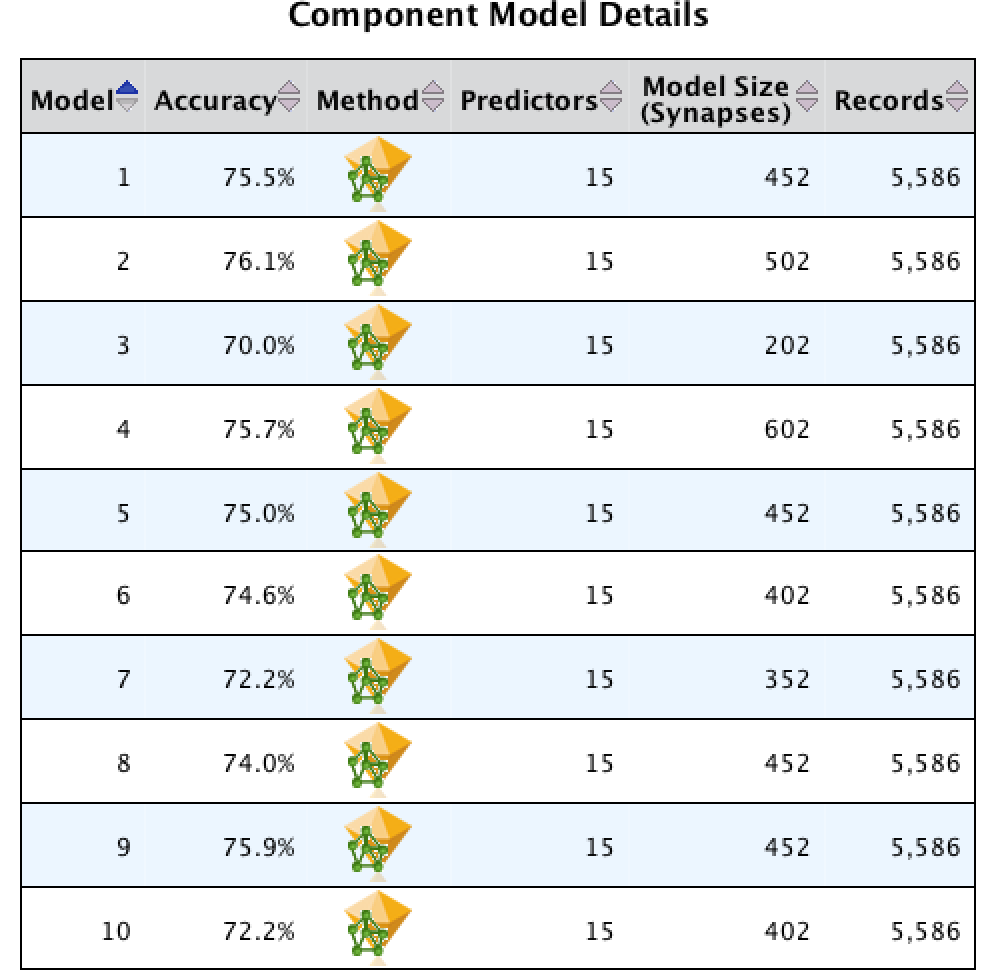


Figure 20: Component\_Details\_Model\_3\_ANN

The above chart gives the details of the component models developed for different sections of the data. We can see that all the models are using all the 15 predictors and their accuracy ranges from 70% to 76% which is around the accuracy of reference model but when combined all these models gives a total accuracy of around 81% i.e. their cumulative accuracy is around 81%.

1. Confusion Matrix

|  |  |  |
| --- | --- | --- |
| **Confusion Matrix** | | |
|  | **Predicted** | |
| **Observed** | yes | no |
| yes | 74.75% | 25.25% |
| no | 12.22% | 87.78% |

Table 4: Confusion\_Matrix\_Model\_3\_ANN

The model identified 87.78% of unwilling customers and 74% willing customers correctly. These statistics looks satisfactory as the model is able to identify the huge chunk of unwilling and willing customers correctly and hence will save us a lot of money and time in contacting the unwilling customers and will be able to help the business in generating a lot of money by contacting the customers willing to take term subscription. **This model is a significant improvement over the models described above.**

### **Ensemble MLP Neural Network with Automatic Layer Computation – Boosting Approach**

* **Basic Setting:**
  + - Objective: Standard Model
    - Model: Multilayer Perceptron
    - Hidden Layers: Auto calculated
    - Over-fit: Prevention set: 30%
    - Predictors Used: Age, Job, Marital Status, Education, Default, Balance, Housing, Loan, Contact, Day, Month, Campaign, Pdays, Previous, Poutcome
* **Refer (Appendix\_Figure 13) for software output.**
* **Fitted Neural Network: Refer the (Appendix\_Figure 15)**
* **Model Structure and Statistics:**

1. Predictor Importance

The predictors classified as most influential towards the output in order of decreasing importance are: Month, Poutcome, Housing, Job, Previous, Campaign, Loan, Pdays, Education, Marital.

1. Accuracy

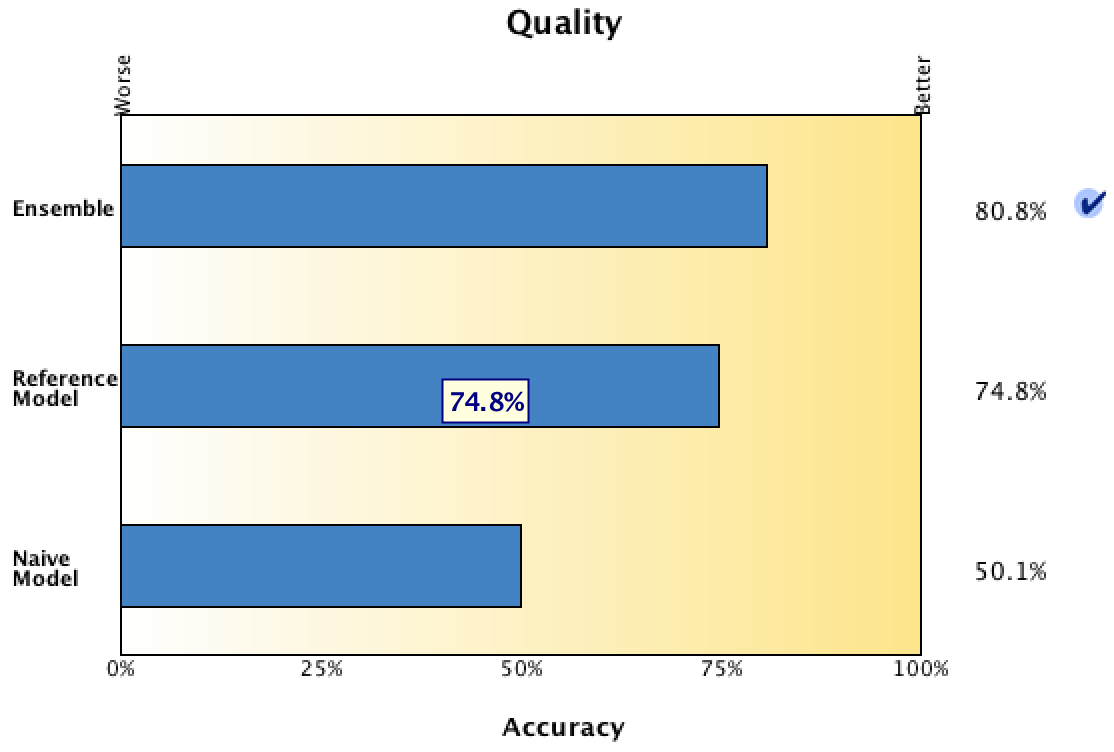


Figure 21: Accuracy\_Model\_4\_ANN

The Ensemble model can accurately classify almost 81% of the observations. The Ensemble model is being selected via voting which means that it selects the final category that has the highest probability, most often across the base model. The Ensemble model has better accuracy than the Reference model Naïve Model. Here reference model is the first component model. The Naïve model represents the accuracy of the case when no model is built, and assigns all the records to the modal category.

1. Predictor Frequency

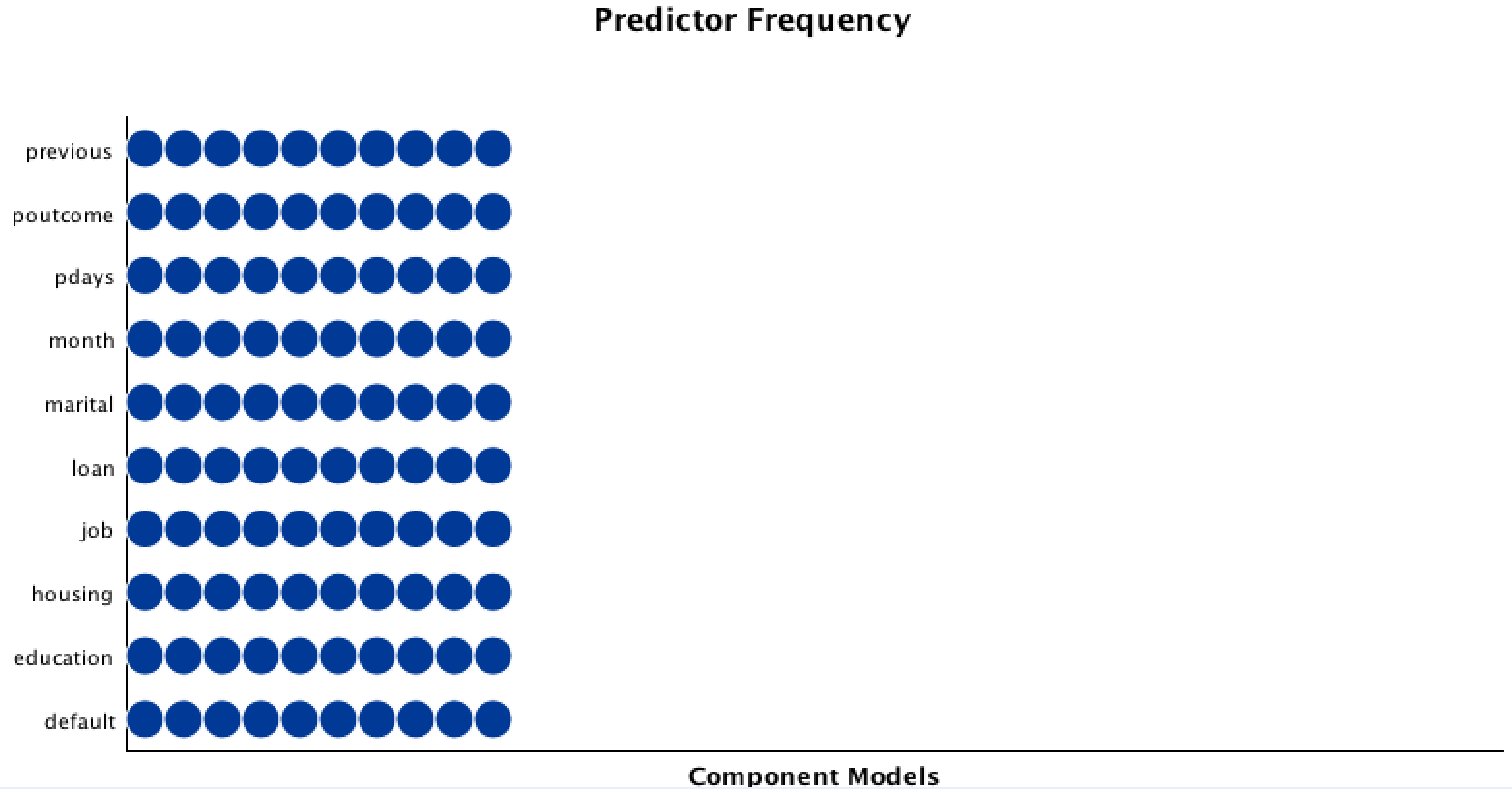


Figure 22: Predictor\_Frequency\_Model\_4\_ANN

This plot shows the distribution of predictors across all the 10 component models of the ensemble. From the plot we can see that Previous has been the most frequent predictor used in the component models and Month which is the most important predictor has also been used in a lot of component models but it ranks 4th in the frequency which means that though it has been used not quite a lot in building ensembles, the model is still sensitive to its input and hence it can be said that it is an important classifier. Same can be said about Poutcome. But it can also be seen that all the predictors have been used in all the 10 models.

1. Component Model Details

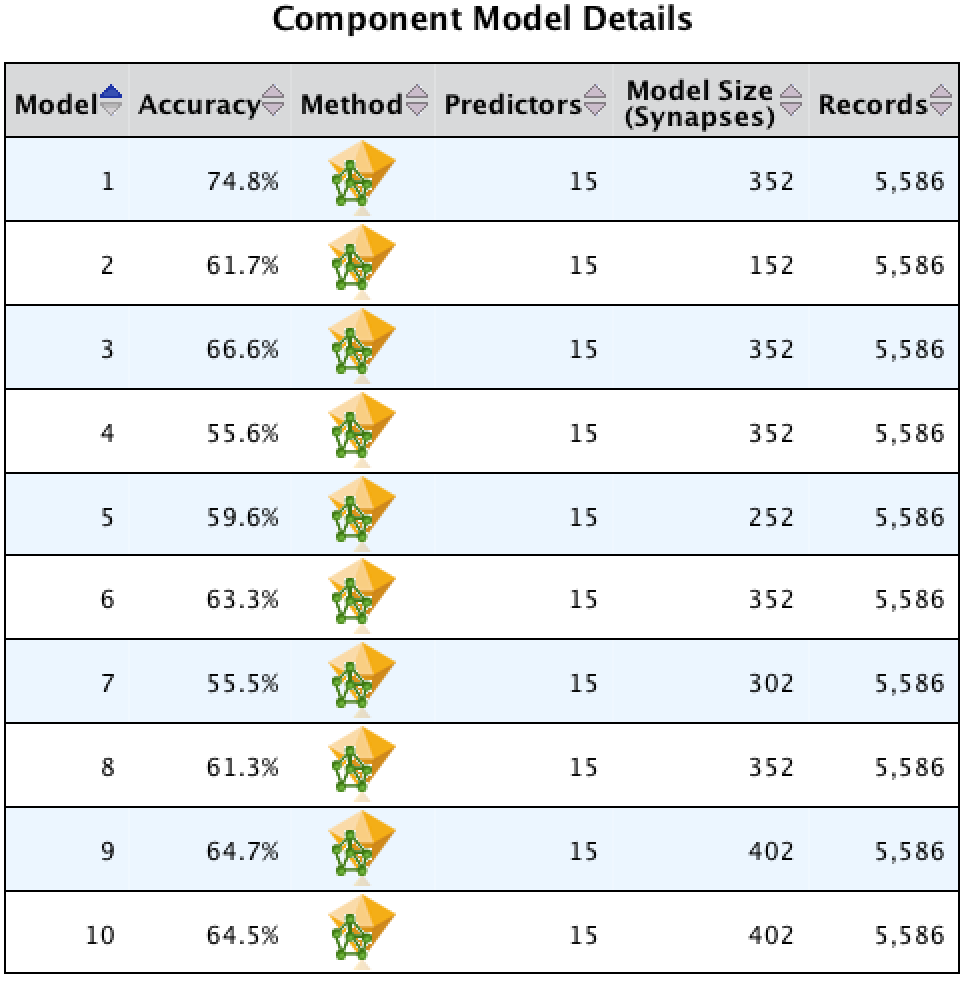


Figure 23: Component\_Details\_Model\_4\_ANN

The above chart gives the details of the component models developed for different sections of the data. We can see that all the models are using all the 15 predictors and their accuracy ranges from 55% to 71% which is around the accuracy of reference model but when combined all these models gives a total accuracy of around 81% which is almost the same as bagging model. The accuracy shown against different component models shows their classification accuracy in different spaces of the problem space.

1. Ensemble Cumulative Accuracy

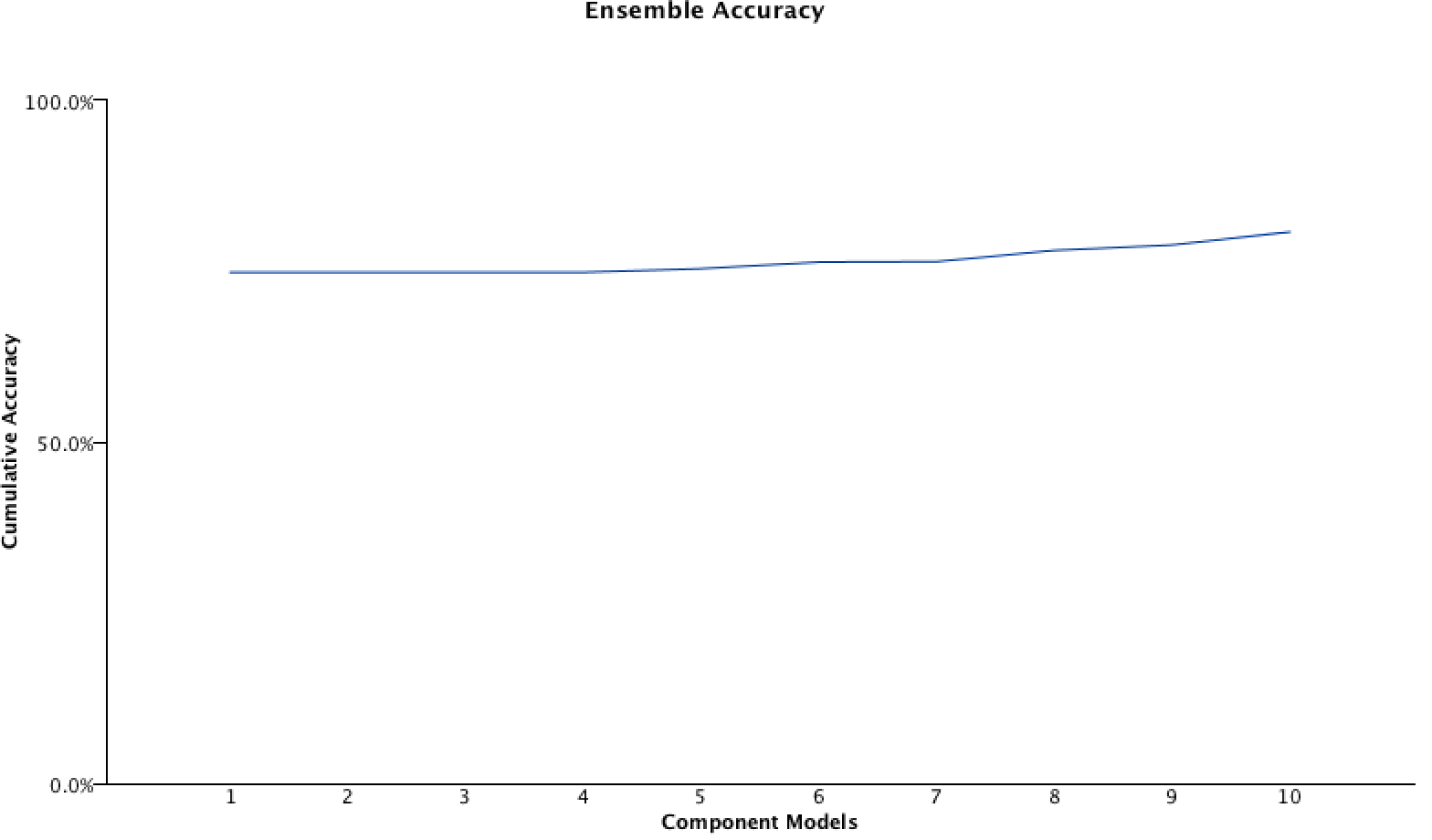


Figure 24: Ensemble\_Accuracy\_Model\_4\_ANN

1. Confusion Matrix

|  |  |  |
| --- | --- | --- |
| **Confusion Matrix** | | |
|  | **Predicted** | |
| **Observed** | yes | no |
| yes | 77.86% | 22.14% |
| no | 16.34% | 83.66% |

Table 5: Confusion\_Matrix\_Model\_4\_ANN

The model identified 83.66% of unwilling customers and 77.86% willing customers correctly. These statistics looks satisfactory as the model is able to identify the huge chunk of unwilling and willing customers correctly and hence will save us a lot of money and time in contacting the unwilling customers and will be able to help the business in generating a lot of money by contacting the customers willing to take term subscription. **This model is a significant improvement over the models described above. Though the bagging model classifies unwilling customers better by 3% but the boosting model increases the correct classification of willing customers by 4% which in turn will eventually bring more business.**

### **Basic RBF Neural Network with Automatic Hidden layers**

* **Basic Setting:**
* Objective: Standard Model
* Model: Radial Bias Function
* Hidden Layers: Auto calculated
* Over-fit: Prevention set: 30%
* Predictors Used: Age, Job, Marital Status, Education, Default, Balance, Housing, Loan, Contact, Day, Month, Campaign, Pdays, Previous, Poutcome
* **Refer (Appendix\_Figure 14) for software output.**
* **Fitted Neural Network: Refer the (Appendix\_Figure 15)**
* **Model Structure and Statistics:**

1. Network Structure

The model has 1 hidden layer with 10 neurons.

1. Predictor Importance

The predictors classified as most influential towards the output in order of decreasing importance are: Poutcome, Month, Pdays, Balance, Previous, Campaign, Age, Job, Day, Housing

1. Accuracy

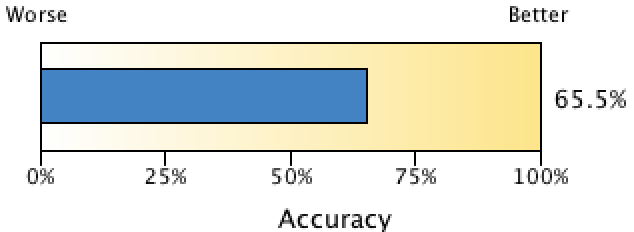


Figure : Accuracy\_Model\_5\_ANN

The model is able to accurately classify 65.5% of the observations.

1. Confusion Matrix

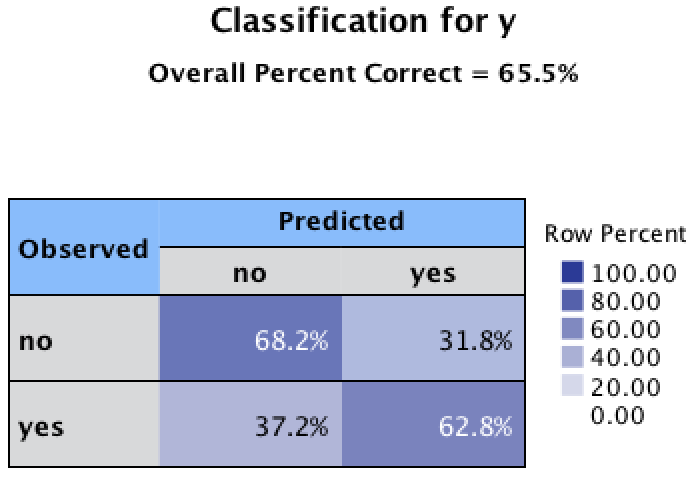


Figure : Classification\_Model\_5\_ANN

The model identified 68.2% of unwilling customers and 62.8% willing customers correctly. These statistics looks satisfactory as the model is able to identify the huge chunk of unwilling and willing customers correctly and hence will save us a lot of money and time in contacting the unwilling customers and will be able to help the business in generating a lot of money by contacting the customers willing to take term subscription. But still **this model will be losing 37% of the potential customers which is same as the first model but it is also misclassifying 31.8% of the unwilling customers as willing customers which in this scenario will be a lot of customers and thus will lead us to spend our resources on customers who potentially will not yield any business**.

## **Model Testing**

We test all the five models on the testing dataset now to look at their individual performance and to compare them with each other based on the evaluation criterions defined above.

### **Confusion Matrix of models**

The analysis of testing measure for all the models are as follows:







Table : Confusion\_Matrix\_Comparison

* **Sensitivity, Specificity, Precision, Accuracy and f-ratio**

On running the testing data on the model generated in the training phase, we get the following results:



Table : Model Evaluation\_Criteria\_ANN

**Analysis:**

1. In terms of accuracy, Bagging and Boosting models performs the best for testing data, 76% and 73% respectively. These models identify the willing and unwilling customers the most
2. Boosting algorithm ranks first in the Identification of willing customers and after that bagging algorithm ranks second lagging by only 2%.
3. For the identification of unwilling customers, Basic MLP with 2 layers perform the best and bagging algorithms ranks second trailing by only 3%
4. Precision of bagging algorithm is also the best as it misclassifies only 21% of the unwilling customers as willing customers
5. F-ratio for the bagging model is the best i.e. it performs in synergy with Precision and Sensitivity.

* **AUC statistics for models**

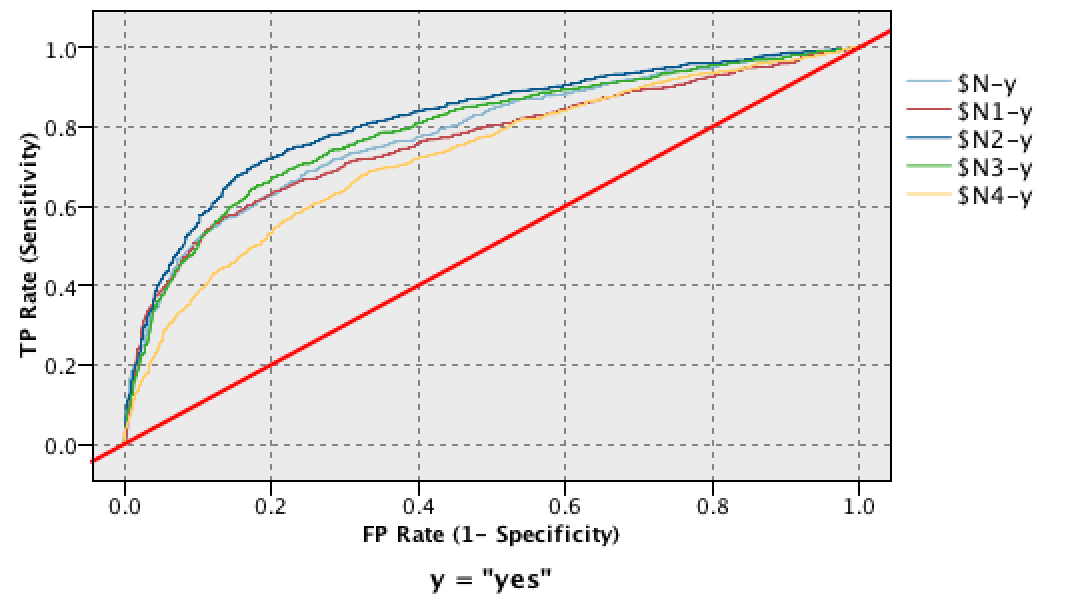
****

Figure : ROC Curves Comparison

$N: Basic MLP Neural Network, $N1: Basic MLP Neural Network with 2 layers

$N2: Bagging MLP Ensemble Models, $N3: Boosting MLP Ensemble Model

$N4: RBF Neural Network



Table : Model Evaluation Criteria\_AUC

Taking AUC into consideration, Ensemble Bagging Models performs the best and hence will be used for analysis.

## **Support Vector Machine as a Classifier**

## **Background**

Support Vector Machine is a classification technique used to predict either continuous or categorical output. It maps the data into a dimensional plane, and then separates the nonlinear data points into different groups by nonlinear transformation, thus creating a hyperplane in the new space which divides the different output classes.

It uses nonlinear kernel function to separate the data into linearly separable spaces and then the hyperplanes are created to separate different classes, and the one with maximum margin is selected to optimize the model.

Four different types of Kernel Functions are available in the SVM node –

1. Linear - it is used when nonlinear relationships in the data is minimum
2. Polynomial – It allows higher order terms in the new dimension space
3. Radial Basis Function - It is used to fit highly nonlinear data
4. Sigmoid – It can also fit highly nonlinear data

## **Create the model on the training dataset. Then test this trained model on the testing set.**

1. We built the following four models –

### **SVM with Linear Kernel Function –**

**Basic Settings:**

**Expert –**

Stopping criteria: 1.01E-3

Regularization parameter(C): 10

Regression Precision(epsilon): 0.1

Kernel Type: Linear

We used the training data set to train this model, and then passed it through the analysis node. This data contained a high nonlinearity relationships, so the accuracy using nonlinear kernel function is not that accurate. The accuracy of this model is 67.17 %

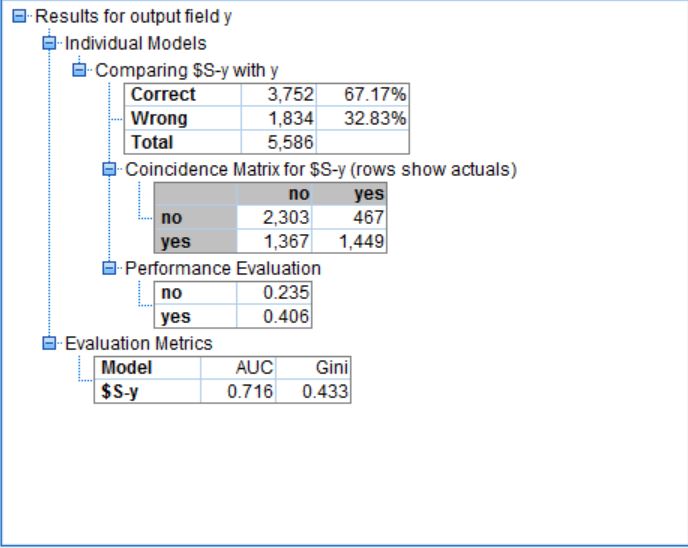


Figure : Training analysis on SVM - Linear

### **SVM with Radial Basis function –**

**Basic Settings:**

**Expert –**

Stopping criteria: 1.01E-3

Regularization parameter(C): 5

Regression Precision(epsilon): 0.1

Kernel Type: RBF

RBF Gamma: 0.25 (It should be between 0.2 and 0.4 based on the number of input variables)

We used autoclassifier node as well to get the best model by different combination of above parameters. As the number of predictors were high, RBF performed really well on the training set and the prediction accuracy was 81.31%.

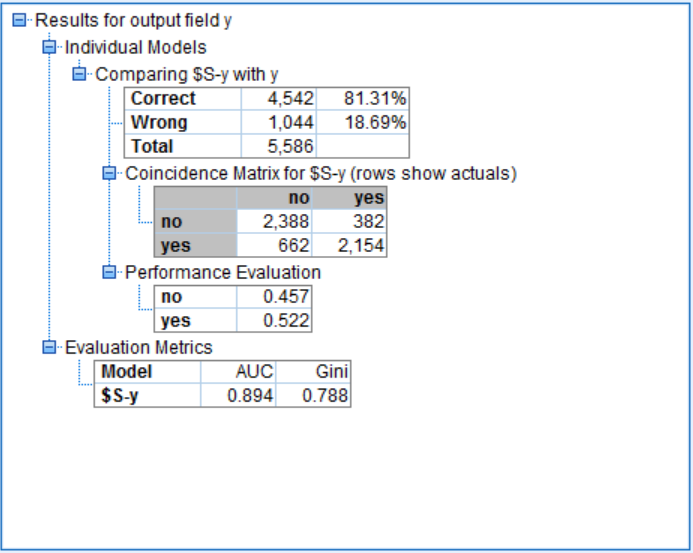


Figure :Training analysis on SVM - RBF

### **SVM with Polynomial kernel function –**

**Basic Settings:**

**Expert –**

Stopping criteria: 1.01E-3

Regularization parameter(C): 10

Regression Precision(epsilon): 0.1

Kernel Type: Polynomial

Gamma: 0.2

Bias: 0

Degree: 3

The SVM with polynomial kernel function provided us the best result. Its prediction accuracy turned out to be 92.12%.

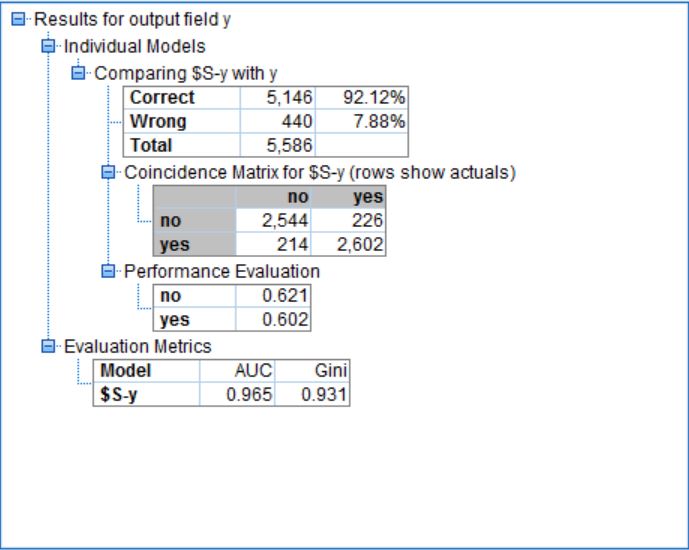


Figure : Training analysis on SVM - Polynomial

### **SVM with Sigmoid kernel function –**

**Basic Settings:**

**Expert –**

Stopping criteria: 1.01E-3

Regularization parameter(C): 10

Regression Precision(epsilon): 0.1

Kernel Type: Sigmoid

Gamma: 0.25

Bias: 0

Sigmoid kernel function turned out to be most inappropriate for our data. It predicted values in training data with 55.62 % accuracy.

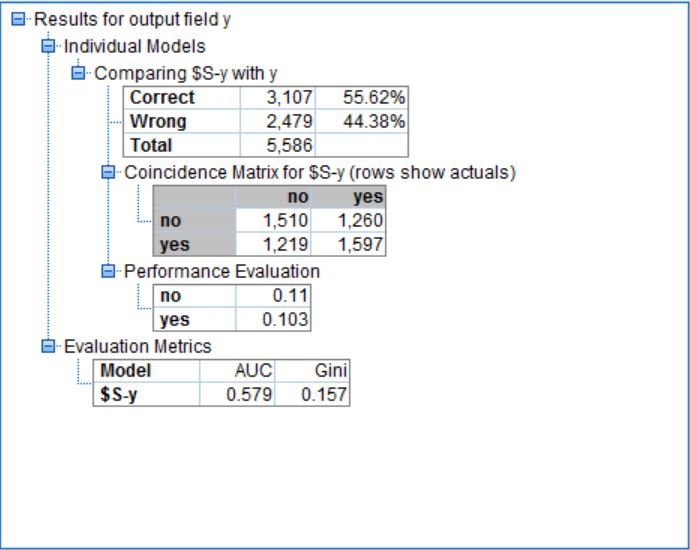


Figure : Training analysis on SVM - Sigmoid

## **The model parameters that we will be using for model evaluation are as follows:**

* Sensitivity/Recall
* Specificity
* Precision
* Area under the ROC curve(AUC)
* Accuracy
* F-ratio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Sensitivity** | **Specificity** | **Precision** | **Accuracy** | **AUC** | **F-Ratio** |
| **Linear** | 0.5146 | 0.8314 | 0.7563 | 0.6718 | 0.7160 | 0.6124 |
| **RBF** | 0.7649 | 0.8621 | 0.8494 | 81.3100 | 0.8940 | 0.8049 |
| **Polynomial** | 0.9240 | 0.9184 | 0.9201 | 92.1200 | 0.9650 | 0.9220 |
| **Sigmoid** | 0.5671 | 0.5451 | 0.5590 | 55.6200 | 0.5790 | 0.5630 |

Table :Model Comparison (training data)

## **Running the trained model on the testing data and the entire data to check how well the model fits the data.**

We will now run all the four models created in the above step, to run them on the test data set to see how well the trained model fits the data. The parameters that we will check for this purpose are the accuracy, sensitivity, specificity and AUC(Area Under the Curve). A comprehensive comparison among all the three models is also shown as part of this study.

A table of comparison for all the four models can help us analyze how well the trained models fit the test data . All these models are then passed through a comparative test to check which model is the best fit, and then we use that to analyze how well this ‘best chosen’ model fits the overall data**.** The comparative analysis for all the four models in the end give us a comparison for the ROC Curves for all the three models and subsequently their respective AUC’s.

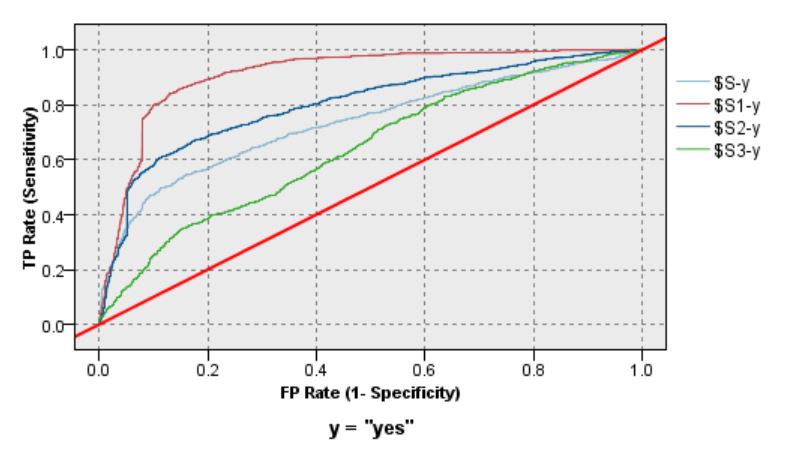


Figure : ROC curve of different models on testing data

### **Testing Model 1: SVM with Linear kernel function**

The analysis after running the SVM with linear kernel function on the testing set is as follows:

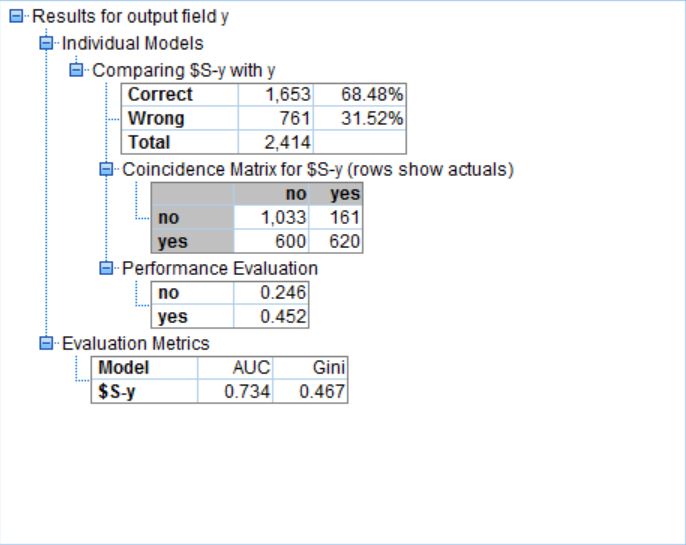


Figure : Testing data analysis on SVM - Linear

### **Testing Model 2: SVM with RBF kernel function**

The analysis after running the SVM with RBF kernel function on the testing set is as follow:

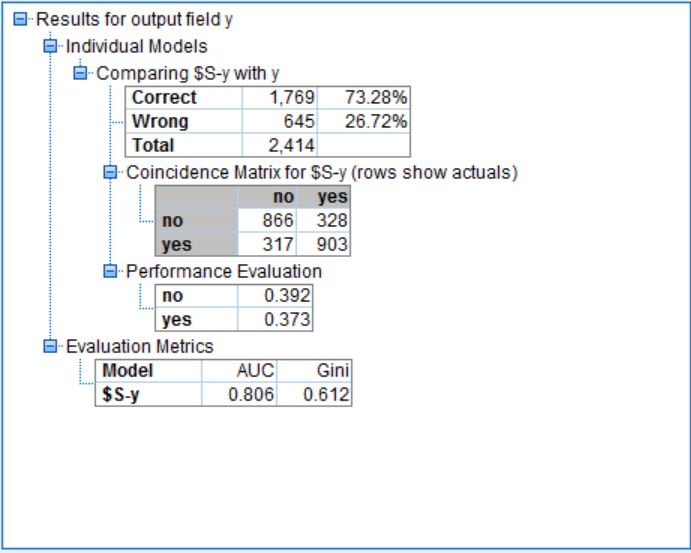


Figure : Testing data analysis on SVM - RBF

### **Testing Model 3: SVM with Polynomial kernel function**

The analysis after running the SVM with polynomial kernel function on the testing set is as follows:

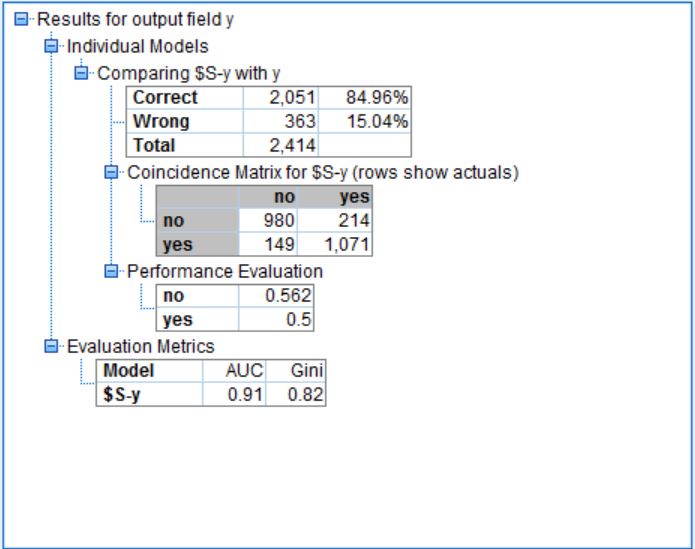


Figure : Testing data analysis on SVM - Polynomial

### **Testing Model : SVM with Sigmoid kernel function**

The analysis after running the SVM with sigmoid kernel function on the testing set is as follows:

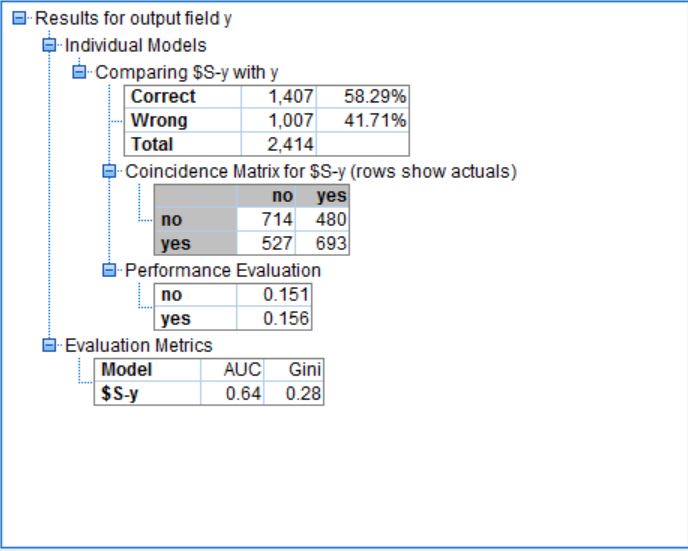


Figure : Testing data analysis on SVM - Sigmoid

## **Performance Metrics of trained models on testing data –**

Following are the performance metrics for the trained model on the training data set. As per the metrics Polynomial kernel function outperforms all other functions in our application. It provides best sensitivity, specificity, precision, accuracy and area under curve.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Sensitivity** | **Specificity** | **Precision** | **Accuracy** | **AUC** | **F-Ratio** |
| **Linear** | 0.5082 | 0.8652 | 0.7939 | 68.48 | 0.7340 | 0.6197 |
| **RBF** | 0.7402 | 0.7253 | 0.7335 | 73.28 | 0.8060 | 0.7368 |
| **Polynomial** | 0.8779 | 0.8208 | 0.8779 | 84.96 | 0.9100 | 0.8779 |
| **Sigmoid** | 0.5680 | 0.5980 | 0.5908 | 58.29 | 0.6400 | 0.5792 |

Table : Model Comparison (testing data)

## **Logistic regression**

## **Background**

Logistic regression is a type of regression in which the response is a binary variable. This response variable can be related to discrete and/or continuous explanatory variables. It varies from linear regression in that sense that, in the linear regression a combination of values taken by the predictors help modeling the value of the response variable. While, in the case of logistic regression, it is the probability or the odds of the response variable is modelled based on a combination of the values taken by the predictor variables. This regression is applicable for the following cases:

* To model the probabilities of response variable as a function of few predictor variables.
* To perform descriptive discriminate analyses.
* To predict the probability of a n individual falling in one of the two categories of a binary response.
* To classify individuals into two categories.

The data set handled in this case is of direct marketing campaigns of a Portuguese banking institution. The data sets informs about the calls made to clients and if the calls were successful in making the clients subscribe to a product. This falls in the category of point iv from the cases discussed above. As, we want to reduce the telemarketing costs and help the institution understand calling which customer would give a higher probability of making him/her subscribe to the product.

## **Model building**

SPSS helps us build logistic models using various methods. The following are the options provided by SPSS.

* Stepwise logistic regression
* Forwards logistic regression
* Backwards logistic regression
* Backwards Stepwise logistic regression

These methods build the model in steps to include or/and exclude variables which can predict the outcome well. In each step of the model building, a variable is considered for addition into or for removal from the set of exploratory variables. The addition and removal is done based on different techniques like F-tests or t-tests.

Stepwise Logistic Regression:

SPSS follows the following process to run stepwise regression:

* First it starts with no variables in the model then by entering the variable with smallest p-value followed, in the next step with the variable with the next smallest p-value and so on.
* Variables already in the equation are removed if their p-value is larger than the default limit as a result of the inclusion of another variable.
* This process terminates when there are no more variables eligible for inclusion or exclusion.

### **Forwards logistic regression:**

SPSS follows the following process to run Forwards logistic regression:

* First it starts with no variables, the by entering the variable with smallest p-value.
* This is followed, in the next step with the variable with the next smallest p-value and so on.
* This process terminates when there are no more variables eligible for inclusion.

### **Backwards logistic regression:**

SPSS follows the following process to run Backwards logistic regression:

* First it starts with all variables included in the model
* Then, the by removing the variable whose removal gives the most statistically insignificant deterioration of the model fit. This is followed by the next variable similarly.
* This process terminates when there are no more variables eligible for exclusion.

Backwards Stepwise logistic regression:

SPSS follows the following process to Backwards Stepwise logistic regression:

* First it starts with all variables in the model then by removing the variable with highest p-value followed, in the next step with the variable with the next highest p-value and so on.
* Variables already not in the equation are added if their p-value is smaller than the default limit because of the inclusion of another variable.
* This process terminates when there are no more variables eligible for inclusion or exclusion.

## **Analysis metrics:**

The results of the regression model can be analyzed based on the Area Under curve, True positive rate, false positive rate and accuracy.

Confusion matrix is a very important metric for understanding a binary classification model. For a binary prediction there can be 4 types of outcomes:

* Prediction value :0, Actual value: 0. This is called true negative i.e., the negative values are correctly predicted.
* Prediction value :0, Actual value: 1. This is called false negative i.e., the values are incorrectly predicted as negative.
* Prediction value :1, Actual value: 0. This is called false positive i.e., the values are incorrectly predicted as positive.
* Prediction value :1, Actual value: 1. This is called true positive i.e., the values are correctly predicted as positive.

The aim of the model building is to optimize on the cost involved in calling clients, by predicting who will have a better probability of buying the product. So, it is important for the model to correctly predict the people who will convert. This means we need a higher true positive rate and a lower false positive rate.

The ROC curve acts as a combined metric of both False positive rate and True positive rate. The higher the area under the curve for ROC, the better the model.

The basic analysis metrics used are as follows:

* Accuracy
* Sensitivity/Recall
* Specificity
* Precision
* f-ratio
* AUC
* -2 log Likelihood
* No of predictor Variables

## **Training Model Analysis:**

Logistic regression was performed on the stratified dataset with 70% of the data separated for training and 30% for testing. After the dataset was separated, a logistic regression was applied on the training dataset. SPSS allows to apply logistic regression in 4 ways, Stepwise, Forwards, Backwards, Backwards Stepwise. All these were applied on the data set and four models were built. The Confusion matrices, TPR, FPR, AUC were recorded by the analysis mode and reported as below.

### **Stepwise Logistic Model**

A model was built using stepwise logistic regression. The analysis node presents results as shown in the following image.

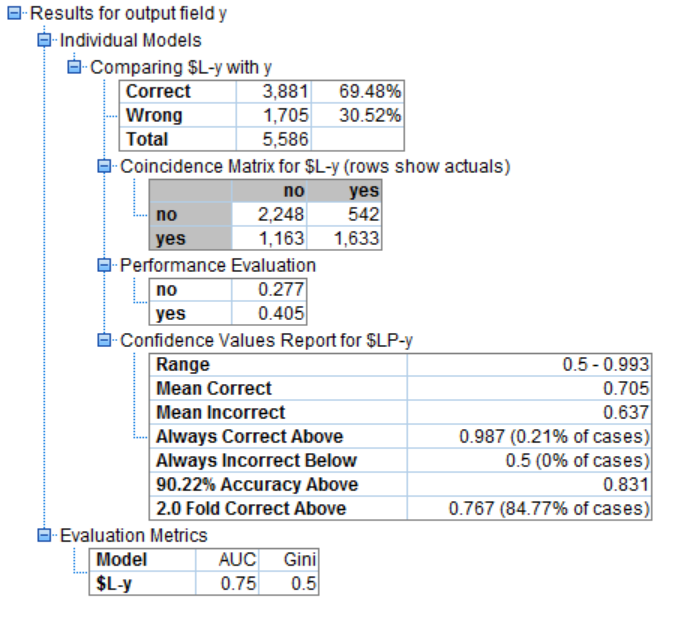


Figure : Stepwise\_Logical\_Model\_Analysis

The Step Summary of the model is as shown below

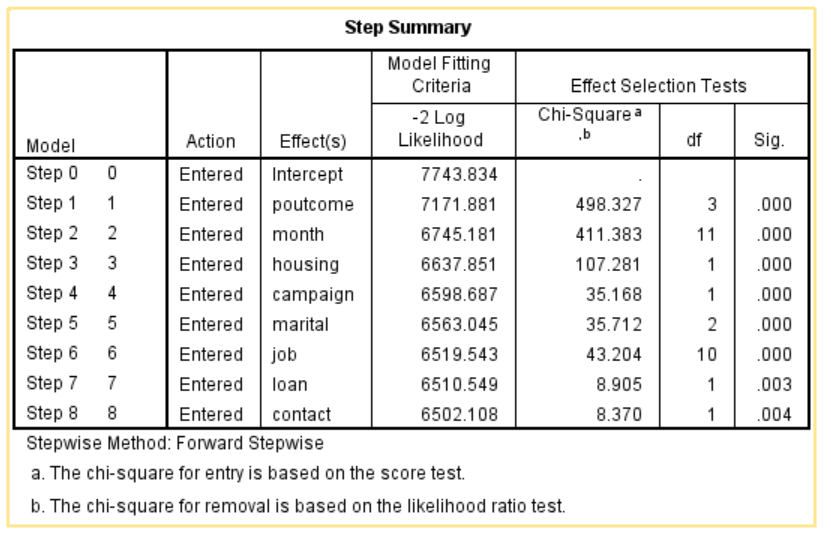


Table : Step Summary

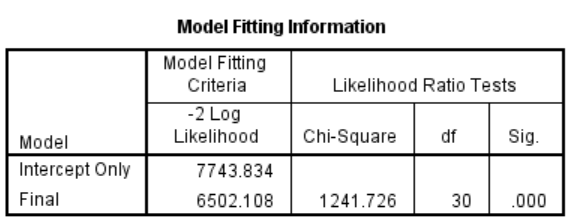


Table : Model Fitting Information

Results:

* The Accuracy of the model produced is 69.48%.
* Sensitivity is 0.5840
* Specificity is 0.8057
* Precision is 0.7504
* F-ratio is 0.6568
* AUC is 0.75
* -2log Likelihood is 6502.108
* The final model has 8 variables

### **Forwards Logistic Model**

A model was built using forwards logistic regression. The analysis node presents results as shown in the following image.

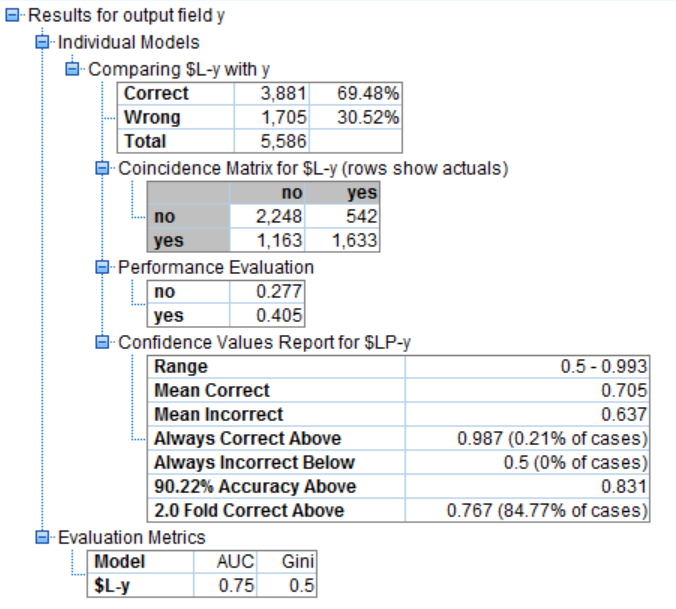


Figure : Forwards Logistic Model\_Analysis

The Step Summary of the model is as shown below

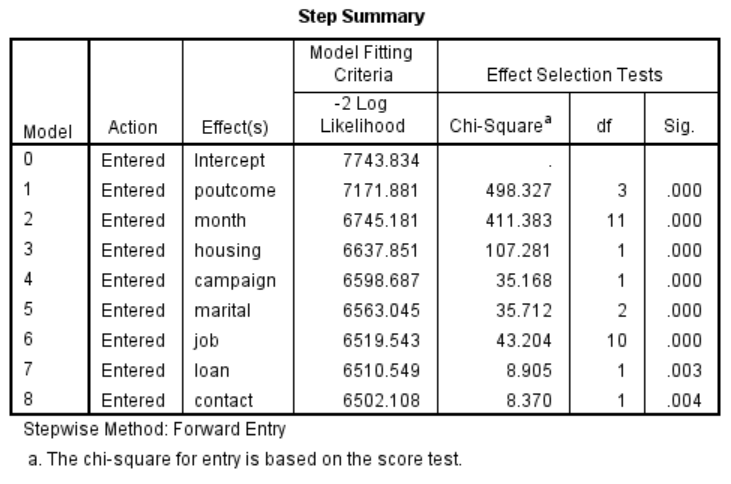


Table : Step\_Summary

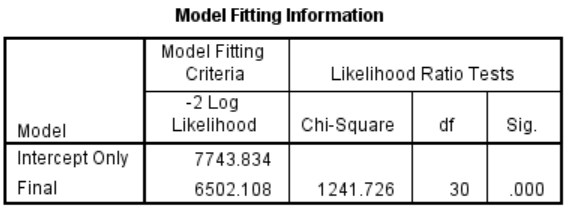


Table : Model Fitting Information

Results:

* The Accuracy of the model produced is 69.48%.
* Sensitivity is 0.5840
* Specificity is 0.8057
* Precision is 0.7504
* F-ratio is 0.6568
* AUC is 0.75
* -2log Likelihood is 6502.108
* The final model has 8 variables

### **Backwards Logistic Model**

A model was built using backwards logistic regression. The analysis node presents results as shown in the following image.

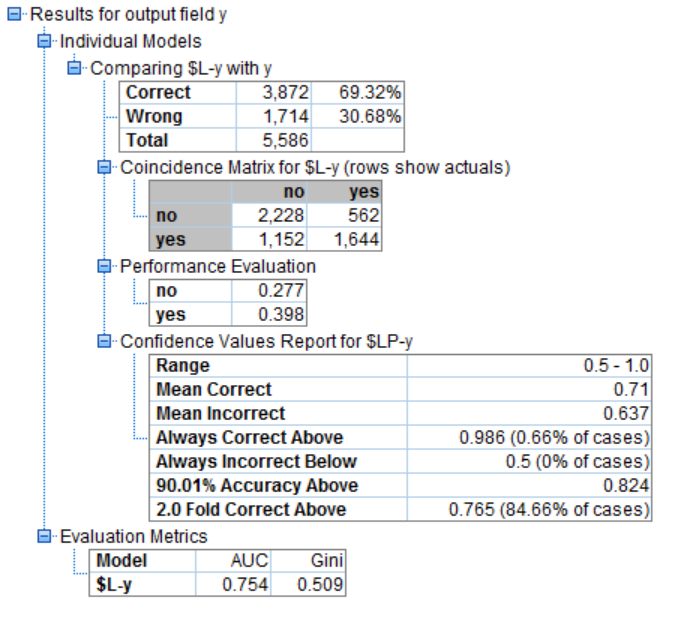


Figure : Backwards Logistic\_Analysis

The Step Summary of the model is as shown below

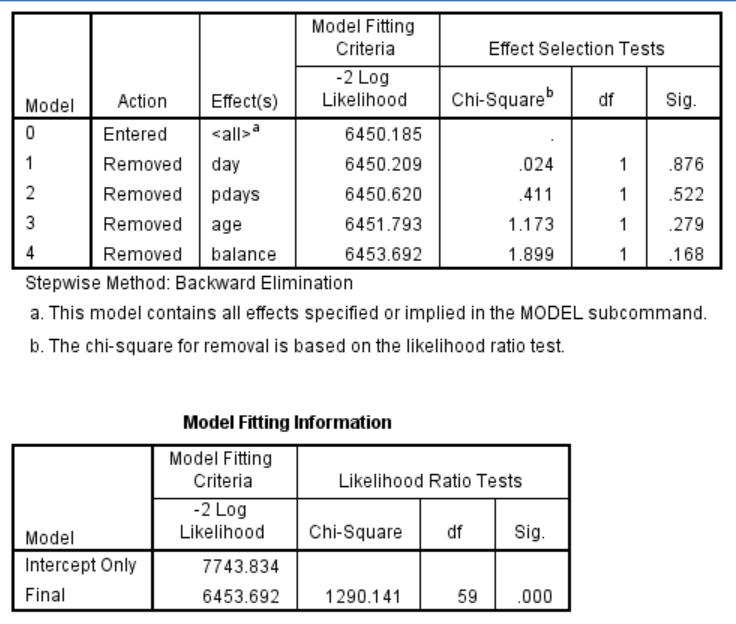


Table : Model Fitting Information

Results:

* The Accuracy of the model produced is 69.32%
* Sensitivity is 0.5879
* Specificity is 0.7985
* Precision is 0.7448
* F-ratio is 0.6571
* AUC is 0.754
* -2log Likelihood is 6453.692
* The final model has 11 variables

### **Backwards Stepwise Logistic Model**

A model was built using backwards stepwise logistic regression. The analysis node presents results as shown in the following image.

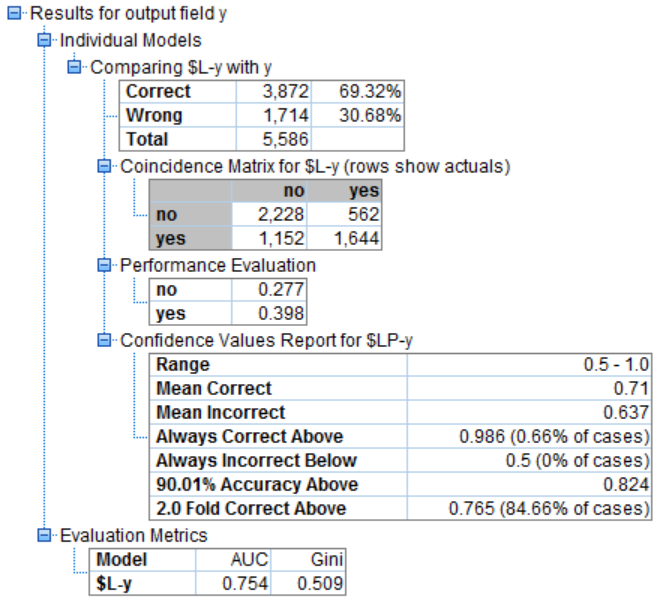


Figure : Backwards\_Stepwise\_Logistic\_Analysis

The Step Summary of the model is as shown below



Table : Backwards\_Stepwise\_Logistic\_Summary

Results:

* The Accuracy of the model produced is 69.32%
* Sensitivity is 0.5879
* Specificity is 0.7985
* Precision is 0.7448
* F-ratio is 0.6571
* AUC is 0.754
* 2log Likelihood is 6453.692
* The final model has 11 variables

### **Training Results Comparison:**

The results from each model generated are compared below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Sensitivity/Recall | Specificity | Precision | f-ratio | AUC | -2 log Likelihood | Variable count |
| Stepwise | 69.48% | 0.5840 | 0.8057 | 0.7504 | 0.6568 | 0.75 | 6502.108 | 8 |
| Forwards | 69.48% | 0.5840 | 0.8057 | 0.7504 | 0.6568 | 0.75 | 6502.108 | 8 |
| Backwards | 69.32% | 0.5879 | 0.7985 | 0.7448 | 0.6571 | 0.754 | 6453.692 | 11 |
| Backwards Stepwise | 69.32% | 0.5879 | 0.7985 | 0.7448 | 0.6571 | 0.754 | 6453.692 | 11 |

Table : Training\_Results\_Comparison

Observations:

* Accuracy, specificity, precision values of Stepwise, forward are better than the backwards and backwards stepwise models
* Sensitivity, f-ratio, AUC of Stepwise, forward are lesser than the backwards and backwards stepwise models
* Stepwise, forwards models have 8 predictor variables while Backwards and backwards stepwise have 11 predictor variables.
* This hints the model might be overfit in the second case.

## **Testing Model Analysis:**

All the models were tested on the testing data set individually. The Confusion matrices, AUC were recorded by the analysis node and reported as below.

## **Stepwise Logistic Model**

A model was built using Stepwise logistic regression. The analysis node presents results as shown in the following image.

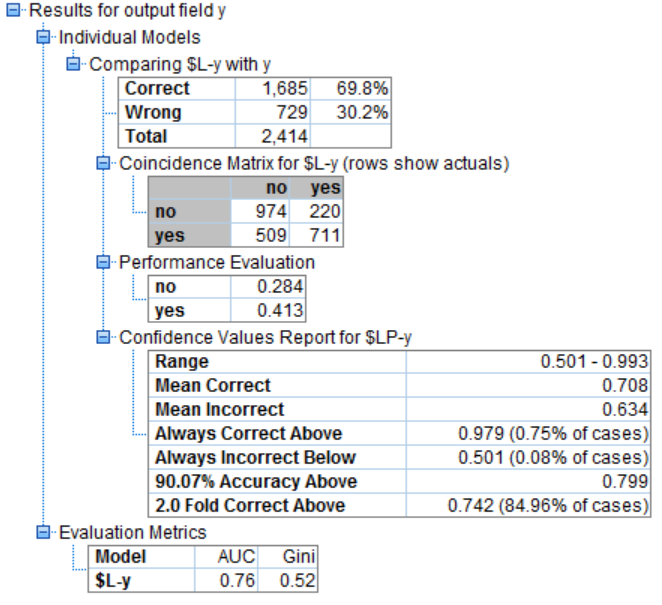


Figure : Stepwise\_Logical\_Analysis

Results:

* The Accuracy of the model produced is 69.92%
* Sensitivity is 0.5827
* Specificity is 0.8157
* Precision is 0.7597
* F-ratio is 0.6596
* AUC is 0.75

## **Forwards Logistic Model**

A model was built using forwards logistic regression. The analysis node presents results as shown in the following image.

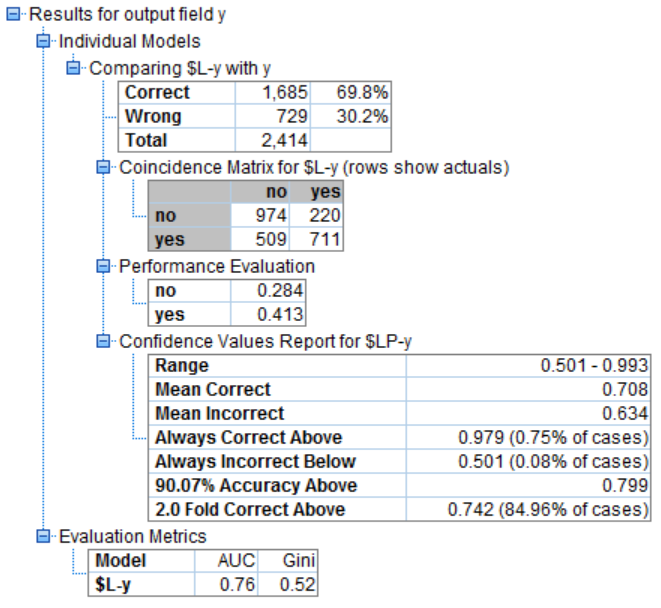


Figure : Forwards\_Logistic\_Analysis

Results:

* The Accuracy of the model produced is 69.92%
* Sensitivity is 0.5827
* Specificity is 0.8157
* Precision is 0.7597
* F-ratio is 0.6596
* AUC is 0.75

## **Backwards Logistic Model**

A model was built using backwards logistic regression. The analysis node presents results as shown in the following image.

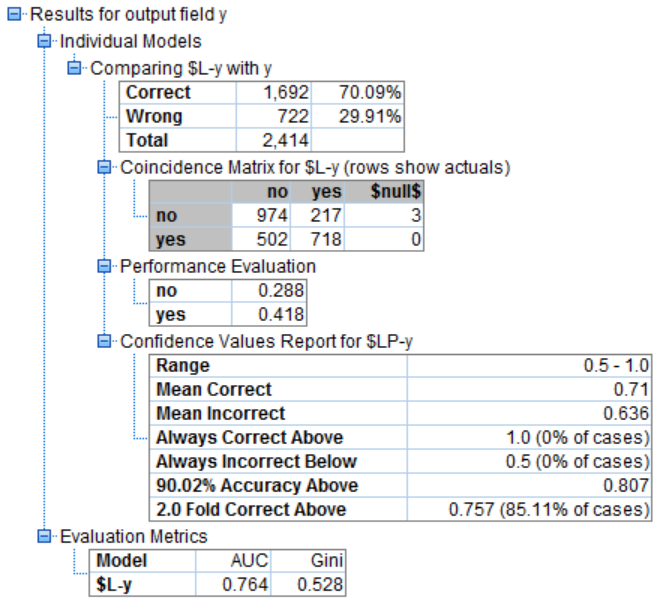


Figure : Backwards\_Logistic\_Analysis

Results:

* The Accuracy of the model produced is 70.31%
* Sensitivity is 0.5885
* Specificity is 0.8178
* Precision is 0.7635
* F-ratio is 0.6647
* AUC is 0.764

## **Backwards Stepwise Logistic Model**

A model was built using backwards stepwise logistic regression. The analysis node presents results as shown in the following image.

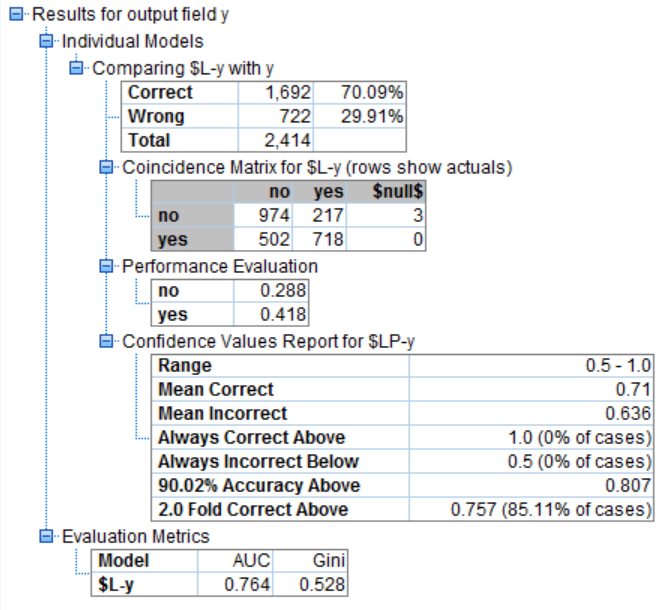


Figure : Backwards\_Stepwise\_Logistic\_Analysis

Results:

* The Accuracy of the model produced is 70.31%
* Sensitivity is 0.5885
* Specificity is 0.8178
* Precision is 0.7635
* F-ratio is 0.6647
* AUC is 0.764

## **Testing Results Comparison:**

The results from each model generated are compared below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Sensitivity/Recall | Specificity | Precision | f-ratio | AUC |
| Stepwise | 69.92% | 0.5827 | 0.8157 | 0.7597 | 0.6596 | 0.75 |
| Forwards | 69.92% | 0.5827 | 0.8157 | 0.7597 | 0.6596 | 0.75 |
| Backwards | 70.31% | 0.5885 | 0.8178 | 0.7635 | 0.6647 | 0.764 |
| Backwards Stepwise | 70.31% | 0.5885 | 0.8178 | 0.7635 | 0.6647 | 0.764 |

Table : Testing\_Results\_Comparison

Observations:

* All metrics except for f-ratio of Stepwise, forward models are lesser than the backwards and backwards stepwise models

## **ROC Comparison**

After this the ROC’s of all the models built were compared on the testing data. The plot of ROC was as below.

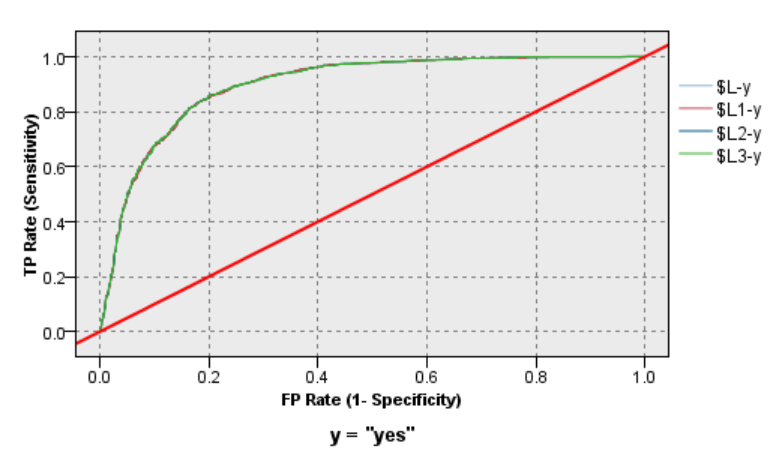


Figure : ROC Curve Comparison

It can be observed that all the models have overlapping ROCs and are very close in predictions.

## **Comparison of Testing and Training results:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Accuracy | Sensitivity/Recall | Specificity | Precision | f-ratio | AUC | -2 log Likelihood | Variable count |
| Stepwise  (Training) | 69.48% | 0.5840 | 0.8057 | 0.7504 | 0.6568 | 0.75 | 6502.108 | 8 |
| Stepwise  (Testing) | 69.92% | 0.5827 | 0.8157 | 0.7597 | 0.6596 | 0.75 |  |  |
| Forwards  (Training) | 69.48% | 0.5840 | 0.8057 | 0.7504 | 0.6568 | 0.75 | 6502.108 | 8 |
| Forwards  (Testing) | 69.92% | 0.5827 | 0.8157 | 0.7597 | 0.6596 | 0.75 |  |  |
| Backwards  (Training) | 69.32% | 0.5879 | 0.7985 | 0.7448 | 0.6571 | 0.754 | 6453.692 | 11 |
| Backwards  (Testing) | 70.31% | 0.5885 | 0.8178 | 0.7635 | 0.6647 | 0.764 |  |  |
| Backwards Stepwise  (Training) | 69.32% | 0.5879 | 0.7985 | 0.7448 | 0.6571 | 0.754 | 6453.692 | 11 |
| Backwards Stepwise  (Testing) | 70.31% | 0.5885 | 0.8178 | 0.7635 | 0.6647 | 0.764 |  |  |

Table : Comparison of Training and Testing Results

Observations:

* The Backwards models had higher metrics than the forwards models. So, It was thought that it was probably because of over fitting. But, the testing results are pretty similar to results on training dataset. So, it can’t be because of overfitting.
* It can be observed that Backwards selection regression models can be the best models
* Any of the two (Backwards selection regression and Backwards Stepwise regression model) can be selected as the best model as these are similar with same metrics and same predictor variables in the final model.

# **EVALUATION-(CRISP – DM Methodology) Recommendations**

Based on the results from the model testing analysis, we finalized 4 models to be the best among their category. Now we will compare them based on the preset criteria for model evaluation.

The finalized models are: CHAID – Bagging Ensemble, ANN – Bagging Ensemble, SVN – Polynomial Kernel and Logistic Regression with backward stepwise selection.

The model statistics to be used for their evaluation are as follows:

* **Recommendation Based on Accuracy, Sensitivity, Specificity, Precision, f-ratio**



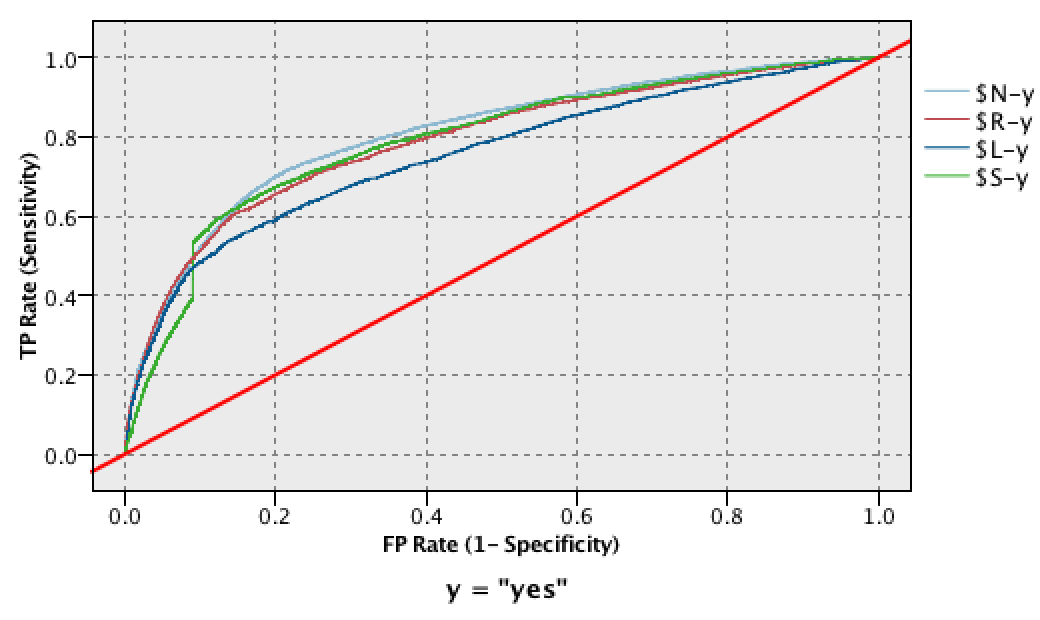
Table : Recommendation\_Analysis

**Analysis:**

1. CHAID classifies 81% of the customers correctly into unwilling and willing customers and has the highest accuracy among all the other models
2. SVM classifies 72% of the customers correctly into willing customers while ANN classifies 71% of the customers as willing. Hence both performs well in identifying the willing customers
3. CHAID classifies 84% of the customers correctly into unwilling customers while Logistic Regression and ANN classifies 80% of the customers as willing. Hence both perform well in identifying the willing customers
4. CHAID has the highest precision among all the algorithms followed by ANN
5. F-ratio should be the governing factor as it takes into consideration the number of customers correctly identified as willing, incorrectly identified as willing and incorrectly identified customers as unwilling. All these types of customers directly affect the business either in the sense of generating revenue or in case of proper utilization of resources. **Since both CHAID and ANN give the same f-ratio, we will then look at the second-best performance metric for this business model, sensitivity. By looking at the sensitivity values, ANN performs way between than CHAID in identifying the willing customers (revenue generating customers and hence it will be our final recommendation for model.**

* **Recommendation Based on AUC and ROC**





$N – AN(Bagging), $R - CHAID(Bagging), $L – Logistic Regression, $S – SVN

Figure : ROC Curve Comparison for Recommendation

**Analysis:**

The results of the AUC conform with the results obtained from the previous analysis. Since ANN has the maximum Area under the ROC curve, we will recommend this as our final model.

# **Conclusion:**

Our recommended model for predicting the prospective willing customers will be Ensemble ANN with Bagging. Since, month is the most influential predictor for that model, on further analysis we found out that in the second and third quarter i.e. from Apr to Sep, the count of willing customers increases and hence the bank must focus more during these months.

# **REFERENCES:**

* http://stats.stackexchange.com/
* https://www.ibm.com/support/knowledgecenter/en
* http://stats.idre.ucla.edu/spss/
* https://en.wikipedia.org/wiki/Main\_Page
* http://www.unige.ch/ses/sococ/cl/spss/cmd/regression.methods.html
* https://www.spss-tutorials.com/stepwise-regression-in-spss-example/
* https://onlinecourses.science.psu.edu/stat504/node/149
* http://www.morganclaypool.com/doi/abs/10.2200/S00240ED1V01Y200912DMK002
* Data Mining Concepts and Techniques by Jaiwe, Mitcheline Kamber, Jian Pei
* An Introduction to Neural Networks by Kevin Gurnui.
* Data Mining- The Text Book by Charu C. Agarwal