Mining a Bank Telemarketing Dataset

APRIL 22, 2016

Basak, Debasmita (basakda)
Bhimaraju, Sai Deepak(bhimarsk)
Bose, Raunak(boserk)
Devadula, Sandeep Gupta(devadusa)

Preface

This project has been conducted by a group of four members for the Advanced Business Intelligence course with dataset obtained by downloading bank-additional-full.csv (contained in bank-additional.zip) from https://archive.ics.uci.edu/ml/datasets/Bank+Marketing. The purpose of the project was to utilise several data mining techniques through the use of SPSS Modeller to understand and build models in predicting the best campaign contact with the clients for subscribing deposit. The performances are measured by statistical measures such as *Accuracy, Precision and Recall*. The model performance has been examined based on its F4 score.

Table of Contents

1. Introduction

In today's world, most bank marketing campaigns are dependent on customers' huge electronic data. The volume of these data source makes it impossible for a human analyst to come up with interesting information that will help in decision-making process to call or not to call a given individual. Moreover, given the fact that most people have become used to saying no to direct phone marketing, it is extremely essential to understand who exactly are the people where one may achieve success. Data mining models are very useful in improving performance of these campaigns. This project introduces applications of important models of data mining; Support Vector Machine(SVM), Ross Quinlan decision tree model(C5.0), CART, Logistic Regression and Artificial Neural Networks .

This data was originally collected and studied done by Moro et al. The objective is to examine the performance of these models on a real-world data of bank deposit subscription for a Portuguese banking institution and identify the main characteristics that affect a success (Term deposit subscribed by the client) in order to increase effectiveness of campaigns. We have implemented CRISP-DM methodology for our analysis. The experimental results demonstrate, with higher accuracies, the success of these models in predicting the best campaign contact with the clients for subscribing deposit. The performances are measured by statistical measures such as *Accuracy, Precision and Recall*. The model performance is been examined based on its F4 score.

2. Background

With the ever increasing rate at which new products are created, it is essential that customers are aware of the nuances among the various products which would enable them to make an informed choice. However, the very fact that there are way too many products in the market has led most people to become numb to marketing campaigns which are meant to be their source of information. Market segmentation is one useful tool in understanding one's customers broadly with the help of simple parameters. However, even this approach has been failing as the number of products and the variety among customers has grown. In other words, it is not humanly possible for one to simply look at the data and arrive at conclusions. The use of Data Mining approaches is one way to solve this conundrum and we make use of a number of them to arrive at a model which would help us pinpoint at a far lesser number of customers to contact while increasing the overall success rate.

The original study was conducted by Moro et al [1] where they modeled the data collected from a Portuguese bank to better predict success in direct telemarketing for selling long term deposits. This dataset which is now publicly available from the UCI online Machine Learning repository is used by us to create models similar to the ones created by Moro et al. We also perform an analysis which would improve our understanding of the various Data Mining approaches as well as using the IBM SPSS Modeler software package.

3. Problem Analysis and Design

As mentioned in the earlier sections, we make use of a bank marketing dataset to predict whether or not a particular person would be interested in the product. We adopted the CRISP-DM process as shown in Figure xx to achieve the same and as a result, we we begin with the first step which is Business Understanding.

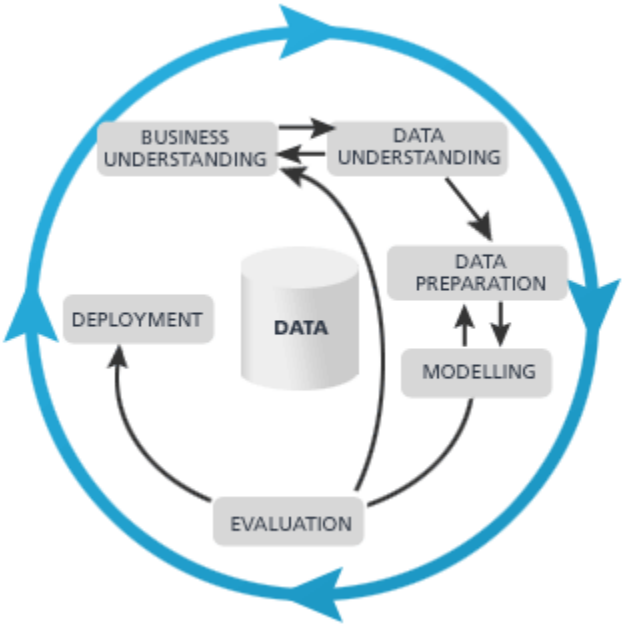


Figure xx

The business in question is the selling of long term deposits where a human agent telephones a number of potential customers and explains them the benefits of their new product. On the completion of the call, the agent has a “yes” or a “no” answer, thus indicating whether or not was the call a success. In the absence of any modeling, one has to call every single potential customer in order to maximize sales. This leads to hiring a large number of agents which can be costly and time consuming, as a lot of effort goes in training them as well. The models we intend to generate aim to potentially decrease the total number of calls made while trying to

maintain the total number of positive responses. To compare the models against one another, we use both Precision and Recall which achieve the purposes mentioned above. One good measure which makes use of both of the factors is the $F_\beta$score, where $\beta$is the weightage one gives to Recall against Precision.

$$F_\beta = (1 + \beta^2) * Recall * Precision/(\beta^2 * Precision + Recall)$$

The closer the value of $F_\beta$is, the closer one is in achieving the promise of finding an interested customer for every β calls made. The bone of contention in this context is to understand what would the value of βbe. Since no clear instructions have been given in the dataset or its associated documentation, we would only be able to fix a number once we have a good understanding of the data.

4. Data Understanding

The next step in the CRISP-DM process is data understanding and preparation. This data set was obtained by downloading bank-additional-full.csv (contained in bank-additional.zip) from https://archive.ics.uci.edu/ml/datasets/Bank+Marketing. The table contains 41,188 rows and 21 columns.

Input Variables

There are 20 columns in the table that provide information about each client, such as age, marital status, and education level. A subset of these are related to the last contact of the current campaign, such as the month and day of the week the last contact was made as well as the number of days since the client was last contacted in a previous campaign. There are 10 columns in the table that are categorial, meaning that they contain textual values that correspond to a particular category for a given variable. Upon close examination we find that the input variables consists of the feature “duration” which does not make sense as one does not know the duration until a call has been completed - by which one could also tell certainly whether or not did the potential customer say yes or no.

Column Name	Description	Type
Age	Age of the client	Numeric
Job	Client's occupation	Categorial
marital	Marital status	Categorial
education	Client's education level	Categorial
default	Indicates whether the client has credit in default	Categorial

housing	Indicates whether the client has a housing loan	Categorial
loan	Indicates whether the client as a personal loan	Categorial
contact	Type of contact communication	Categorial
month	Month that last contact was made	Categorial
day_of_week	Day that last contact was made	Categorial
duration	Duration of last contact in seconds	Numeric
campaign	Number of contacts performed during this campaign for this client (including last contact)	Numeric
pdays	Number of days since the client was last contacted in a previous campaign	Numeric
previous	Number of contacts performed before this campaign for this client	Numeric
poutcome	Outcome of the previous marketing campaign	Categorial
emp.var.rate	Employment variation rate (quarterly indicator)	Numeric
cons.price.idx	Consumer price index (monthly indicator)	Numeric
cons.conf.idx	Consumer confidence index (monthly indicator)	Numeric
euribor3m	Euribor 3-month rate (daily indicator)	Numeric
nr.employed	Number of employees (quarterly indicator)	Numeric

Output Variables

There is one column in the table that corresponds to our target value.

Column Name	Description	Type
Y	Indicates whether the client has subscribed for a term deposit	Binary (yes or no)

Data Quality

As we can see below there are no missing values in the data set. Our dataset is complete.

Audit | Quality | Annotations

Complete fields (%): 100% Complete records (%): 100%

Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method	% Complete	Valid Records
age	Continuous	365	4	None	Never	Fixed	100	41188
job	Nominal	--	--	--	Never	Fixed	100	41188
marital	Nominal	--	--	--	Never	Fixed	100	41188
education	Nominal	--	--	--	Never	Fixed	100	41188
default	Nominal	--	--	--	Never	Fixed	100	41188
housing	Nominal	--	--	--	Never	Fixed	100	41188
loan	Nominal	--	--	--	Never	Fixed	100	41188
contact	Flag	--	--	--	Never	Fixed	100	41188
month	Nominal	--	--	--	Never	Fixed	100	41188
day_of_week	Nominal	--	--	--	Never	Fixed	100	41188
campaign	Continuous	565	304	None	Never	Fixed	100	41188
pdays	Continuous	0	1515	None	Never	Fixed	100	41188
previous	Continuous	754	310	None	Never	Fixed	100	41188
poutcome	Nominal	--	--	--	Never	Fixed	100	41188
emp.var.rate	Continuous	0	0	None	Never	Fixed	100	41188
cons.price.idx	Continuous	0	0	None	Never	Fixed	100	41188
cons.conf.idx	Continuous	0	0	None	Never	Fixed	100	41188
euribor3m	Continuous	0	0	None	Never	Fixed	100	41188
nr.employed	Continuous	0	0	None	Never	Fixed	100	41188
y	Flag	--	--	--	Never	Fixed	100	41188

Balance between Precision and Recall

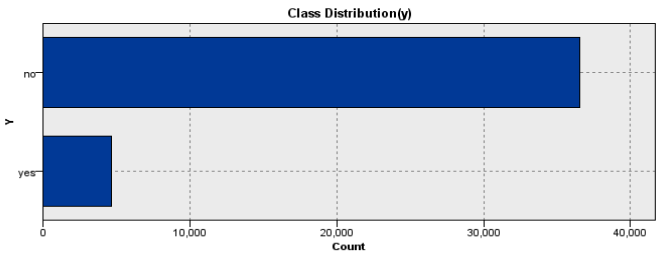


Fig xx

As seen in Fig xx, the dataset has 8 times as many negative candidates as the positive ones. Therefore, the aim of this modeling approach would be to reduce enable the bank to be able to correctly identify one customer correctly out of every four calls made, thus ensuring a 50%

decrease in total calls made to reach the same number of interested customers. As a result, we attach a value of 4 to β thus establishing that Recall is four times as important as Precision.

5. Algorithms and Results

5.1 Decision Tree

Decision trees are a very popular tool for predictive analytics because they are relatively easy to use, perform well with non-linear relationships and produce highly interpretable output. For our analysis we will be comparing two most popular Decision tree algorithms: CART and C5.0.

CART

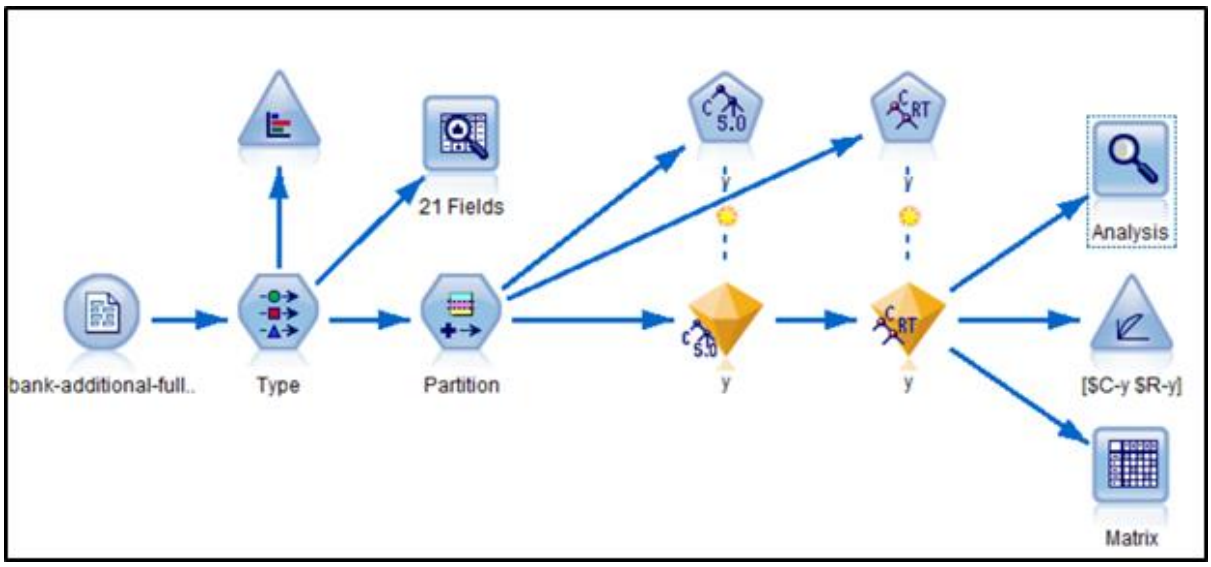
CART, or Classification and Regression Trees, uses an algorithm which proceeds recursively, successively splitting the data set into smaller segments. It uses the Gini rule to define splits, supports binary splits only and identifies the best binary split for complex categorical or continuous predictors. This algorithm also prunes the tree by testing it against an independent (validation) data set or through n-fold cross-validation. It works with both categorical and continuous data. For our analysis, we have enabled Misclassification cost and assigned 2.0 for False Negative outcomes since this would mean loss of potential customers for the bank.

C5.0

C5.0 is a tree-learning algorithm developed by Ross Quinlan, an Australian computer science researcher. This algorithm uses the entropy or information gain measures to define splitting rules. C5.0 works with both categorical and continuous variables and handles missing data. The algorithm also includes a pruning function. Misclassification cost for False Negatives have been set to 2.0. Also, to improve efficiency, we have enabled Expert mode. Moreover, to help prevent overtraining with noisy data, we have set Min. records per Child Node to 10.

Analysis

Figure x. shows the SPSS Stream generated for the Decision Trees (C5.0 and CART).



Below chart illustrates the importance of attributes with respect to models CART and C5.0 . Nr.employed(0.4 by C5.0 and 0.62 by CART) is the most important for three examined models followed by pdays (0.25 by C5.0 and 0.18 by CART). Some of the attributes like Housing, job etc. have been removed because they are trivial or not has any degree of importance with respect these models.

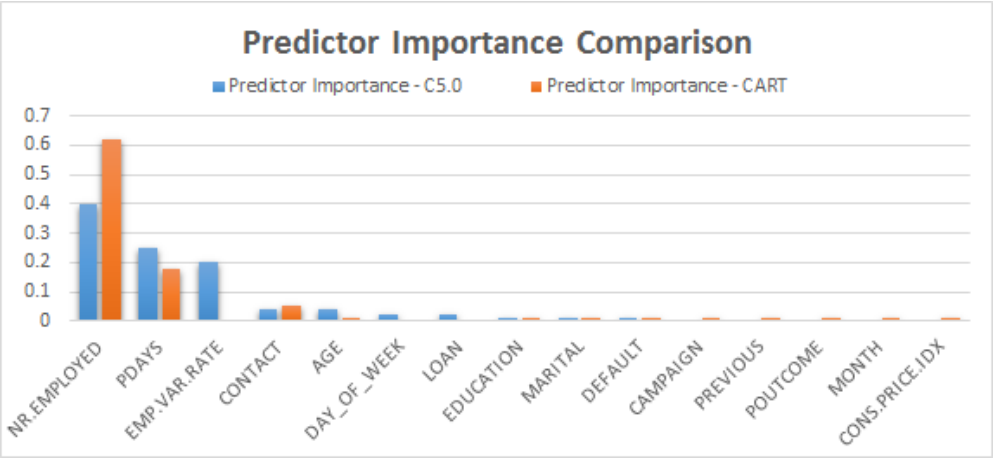


Fig 2: Comparing Predictor Importance of C5.0 and CART

Comparing $C-y with y

'Partition'	1_Training		2_Testing	
Correct	24,793	89.82%	12,230	90.03%
Wrong	2,810	10.18%	1,355	9.97%
Total	27,603		13,585	

Coincidence Matrix for $C-y (rows show actuals)

'Partition' = 1_Training	no	yes
no	23,297	1,164
yes	1,646	1,496

'Partition' = 2_Testing	no	yes
no	11,495	592
yes	763	735

Fig 3: Confusion Matrix for C5.0

Comparing $R-y with y

'Partition'	1_Training		2_Testing	
Correct	24,323	88.12%	12,016	88.45%
Wrong	3,280	11.88%	1,569	11.55%
Total	27,603		13,585	

Coincidence Matrix for $R-y (rows show actuals)

'Partition' = 1_Training	no	yes
no	22,926	1,535
yes	1,745	1,397

'Partition' = 2_Testing	no	yes
no	11,337	750
yes	819	679

Fig 4: Confusion Matrix for CART

The predictions of both models are compared to the original classes to construct the confusion matrix and derive values of Statistical parameters - Accuracy, Recall, Precision and the F Score as tabulated in below table.

Model	Partition	Accuracy	Recall	Precision	*F*4Score
C5.0	Training	89.82%	47.61%	56.24%	0.48
	Testing	90.03%	49.07%	55.39%	0.49
CART	Training	88.12%	44.46%	47.65%	0.45
	Testing	88.45%	45.33%	47.52%	0.45

Fig 5: Comparing Recall, Precision and Accuracy for C5.0 and CART

We can see that overall Accuracy of the models do not vary widely. On the other hand, Recall in case of C5.0 is higher for both Training and Testing datasets as compared to that of CART. This essentially means that our C5.0 model is better at predicting potential customers who will end up buying Term Deposit. Precision for C5.0 is also higher than that of CART. This shows that our model is not wrongly predicting all records as true positive.

We also analysed our models based on the cumulative Gain charts for training and testing subsets. The higher lines indicate better models, especially on the left side of the chart. The

two curves are same for the test subset and almost identical to the training one. Also, we can see that until about 15 percentile of training and testing subsets, both C5.0 and CART follow the same trajectory and thereafter C5.0 reaches closer to the best fit line leaving CART line behind.

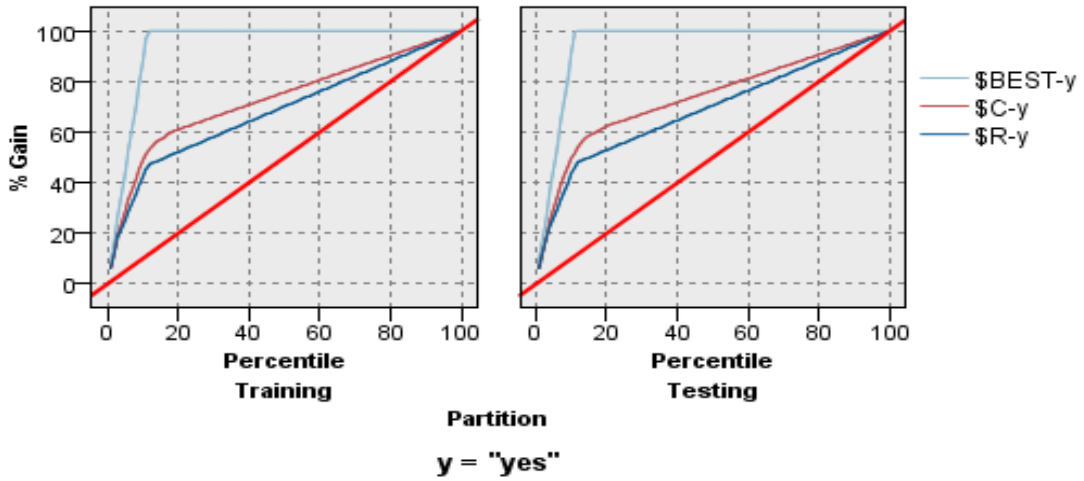


Fig 6: Comparison using Gain Chart

Therefore, from the previews we can conclude that among both the models C5.0 is better at classifying whether a client will sign up for Term Deposit or not. However, since the statistics for these two are not very high we'll look into some more algorithms in the next section.

5.2 Support Vector Machines

Support Vector Machines (SVM) are supervised learning models used for classification and regression tasks. They are widely known for their efficient performance while operating on linear datasets which also extends to non-linear datasets using the kernel trick. To achieve the objective of classification, SVM uses the concept of a hyperplane to differentiate between two or more sets of data. For any given set of data, one or more hyperplanes exist which might classify the data. SVM then selects the maximum margin hyperplane to achieve the required level of accuracy while classifying data.

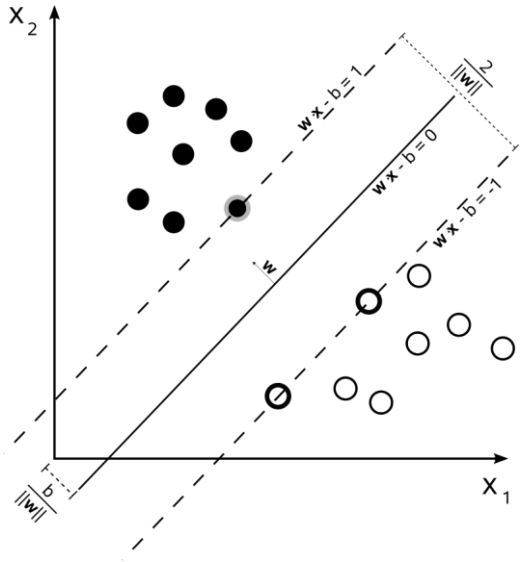


Figure xx

As mentioned earlier, there are a number of kernel “tricks” or functions which can be used to transform non-linear separations to linear ones. Some of the important ones are as follows:

1. Linear: This is the function which is used in SVM by default. Here, we assume that the non linear relationships in the data. The function is represented as

$$K(x_i, x_j) = x_i . x_j$$

2. Polynomial: The polynomial function represents the vectors allow learning of models using polynomials of the original features. The function is defined as:

$$K(x_i, x_j) = (x_i . x_j + 1)^{p}$$

3. Radial: The radial function fits highly non linear data by making use of exponentials powers, similar to a polynomial function. An example is shown in figure xx. The function is defined as:

$$K(x_i, x_j) = e^{(-(x_i - x_j)^2 / s^2)}$$

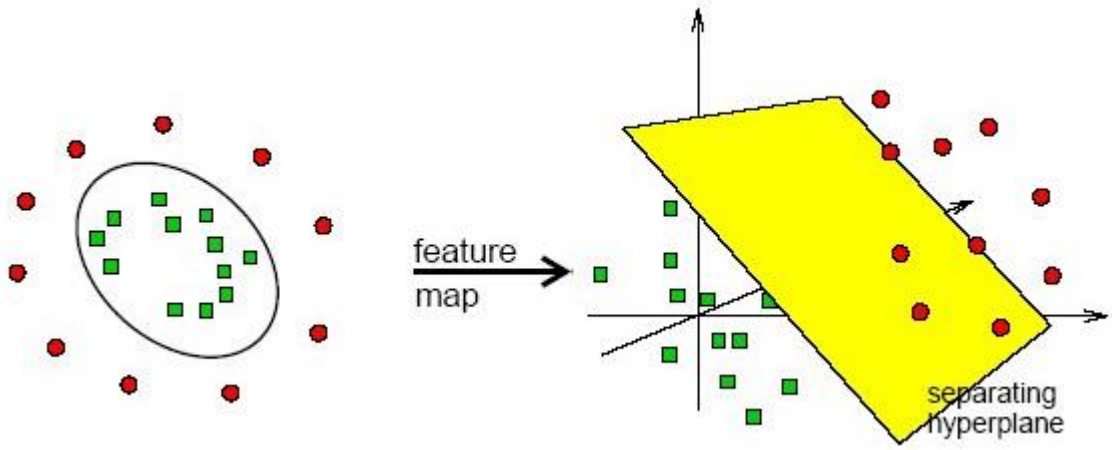


Figure xx

4. Sigmoid: The sigmoid function models the training data into a way similar to a two layer neural network. This is also very good in fitting highly non linear data. The function is represented as:

$$K(x_i, x_j) = tanh(s(x_i . x_j) + c)$$

Data Preprocessing

The analysis is done using all of the kernel models presented above. Since we had around 45,000 datasets, we adopted a 90:10 ratio for training and testing data. A Partition node with the above mentioned ratio was added to the stream. Upon close investigation of the data, it was seen that the number of "no" are far greater than "yes". As a result, a balance node was used to

train the SVM algorithm with equal number of “yes” and “no” cases. To achieve this purpose, a balance directive with a Factor of 0.12 and Condition y = “no” was added to the Balance node. As instructed in the UCI Machine Learning repositories, the feature “duration” was not used since it cannot be calculated unless a call has been completed, by when one can tell without the use of a data model what the response of the customer is.

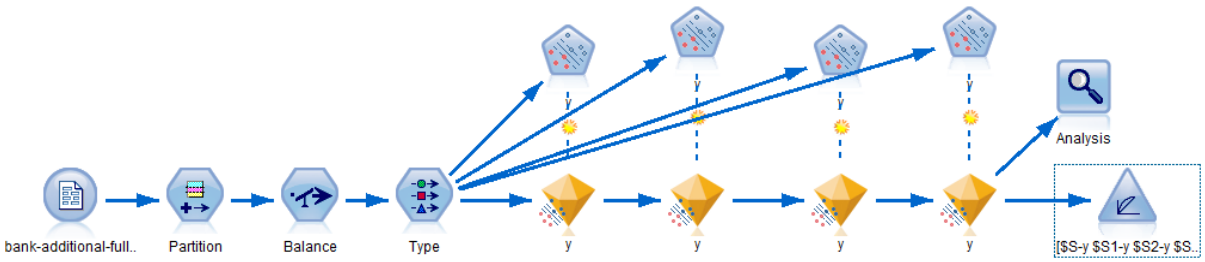


Figure xx

Stream Setup

Each of the models were individually optimized before combining them together as can be seen in Figure xx. The stopping criteria for all the models was 0.01. The optimum values for parameters specific to each of the kernel models were obtained after extensive iterative testing and are shown in Table xx.

	Linear	Polynomial (Degree = 3)	Radial Basis	Sigmoid
Regularization Parameter	10	10	10	10
Gamma	NA	0.1	0.3	0.2

Results

With parameters as mentioned in the table above, the stream was run and a number of results were obtained. The Predictor Importance chart shows that in the absence of the “duration” feature, emp.var.rate is the more important feature, closely followed by nr.employed.

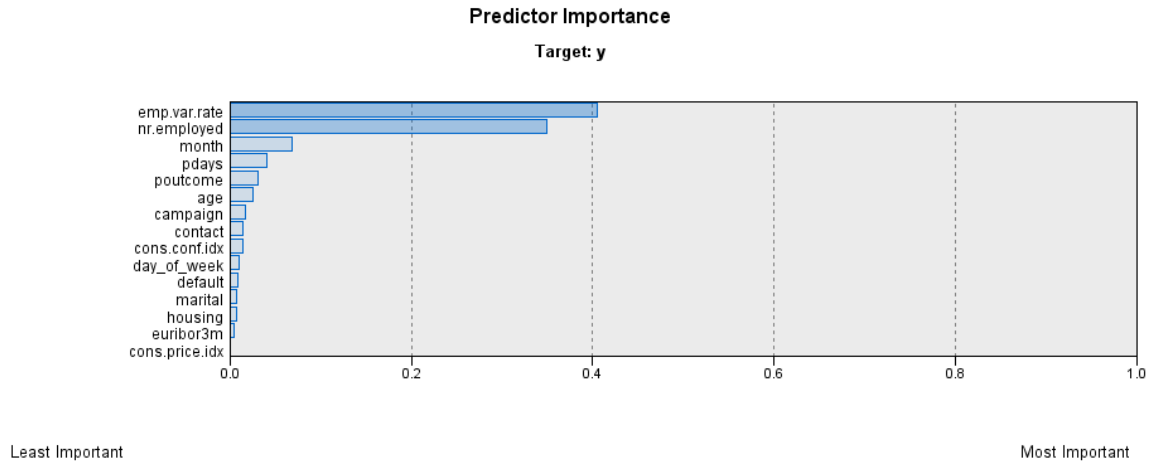


Fig xx

The confusion matrices generated from the testing data for each of the models are as follows:

	Predicted True	Predicted False	
Radial	313	157	Actual True
	1364	2258	Actual False
Polynomial	282	188	Actual True
	1563	2059	Actual False
Sigmoid	273	197	Actual True
	998	2624	Actual False
Linear	241	229	Actual True
	473	3149	Actual False

Analysis

Since the goal here is to maximise the number of true positive cases while minimizing the number of positive predictions, Precision and Recall are the two major considerations for us. They are calculated and shown for each model in Table xx

Model	Accuracy	Precision	Recall	F_4Score
Radial Basis	0.63	0.19	0.67	0.58
Polynomial	0.57	0.15	0.60	0.51
Sigmoid	0.71	0.21	0.58	0.53
Linear	0.82	0.34	0.51	0.50

From the table above, it can be clearly seen that Recall is very high for the Radial Basis function with Precision equalling the other data sets. Therefore, the Radial Basis function is the most appropriate of all among the various SVM models.

5.3 Logistic Regression (Stepwise Regression Method)

Logistic regression is a regression model where the dependent/target variable is categorical.

Logistic regression was developed David Cox in 1958. The logistic model is based on a binary target variable and estimates the probability of a dependent binary variable which is based on one or more predictor/independent variables.

Logistic regression estimates the probabilities using a logistic function, which is the cumulative logistic distribution and is thus able to establish the relationship between the categorical dependent variable and one or more independent variables.

Logistic regression is considered a special type of general linear regression model. But there are differences between logistic regression and linear regression model. The assumptions considered in case of the relationship between dependent and independent variables for logistic regression are different from those considered in linear regression. Two particular differences are seen.

The first difference is that the conditional distribution y/x is a Bernoulli distribution for logistic regression rather than a Gaussian distribution in case of a linear regression. This is because the target/dependent variable is a binary response.

The second difference is that the predicted values are probabilities and as such can take values (0,1) based on the logistic function using cumulative logistic distribution as logistic regression predicts the probability of particular outcomes. Logistic regression also makes no assumption on the distribution of the independent variables.

Stepwise regression is a process that is semi-automated way of constructing a model which proceeds by succesive adding or removing variables on the basis of their t-statistics of their estimated coefficients.

Method :

In every prediction model there is a set of potential independent variables from which we to try to extract the best subset to use for prediction.

The stepwise regression allows us to either begin with zero variables in the model and move forward (forward selection) and at each step adding one variable or begin with all the potential variables in the model and move backwards (backward selection) and at each step removing one variable.

During each step, the algorithm performs the following calculations:

For each variable not present in the model in the forward move, algorithm calculates the t-statistic that the variable's coefficient would have if it were the next variable added and then squares it and reports this as its 'F to enter' statistic.

For each variable present in the model in case of the backward move, algorithm calculates the t-statistic for the estimated coefficient of the variable and then squares it and reports this as its 'F to remove' statistic.

At the next step, the algorithm automatically enters the variable with the highest F-to-enter statistic (forward selection), or removes the variable with the lowest F to remove statistic (backward selection), in accordance with certain control parameters specified i.e. the threshold value for F to enter or F to remove.

Method of choice :

To choose forward or backward selection method we need to examine the set of potential independent variables from which we want to extract the variables to be used in our model. If there is a very large set of potential independent variables from which we would like to consider a few to be used in our model then we should choose forward. If, on the other hand, if we have a mid-sized set of potential variables from which we want to remove a few we should choose backward.

In our model, presence of a mid -sized set of variables allows us to choose backward selection procedure. However we have used both procedures to check the differences between the two procedures for the correct/wrong determination of training/testing data.

Partitioning of data with 67% training and 33% testing data. Upon close investigation of the data, it was seen that the number of "no" are far greater than "yes". As a result, a second model was created where balance node was used to train the algorithm with equal number of "yes" and "no" cases. To achieve this purpose, a balance directive with a Factor of 0.12 and Condition y = "no" was added to the Balance node.

Base category for target kept as no in both forward and backward selection procedures, due to very high number of "no" versus "yes" and to find the equation to predict "yes" value for term deposit.

Model Layout

A) Original model layout

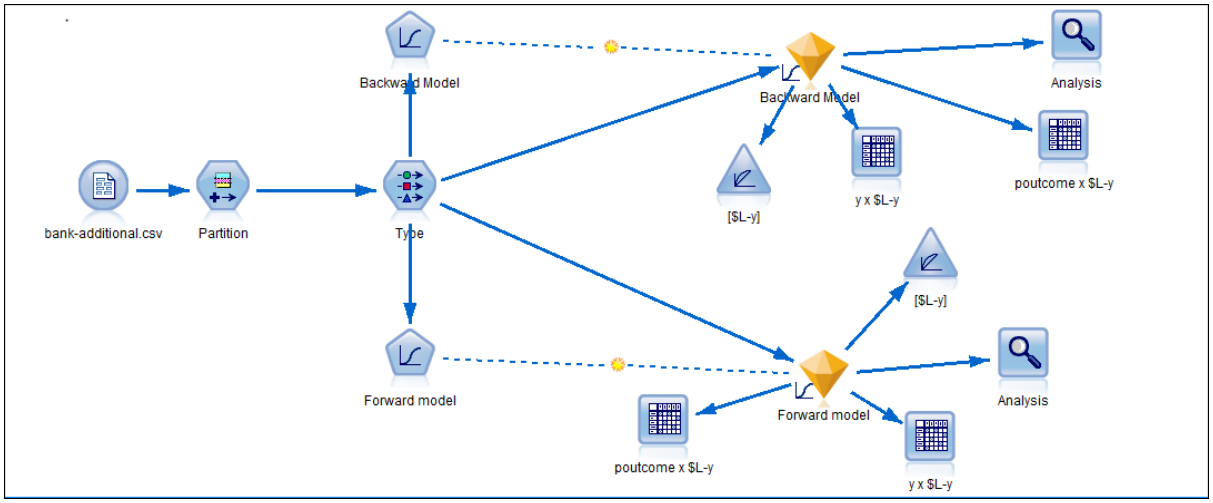


Figure: Logistic Regression With Backward and Forward Selection procedures for variables

B) New model layout with balance node

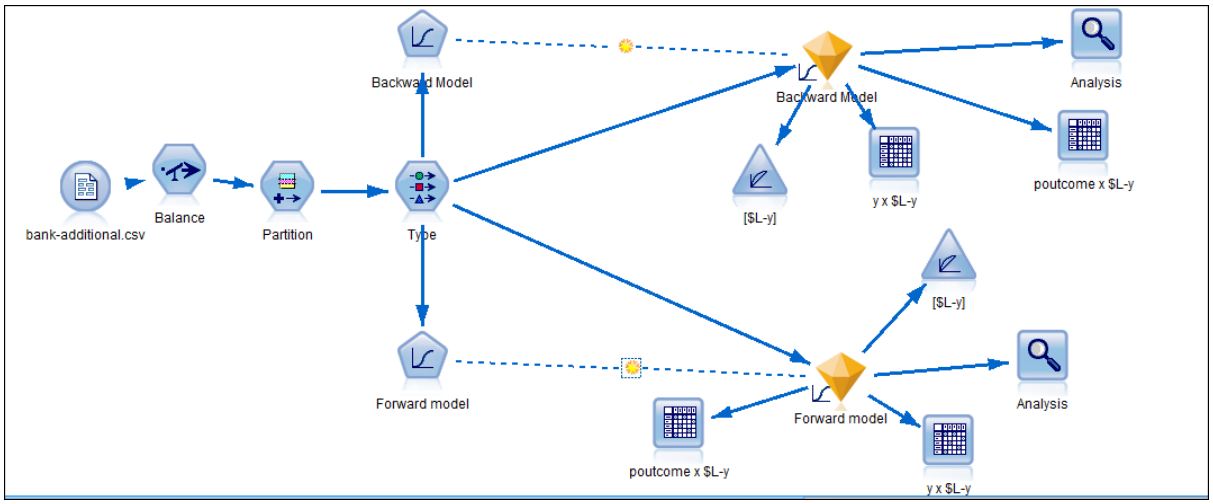
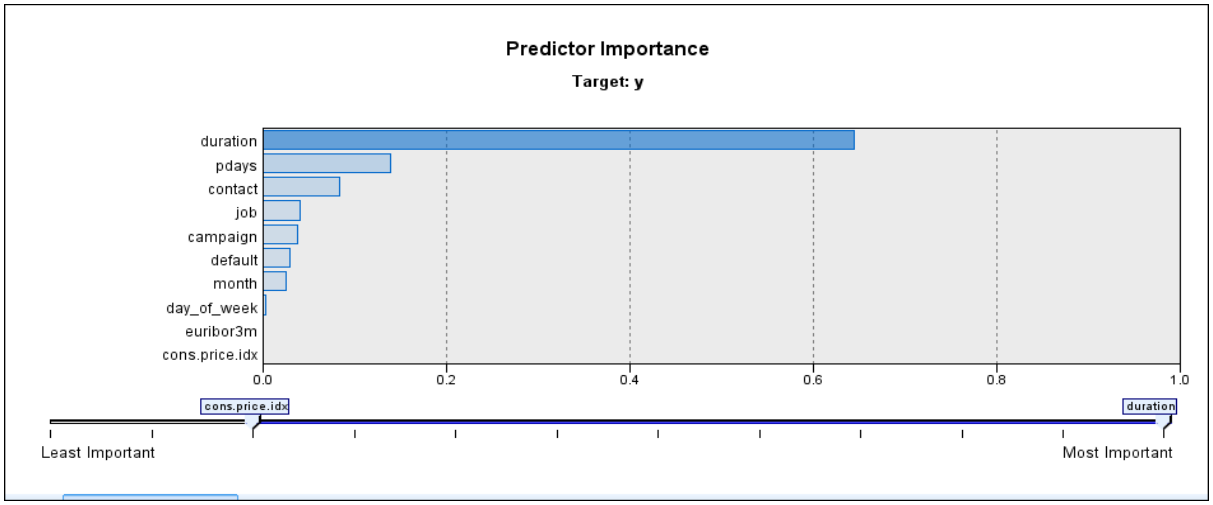


Figure: Logistic Regression With Backward and Forward Selection procedures for variables containing balance node

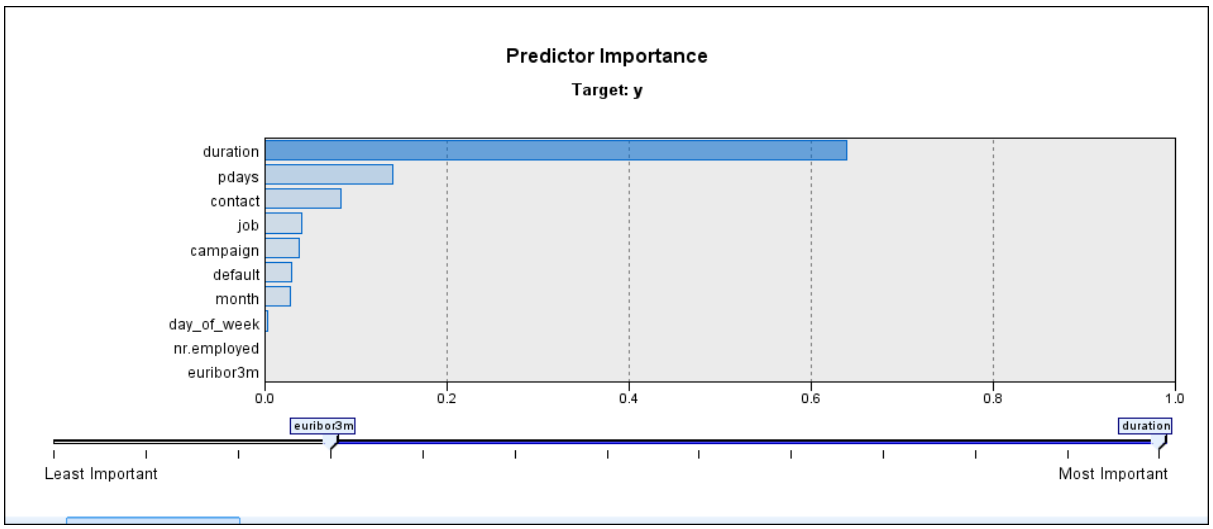
Below chart illustrates the importance of attributes with respect to logistic regression models with backward and forward selection procedures for variables.

Original Model layout

Backward Selection :



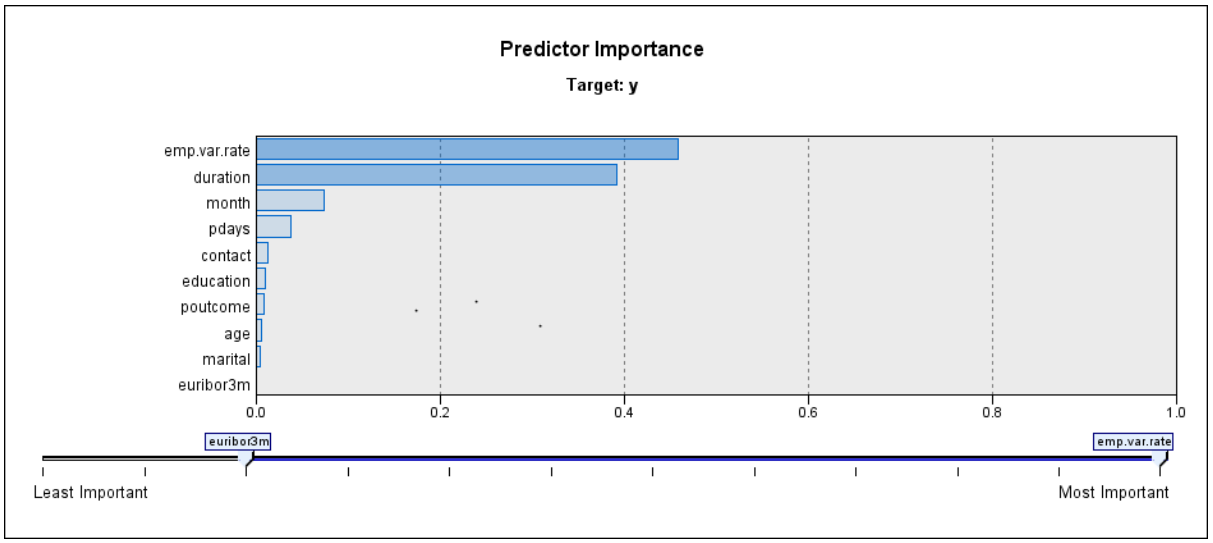
Forward Selection:



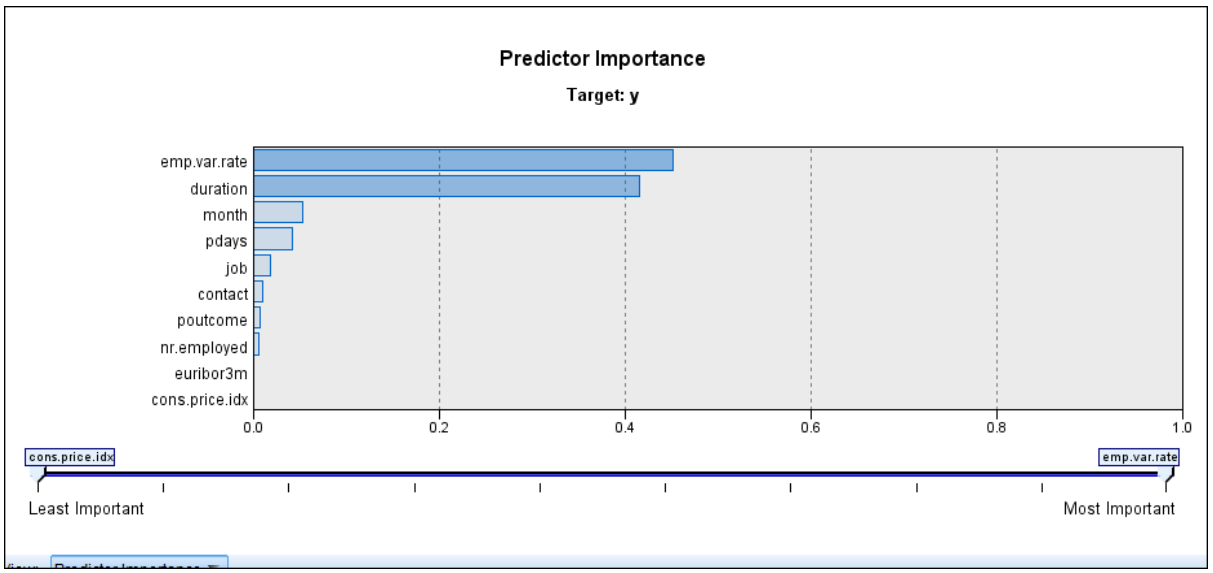
Both show similar predictor importance which is consequent of modest sized set of independent variables. Clearly , duration is the most important predictor followed by pdays , contact , job , campaign , default and others.

New model with balance node :

Backward Selection:



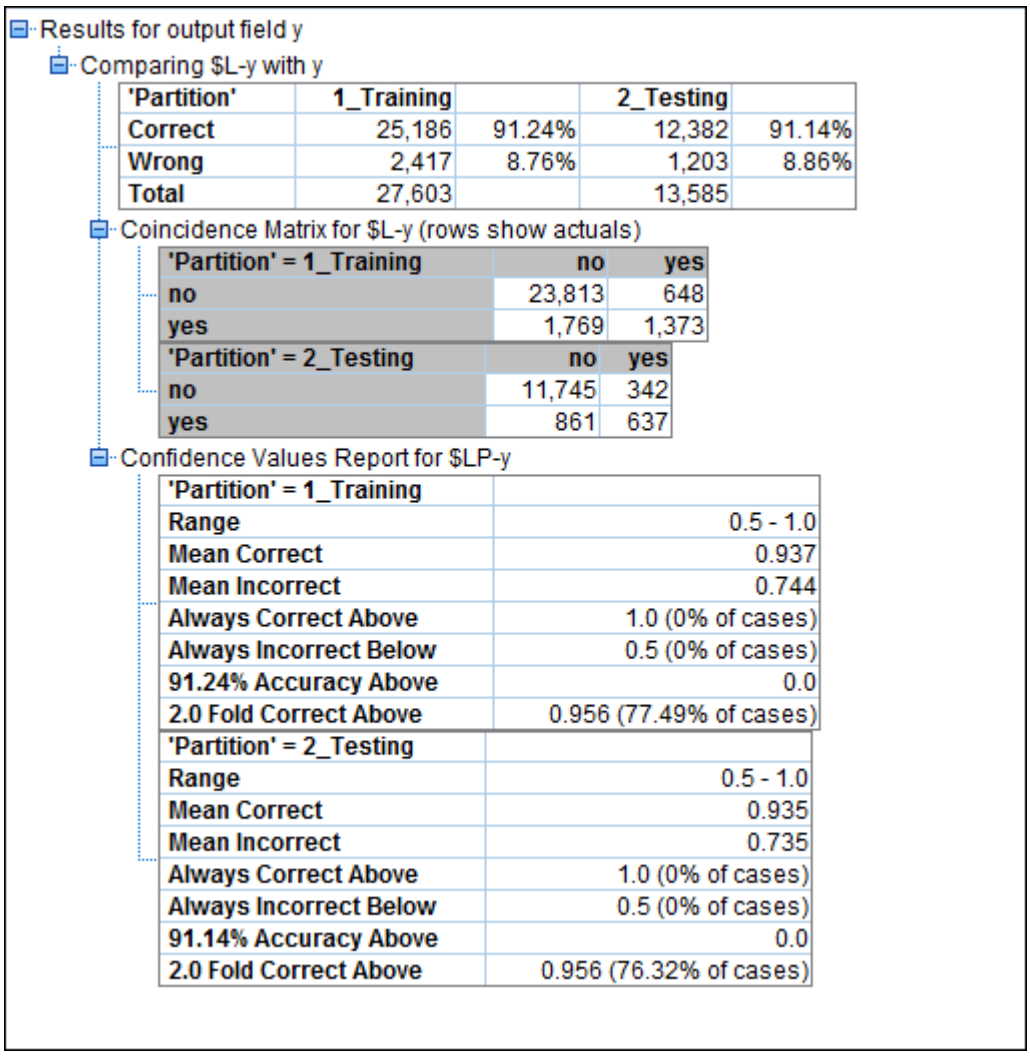
Forward Selection :



Both show similar predictor importance which is consequent of modest sized set of independent variables. Clearly, different from the original model, as here emp.var.rate is the most important predictor followed by duration , month , pdays and others.

Original model layout

Backward Selection :



Results for output field y

Comparing $L-y with y

'Partition'	1_Training		2_Testing	
Correct	25,186	91.24%	12,382	91.14%
Wrong	2,417	8.76%	1,203	8.86%
Total	27,603		13,585	

Coincidence Matrix for $L-y (rows show actuals)

'Partition' = 1_Training	no	yes
no	23,813	648
yes	1,769	1,373

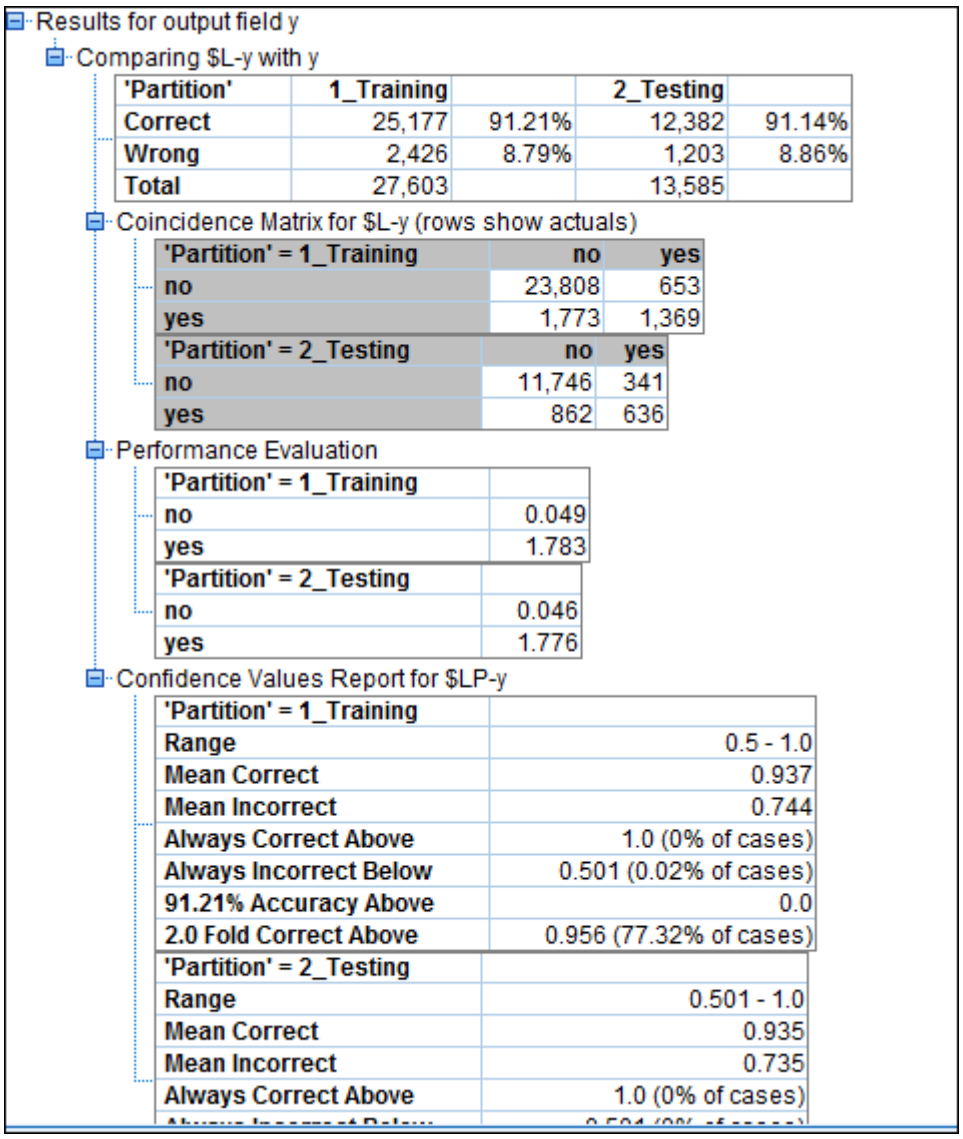
'Partition' = 2_Testing	no	yes
no	11,745	342
yes	861	637

Confidence Values Report for $LP-y

'Partition' = 1_Training	
Range	0.5 - 1.0
Mean Correct	0.937
Mean Incorrect	0.744
Always Correct Above	1.0 (0% of cases)
Always Incorrect Below	0.5 (0% of cases)
91.24% Accuracy Above	0.0
2.0 Fold Correct Above	0.956 (77.49% of cases)

'Partition' = 2_Testing	
Range	0.5 - 1.0
Mean Correct	0.935
Mean Incorrect	0.735
Always Correct Above	1.0 (0% of cases)
Always Incorrect Below	0.5 (0% of cases)
91.14% Accuracy Above	0.0
2.0 Fold Correct Above	0.956 (76.32% of cases)

Forward Selection :



Results for output field y

Comparing $L-y with y

'Partition'	1_Training		2_Testing	
Correct	25,177	91.21%	12,382	91.14%
Wrong	2,426	8.79%	1,203	8.86%
Total	27,603		13,585	

Coincidence Matrix for $L-y (rows show actuals)

'Partition' = 1_Training	no	yes
no	23,808	653
yes	1,773	1,369

'Partition' = 2_Testing	no	yes
no	11,746	341
yes	862	636

Performance Evaluation

'Partition' = 1_Training	
no	0.049
yes	1.783

'Partition' = 2_Testing	
no	0.046
yes	1.776

Confidence Values Report for $LP-y

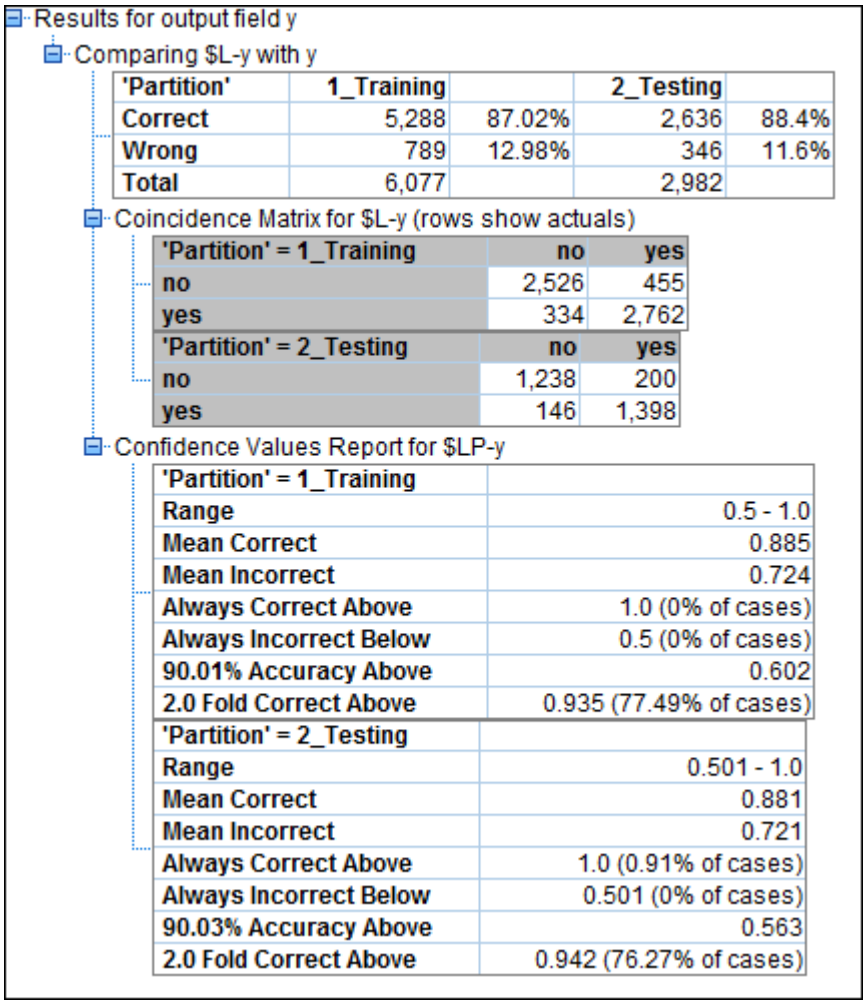
'Partition' = 1_Training	
Range	0.5 - 1.0
Mean Correct	0.937
Mean Incorrect	0.744
Always Correct Above	1.0 (0% of cases)
Always Incorrect Below	0.501 (0.02% of cases)
91.21% Accuracy Above	0.0
2.0 Fold Correct Above	0.956 (77.32% of cases)

'Partition' = 2_Testing	
Range	0.501 - 1.0
Mean Correct	0.935
Mean Incorrect	0.735
Always Correct Above	1.0 (0% of cases)

In both cases the training and testing data accuracy is high of over 90% with forward selection showing slightly less values. The high accuracy is due to presence of large number of" no" versus "yes". If we see the number "no" values correctly predicted, it is very high but in case of "yes" it is very low so model is not very good.

New model layout with balance node

Backward Selection :



Results for output field y

Comparing $L-y with y

'Partition'	1_Training		2_Testing	
Correct	5,288	87.02%	2,636	88.4%
Wrong	789	12.98%	346	11.6%
Total	6,077		2,982	

Coincidence Matrix for $L-y (rows show actuals)

'Partition' = 1_Training	no	yes
no	2,526	455
yes	334	2,762

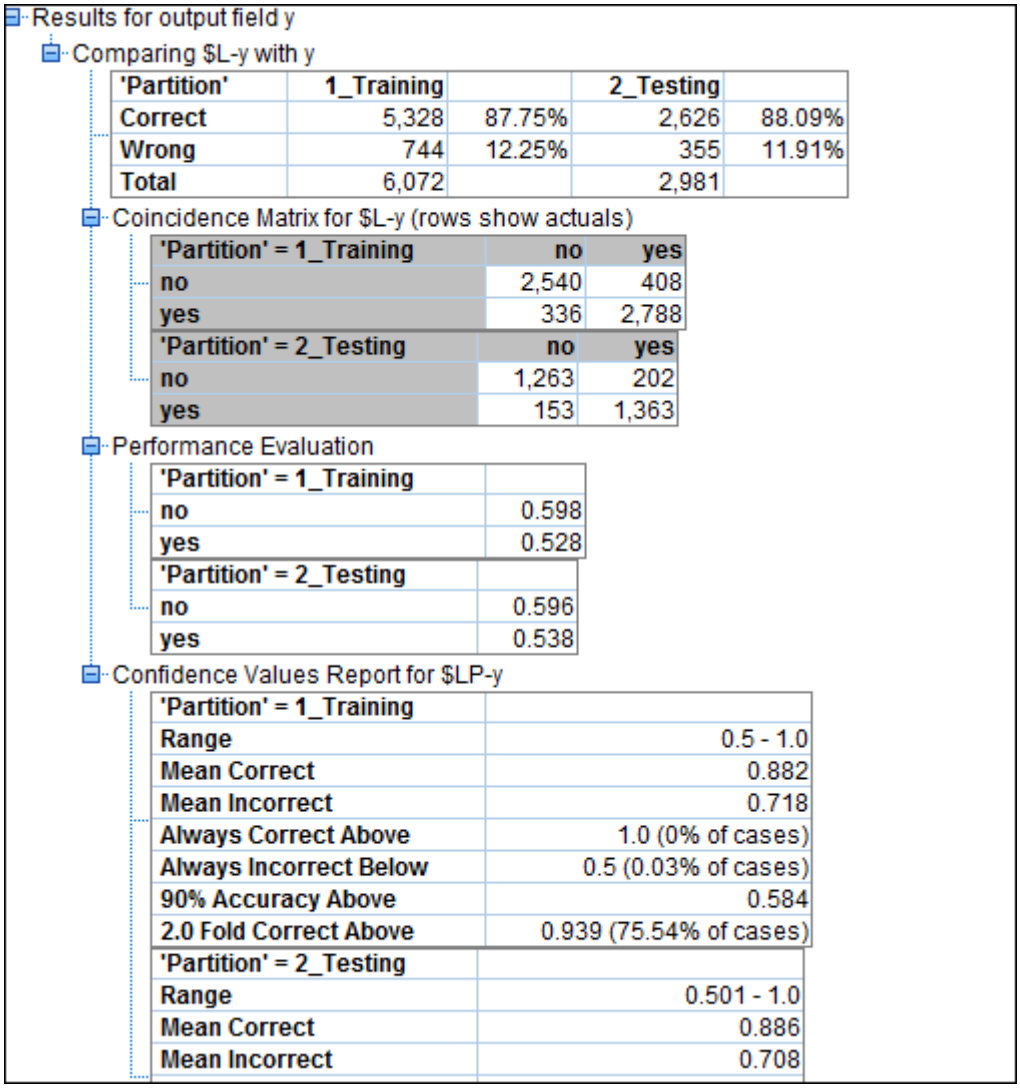
'Partition' = 2_Testing	no	yes
no	1,238	200
yes	146	1,398

Confidence Values Report for $LP-y

'Partition' = 1_Training	
Range	0.5 - 1.0
Mean Correct	0.885
Mean Incorrect	0.724
Always Correct Above	1.0 (0% of cases)
Always Incorrect Below	0.5 (0% of cases)
90.01% Accuracy Above	0.602
2.0 Fold Correct Above	0.935 (77.49% of cases)

'Partition' = 2_Testing	
Range	0.501 - 1.0
Mean Correct	0.881
Mean Incorrect	0.721
Always Correct Above	1.0 (0.91% of cases)
Always Incorrect Below	0.501 (0% of cases)
90.03% Accuracy Above	0.563
2.0 Fold Correct Above	0.942 (76.27% of cases)

Forward Selection :



Results for output field y

Comparing $L-y with y

'Partition'	1_Training		2_Testing	
Correct	5,328	87.75%	2,626	88.09%
Wrong	744	12.25%	355	11.91%
Total	6,072		2,981	

Coincidence Matrix for $L-y (rows show actuals)

'Partition' = 1_Training	no	yes
no	2,540	408
yes	336	2,788

'Partition' = 2_Testing	no	yes
no	1,263	202
yes	153	1,363

Performance Evaluation

'Partition' = 1_Training	
no	0.598
yes	0.528

'Partition' = 2_Testing	
no	0.596
yes	0.538

Confidence Values Report for $LP-y

'Partition' = 1_Training	
Range	0.5 - 1.0
Mean Correct	0.882
Mean Incorrect	0.718
Always Correct Above	1.0 (0% of cases)
Always Incorrect Below	0.5 (0.03% of cases)
90% Accuracy Above	0.584
2.0 Fold Correct Above	0.939 (75.54% of cases)

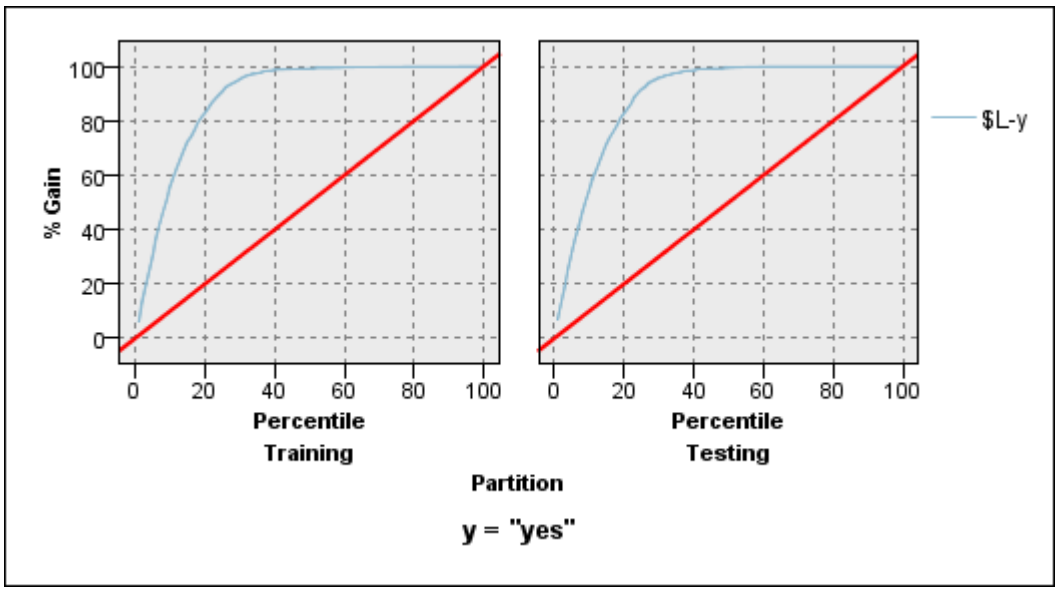
'Partition' = 2_Testing	
Range	0.501 - 1.0
Mean Correct	0.886
Mean Incorrect	0.708

In both cases the training and testing data accuracy is now lower at over 85 % with forward selection showing slightly higher values for training data. Here, although correctly predicted is lower by about 4.0% compared to original model, it is an acceptable difference as over 85% correct prediction is high, but we see that the number of “yes” correctly predicted is far higher than in original model , so this model is better.

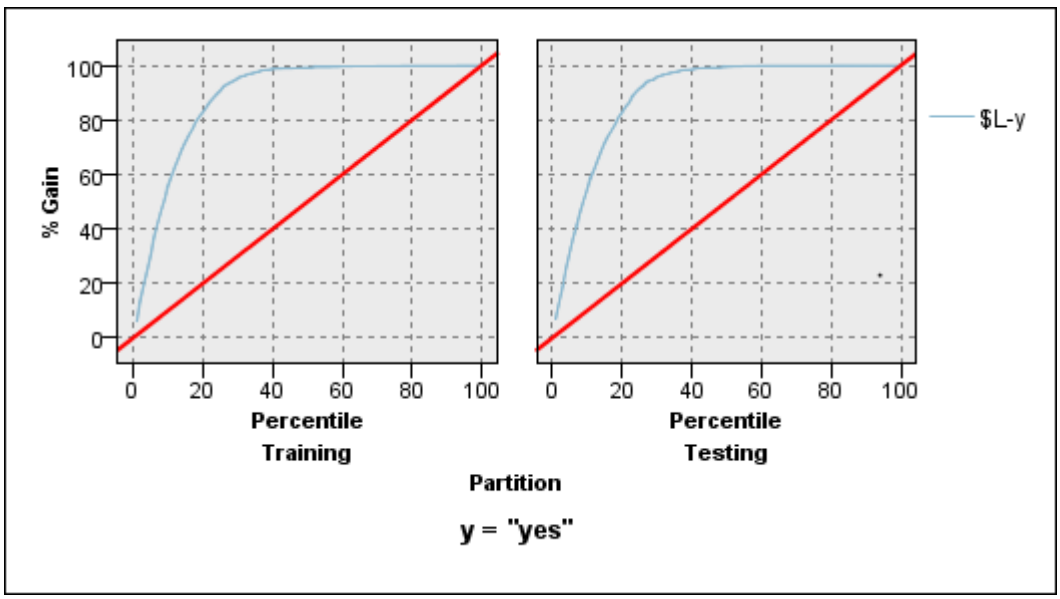
Analysis of Gain

Original Model :

Backward Selection :



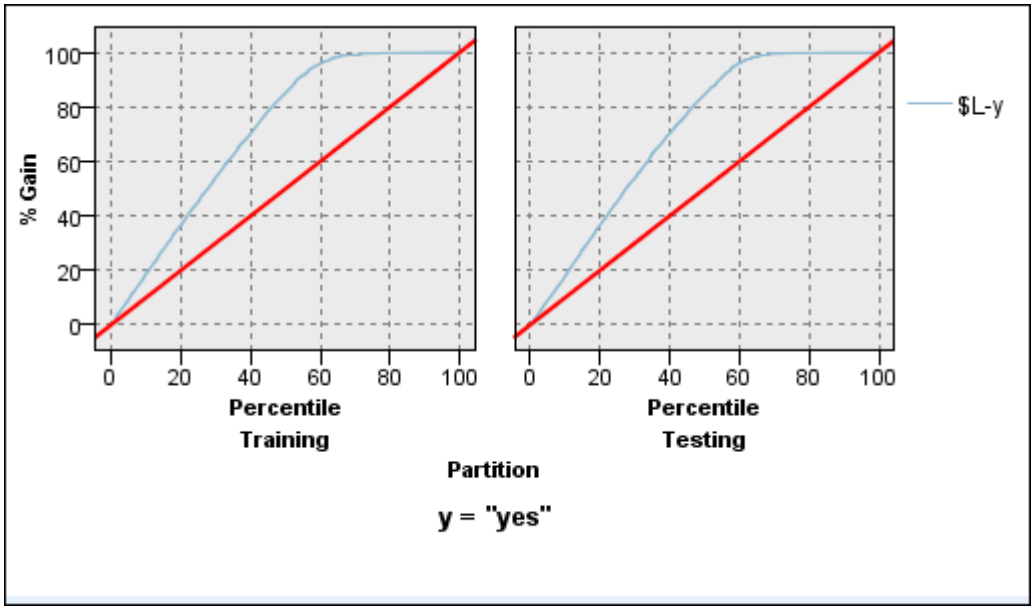
Forward Selection :



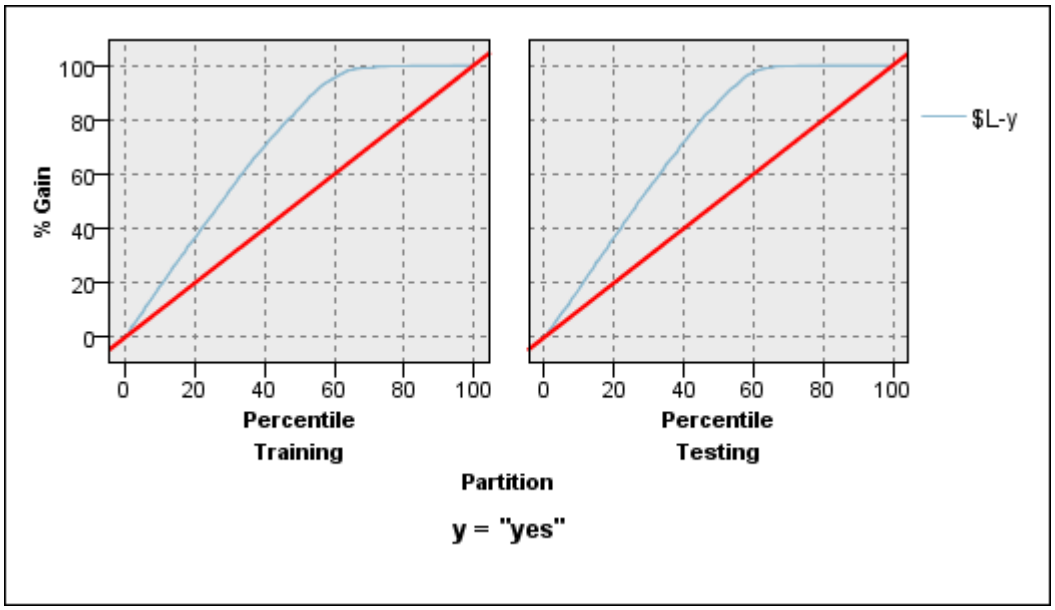
Both show similar gain as difference in testing and training data correctly predicted set is very low.

New Model with balance node:

Backward Selection:



Forward Selection:



Here, both gains show similar characteristics for both backward and forward due to small difference in correctly predicted in training and testing data. However, this is different from original layout as it had higher correct prediction.

Generated Model Equation

Original Model

Backward Selection:

Equation For yes

0.2919 * [job=admin.] + 0.02886 * [job=blue-collar] + 0.2051 * [job=entrepreneur] + 0.2375 * [job=housemaid] + 0.3118 * [job=management] + 0.5013 * [job=retired] + 0.1746 * [job=self-employed] + 0.1132 * [job=services] + 0.4773 * [job=student] + 0.2918 * [job=technician] + 0.4015 * [job=unemployed] + 11.09 * [default=no] + 10.7 * [default=unknown] + 0.5681 * [contact=cellular] + (-0.231) * [month=apr] + 0.602 * [month=aug] + (-0.02905) * [month=dec] + (-0.07337) * [month=jul] + (-0.654) * [month=jun] + 1.756 * [month=mar] + (-0.7688)* [month=may] + (-0.8006) * [month=nov] + (-0.2057)* [month=oct] + (-0.1723)* [day_of_week=fri] + (-0.2933) * [day_of_week=mon] + (-0.1574) * [day_of_week=thu] + (-0.1052) * [day_of_week=tue] + 0.004661 * duration + (-0.06036) * campaign + (-0.001303) * pdays + (-0.5451) * [poutcome=failure] + (-0.02827)* [poutcome=nonexistent] +(-1.721) * emp.var.rate + 1.804 * cons.price.idx + 0.5999 * euribor3m + (-184.8)

Forward Selection:

Equation For yes

0.2915 * [job=admin.] + 0.02674 * [job=blue-collar] + 0.2032 * [job=entrepreneur] + 0.2385 * [job=housemaid] + 0.3111 * [job=management] + 0.502 * [job=retired] + 0.1738 * [job=self-employed] + 0.1122 * [job=services] + 0.4764 * [job=student] + 0.2922 * [job=technician] + 0.4022 * [job=unemployed] + 11.09 * [default=no] + 10.7 * [default=unknown] + 0.5682 * [contact=cellular] + (-0.3095)* [month=apr] + 0.5747 * [month=aug] + (-0.05613) * [month=dec] + (-0.14)* [month=jul] + (-0.7616) * [month=jun] + 1.723 * [month=mar] + (-0.8273) * [month=may] + (-0.8479) * [month=nov] + (-0.2274) * [month=oct] + (-0.1735) * [day_of_week=fri] + (-0.2937) * [day_of_week=mon] + (-0.1585)* [day_of_week=thu] + (-0.1044)* [day_of_week=tue] + 0.004662 * duration + (-0.06044) * campaign + (-0.001306) * pdays + (-0.5459) * [poutcome=failure] + (-0.02927) * [poutcome=nonexistent] + (-1.775) * emp.var.rate + 1.912 * cons.price.idx + 0.5704 * euribor3m + 0.001354 * nr.employed + (-201.7)

Some changes observed in coefficients of the variables as well as variables used in backward and forward selection.

New Model with balance node:

Backward Selection:

Equation For yes

0.009015 * age + -0.4335 * [marital=divorced] + -0.2987 * [marital=married] + (-0.08703)* [marital=single] + (-0.3547)* [education=basic.4y] +(-0.1619) * [education=basic.6y] + (-0.2857)* [education=basic.9y] +(-0.06979)* [education=high.school] + 1.055 * [education=illiterate] + (-0.001015)* [education=professional.course] + 0.2155 * [education=university.degree] + 0.312 * [contact=cellular] + 0.3528 * [month=apr] + 0.9844 * [month=aug] + (-0.4973) * [month=dec] + 0.2029 * [month=jul] +(-0.5145)* [month=jun] + 2.545 * [month=mar] +(-0.8331)* [month=may] + (-0.4639) * [month=nov] + 1.002 * [month=oct] +(-0.3138) * [day_of_week=fri] +(-0.298) * [day_of_week=mon] + (-0.2773)* [day_of_week=thu] +(-0.1677) * [day_of_week=tue] + 0.007244 * duration + (-0.001086)* pdays +(-0.8501)* [poutcome=failure] +(-0.2576)* [poutcome=nonexistent] + (-1.867)* emp.var.rate + 1.691 * cons.price.idx + 0.5374 * euribor3m + (-161.9)

Forward Selection :

Equation For yes

(-0.3608) * [job=admin.] +(-0.8896) * [job=blue-collar] +(-0.7979) * [job=entrepreneur] + (-0.4283)* [job=housemaid] +(-0.4969)* [job=management] +(-0.4023) * [job=retired] + (-0.6135) * [job=self-employed] +(-0.8624) * [job=services] +(-0.2238) * [job=student] + (-0.4892) * [job=technician] +(-0.4143) * [job=unemployed] + 0.317 * [contact=cellular] + 0.3282 * [month=apr] + 0.9026 * [month=aug] +(-0.5361) * [month=dec] + 0.1641 * [month=jul] + (-0.4972) * [month=jun] + 2.464 * [month=mar] +(-0.8465) * [month=may] +(-0.4672)* [month=nov] + 0.9861 * [month=oct] + 0.007232 * duration + (-0.001059) * pdays +(-0.8753) * [poutcome=failure] + (-0.2784)* [poutcome=nonexistent]

+(-1.816) * emp.var.rate +1.619 * cons.price.idx + 0.527 * euribor3m + (-0.0002108) * nr.employed + (-153.7)

Some changes observed in coefficients of the variables as well as variables used in backward and forward selection. The new model equations for backward and forward are completely different from the original model. The equations of this model are to be considered superior to original model due to their superior predictive power in correctly predicting “yes” as seen earlier in analysis.

Performance Metrics:

Original model

Backward Selection:

$L-y

y	no	yes
no	35558	990
yes	2630	2010

Forward Selection:

$L-y

y	no	yes
no	35554	994
yes	2635	2005

New Model with balance node

Backward Selection:

$L-y

y	no	yes
no	3772	684
yes	517	4123

Forward Selection:

$L-y

y	no	yes
no	3752	674
yes	514	4126

Calculated Metrics :

Original Model			New Model		
	Backward	**Forward**		**Backward**	**Forward**
Accuracy	0.91	0.91	**Accuracy**	0.87	0.87
Precision	0.67	0.67	**Precision**	0.86	0.86
Recall	0.43	0.43	**Recall**	0.89	0.89

Recommendation

From the two models generated in logistics regression we find clearly that the new model with balance node provides better precision and better recall (recall is twice as better than original model), even though the accuracy is lower by 4.0% which is acceptable as 87% accuracy is high enough.

The new model is based in SPSS where every time the SPSS is run, it selects new sample based on balance node and gives a new equation for model with change in precision and recall and accuracy , however the change is very small each time.

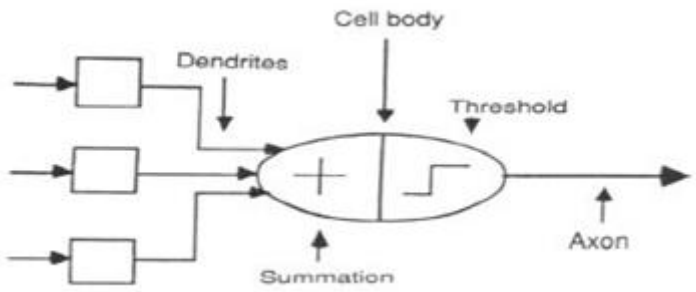
So, we recommend to use the Generated model equation for the new model given above for accurate prediction of ”yes” to term deposits. You may choose from either forward or backward selection model as both are giving same performance metrics (as shown above in Calculated Metrics) with only change in equation variables and coefficients for each selection model.

5.4 Neural Network

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, processes information. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurones.

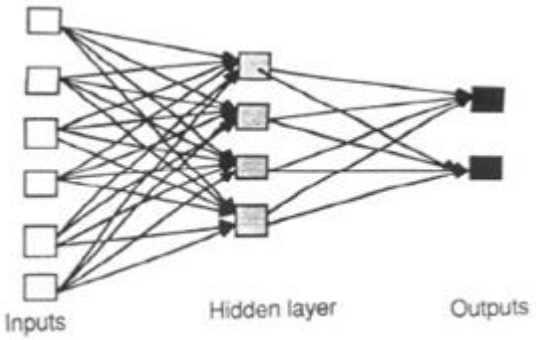
When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

The neural model in resemblance to the human neural model would be as below



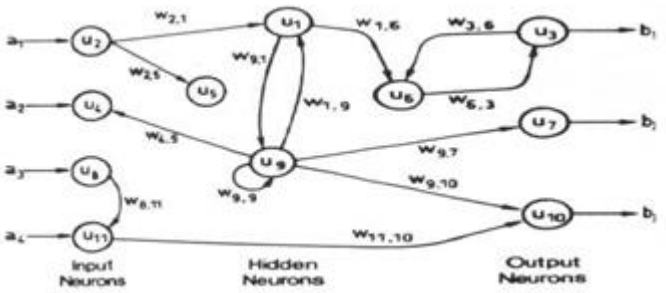
Feed-forward networks

Feed-forward ANNs allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organisation is also referred to as bottom-up or top-down.



Feedback networks

Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organisations.



Network layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units

The activity of the input units represents the raw information that is fed into the network. The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units. The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

Multilayer Perceptron

Multi-Layer Perceptron (MLP) is a popular architecture used in ANN. The MLP can be trained by a backpropagation algorithm. Typically, the MLP is organized as a set of interconnected layers of artificial neurons, input, hidden and output layers. When a neural group is provided with data through the input layer, the neurons in this first layer propagate the weighted data and randomly selected bias through the hidden layers. Once the net sum at a hidden node is determined, an output response is provided at the node using a transfer function.

In multilayer perceptrons the activation function is a sigmoidal activation function in the form of a hyperbolic tangent. The two activation functions commonly used are described as follows

$\phi(v_i) = \tanh(v_i)$ whose range is normalized from -1 to 1

$\phi(v_i) = (1 + \exp(-v_i))^{-1}$ is vertically translated to normalize from 0 to 1

The MLP network is trained with error correction learning, which means that the desired response for the system must be known. From the system response at PEi at iteration n, yi(n) and the desired response di(n) for a given input pattern, an instantaneous error ei(n) is defined as follows

ei(n)= di(n)- yi(n)

Radial Basis Function

The Radial Basis Function (RBF) is another popular architecture used in ANN. The RBF, which is multilayer and feed-forward, is often used for strict interpolation in multi-dimensional space. The RBF network comprises three layers, i.e. input, hidden and output. The input layer

is composed of input data. The hidden layer transforms the data from the input space to the hidden space using a non-linear function. The output layer, which is linear, yields the response of network.

Each node in the hidden layer is a p-multivariate Gaussian function, given as follows:

$$\rho\big(\,\|\mathbf{x}-\mathbf{c}_i\|\,\big)=\exp\left[-\beta\,\|\mathbf{x}-\mathbf{c}_i\|^2\right]$$

The Gaussian basis functions are local to the centre vector in the sense that

$$\lim_{||x||\to\infty}\rho(\|\mathbf{x}-\mathbf{c}_i\|)=0$$

The output of the network is then a scalar function of the input vector $\varphi:\mathbb{R}^n\to\mathbb{R}$, and is given by

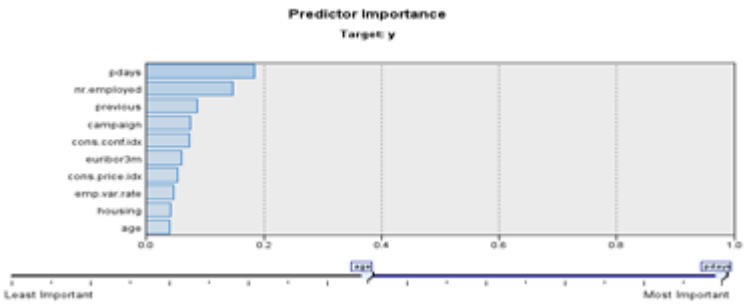
$$\varphi(\mathbf{x})=\sum_{i=1}^{N}a_i\rho(||\mathbf{x}-\mathbf{c}_i||)$$

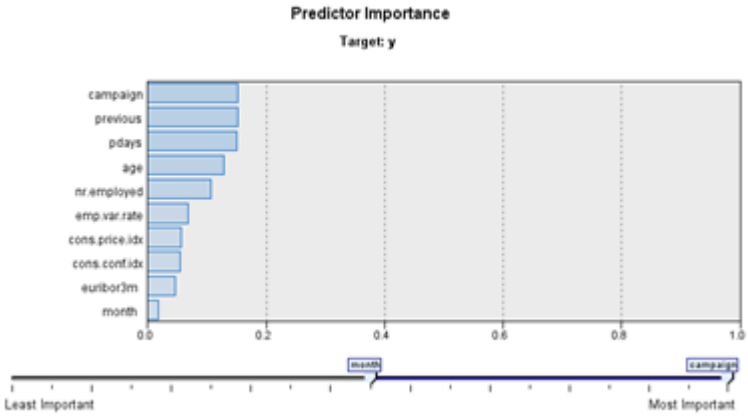
Data Processing

On examining the data using the data audit node it was observed that there has been extreme data values in the pdays column are 5 standard deviation away, hence the data was trimmed to avoid the infusion of such values to distort the prediction values. The partition created here is 70 for training and 30 for testing using the partition node. A stratified sampling node was used to sample a fraction of data of about 10000 rows from the original dataset to avoid zero values in the false positive and true positive values. This was due to the higher number of no's in the dataset which required to balance the data values to get better prediction. Duration was excluded from the analysis as it highly affects the output target variable.

Analysis

Here the analysis was based on training the data by two methods a Multilayer Perceptron (MLP) method and Radial basis function method (RBF). Additionally instead of using the default settings of the MLP and RBF for the hidden layers, customized hidden layers were added, i.e 2 for MLP and RBF and the output obtained from each model was compared by iterative testing. P days was considered as the most important variable for prediction when MLP was used while campaign was the most important variable for prediction while using the RBF method.





Result

All the models were connected to the analysis node to understand the performance of the models developed. It can be clearly seen that the model developed by the MLP method has a better accuracy compared to the other models developed in RBF.

Results for output field y

Individual Models

Comparing $N-y with y

'Partition'	1_Training		2_Testing	
Correct	25,774	89.66%	11,177	89.84%
Wrong	2,973	10.34%	1,264	10.16%
Total	28,747		12,441	

Comparing $N1-y with y

'Partition'	1_Training		2_Testing	
Correct	25,643	89.2%	11,125	89.42%
Wrong	3,104	10.8%	1,316	10.58%
Total	28,747		12,441	

Comparing $N2-y with y

'Partition'	1_Training		2_Testing	
Correct	24,276	84.45%	10,557	84.86%
Wrong	4,471	15.55%	1,884	15.14%
Total	28,747		12,441	

Comparing $N3-y with y

'Partition'	1_Training		2_Testing	
Correct	20,590	71.62%	8,846	71.1%
Wrong	8,157	28.38%	3,595	28.9%
Total	28,747		12,441	

The confusion matrices generated from the testing data for each of the models are as follows:

	Predicted True	Predicted False	
MLP (default)	342	1036	Actual True
	228	10835	Actual False

MLP (2 hidden layers)	429	949	Actual True
	367	10696	Actual False
RBF (default)	773	605	Actual True
	1279	9784	Actual False
RBF (2 hidden layers)	996	382	Actual True
	3213	7850	Actual False

Here the primary goal is to examine the sensitivity or recall of the model performance to determine the best performing model as the impact that sensitivity could have to the business model is high. We use the F4 score to determine the best performing model.

Model	Accuracy	Precision	Recall	F4
MLP (default)	0.89	0.6	0.24	0.25
MLP (2 hidden layers)	0.89	0.54	0.31	0.32
RBF (default)	0.85	0.38	0.56	0.55
RBF (2 hidden layers)	0.71	0.24	0.72	0.64

RBF has clearly outperformed MLP when seen from the values of F4 score, thus to get a better prediction for the business problem here, RBF would be the right model to choose.

6. Discussion

Based on the results from the earlier section, we see that most algorithms perform similarly on the data set provided. The confusion matrices for the best model for each of the four algorithms provides help us to determine the best of all. As mentioned earlier, since we are interested in minimizing the total number of calls made while maximizing the total number of correctly positive (true positive) predictions, the metric of interest is the F_4 Score. Table xx shows Precision, Recall and the F Score for the best model of each algorithm.

Algorithm	Precision	Recall	F_4 Score
Decision Tree - C5.0	0.55	0.49	0.49
SVM - RBF	0.19	0.67	0.58
Logistic Regression	0.86	0.89	0.84
Neural Networks	0.24	0.72	0.64

Table xx

From the results, we see that since Logistic Regression has the highest F Score, that is the best model.

7. Conclusion

According to our Analysis we can tell that the most influential variables are nr.employed, euribor3m, emp.var.rate , duration, pdays and Campaign.

Based on signs of coefficients of variables in logistic regression, “duration” has positive effect on people saying “yes”. This is because the longer the conversations on the phone, the higher interest the customer will show to the term deposit. “nr.employed”, which is the number of employees in the bank, has positive effect for turning people to subscribe the term deposit. This can be due to the fact that the more the number of employees the bank has, the more influential and prestigious this bank is. “euribor3m” is another important variable, which denotes the euribor 3 month rate. This indicator is based on the average interbank interest rates in Eurozone. It also has positive effect since the higher the interest rate the more willingly customer will spend their money on financial tools. If we go by original model in Logistic regression, we find , Campaign and Pdays go hand in hand, Campaign signifies the number of contacts made to a customer, the more the contacts made there is a high probability that could lead to successful subscription similarly if the number of days from the initial contact is more the customer would not retain the information of the subscription and thus contacting the customer on a timely basis would be necessary. Employment variation rate (emp.var.rate) has negative influence, which means the change of the employment rate will make customers less likely to subscribe a term deposit. This makes sense because the employment rate is an indicator of the macroeconomy. A stable employment rate denotes a stable economic environment in which people are more confident to make their investment. Therefore, if banks want to improve their lead generation, what they should do is to hire more people to work for them, improve the quality of conversation on the phone and run their campaigns when interest rates are high and macroeconomic environment is stable

8. Bibliography

http://people.duke.edu/~rnau/regstep.htm

http://file.scirp.org/pdf/JWARP20121000014_80441700.pdf

Wikipedia

9. Appendix

Metadata:

Column Name	Description	Type
age	Age of the client	Numeric
job	Client's occupation	Categorial: ● admin ● blue-collar ● entrepreneur ● housemaid ● management ● retired ● self-employed ● services ● student ● technician ● unemployed ● unknown
marital	Marital status	Categorial: ● divorced ● married ● single ● unknown **Note:** divorced means divorced or widowed
education	Client's education level	Categorial: ● basic.4y ● basic.6y ● basic.9y ● high.school ● illiterate

		• professional.course • university.degree • unknown
default	Indicates whether the client has credit in default	Categorial: • no • yes • unknown
housing	Indicates whether the client has a housing loan	Categorial: • no • yes • unknown
loan	Indicates whether the client as a personal loan	Categorial: • no • yes • unknown
contact	Type of contact communication	Categorial: • cellular • telephone
month	Month that last contact was made	Categorial: • jan • feb • ⋮ • dec
day_of_week	Day that last contact was made	Categorial: • mon • tue • wed • thu • fri
duration	Duration of last contact in seconds	Numeric **Note:** This attribute highly affects the output target (e.g., if duration=0 then y=no). Yet, the duration is not known before a call is performed. Also, after the end of the call, y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

campaign	Number of contacts performed during this campaign for this client (including last contact)	Numeric
pdays	Number of days since the client was last contacted in a previous campaign	Numeric **Note:** 999 means client was not previously contacted
previous	Number of contacts performed before this campaign for this client	Numeric
poutcome	Outcome of the previous marketing campaign	Categorial: ● failure ● nonexistent ● success
empvarrate	Employment variation rate (quarterly indicator) **Note:** This column was named emp.var.rate in the original data set.	Numeric
conspriceidx	Consumer price index (monthly indicator) **Note:** This column was named cons.price.idx in the original data set.	Numeric
consconfidx	Consumer confidence index (monthly indicator) **Note:** This column was named cons.conf.idx in the original data set.	Numeric
euribor3m	Euribor 3-month rate (daily indicator)	Numeric
nremployed	Number of employees (quarterly indicator) **Note:** This column was named nr.employed in the original data set.	Numeric