PREDICTING BANK MARKETING OUTCOMES:

Applying Data Mining and Machine Learning Techniques to the Problem of Binary-class Prediction

Team Bumblebee

G. Patel; R. Ramirez; J. McConnell; T. Englert

Executive Summary

Term deposits are a form of investment that requires a customer to set-aside an established sum of money in exchange for which, the bank applies a healthy interest rate, typically at the end of the term – the longer the deposit remains in the term investment, the greater the amount of interest on that amount is earned. For a banking institution, increasing the number of term deposits also increases the stability of total funds available for lending. For this data mining exercise, our team applies four data models to the problem of increasing term-deposit subscriptions through direct telemarketing campaigns. The goal of this report is to assess the performance of Logistic Regression, Decision Trees, Support Vector Machines and Neural Networks in correctly classifying candidates likely and unlikely to subscribe.

From the publically-available raw data, the team created two training data-sets, each supporting the different categorical-variable handling capacities of our predictive models: BankTrain_1, which contains categorical variables; and BankTrain_2, which applies dummy encoding to categorical variables. From these training sets, the team then constructed two diagrams. In the first diagram, we use BankTrain_2 to refine both Logistic Regression and Neural Network models. In the second, we use BankTrain_1 to build and refine both Support Vector Machine and Decision Tree models. The final evaluation assesses the comparative best-of-class model performance in terms of AUC and misclassification rankings. Of these, the top-performing models, the team identifies Random Forest as the best-performing prediction model, with an AUC of 0.802, and a misclassification rate of ~0.094.

Table of Contents

1. Introduction

For this data mining exercise, our analysis team examined a data set relating to a direct marketing campaign of a Portuguese bank, aimed at increasing the number of term-deposit subscriptions. Our goal was to apply Data Mining techniques to build a competitive model reliably predicting the likelihood that a client would subscribe a term-deposit. This decision problem is one of binary classification; our goal is to approximate the decision boundary between one class likely to subscribe, one the other, unlikely to subscribe. The target variable describes the success/failure outcome of the marketing campaign for a particular client. The input variables are grouped into four categories reflecting (1) bank client stats, (2) last client contact during the present campaign, (3) other details related to campaign stats, and (4) social and economic contexts. Details of these variable names and data types may be found in *Table 1.*

Table 1: Variables by Attribute and Data Type

Attribute/Type	Categorical	Numeric
Bank Client Data		
Age		1
Default	1	
Education	1	
Housing	1	
Job	1	
Loan	1	
Marital	1	
Last Contact, Current Campaign		
Contact	1	
Day_of_Week	1	
Duration		1
Month	1	
Other		
Campaign		1
Pdays		1
Poutcome	1	
Previous		1
Social/economic		
Cons.Conf.Idx		1
Cons.Price.Idx		1
Emp.Var.Rate		1
Euribor3m		1
Nr.employed		1

1.1 Models Overview

Of probability models best equipped to predict the decision boundary distinguishing one group classification from another, Logistic Regression, Decision Trees, Neural Networks and Support Vector Machines stand out as the most likely methods supporting the bank's decisions to targeting the most-likely subscribers (*Table 2*). Essentially a form of linear regression, Logistic Regression assumes a binomial distribution of errors rather than Gaussian, which enables a two-group classification/prediction. However, supposing the decision boundary between two classifications is not linear – this model will underperform, failing to capture nuances in the group distinction. In this case, Support Vector Machines (SVMs) emerge as promising classifiers. SVMs use one of four different kernels to transform original data into a new mathematical space that supports the description of the decision boundary: linear, polynomial, sigmoid and radial.

Neural networks model data relationships through highly interconnected artificial "neurons", that both accept and transmit information forward to the next receptor, and also send some feedback back to an earlier receptor in the model. In this way, neural networks are trained though multiple iterations to optimally assign decision weights. Both decision trees and neural networks are adept at modelling non-linear relationships as well as variable interactions. Unlike the decision tree, however, neural networks do not easily lend to interpretability. Much like how the brain intuitively leaps to conclusions, neural networks deliver results without clear explanations of how the decisions were made. Further, unlike both regression and neural networks, classification trees are well suited to

Table 2: Classification Models by Benefits

Model by Benefits
Categorical Variables
Decision Trees
Interpretability
Decision Trees
Linear SVM
Logistic Regression
Non-linear Classification
Decision Trees
Neural Networks
Non-linear SVM
Variable interactions
Decision Trees
Neural Networks
Non-linear SVM

handling non-binary categorical variables – classification trees naturally handle multi-group classifications. The subsequent analysis is constructed to examine these application of these four binary classification models to the problem of predicting bank term-deposit subscriptions.

1.2 Software Overview

The team leveraged four main software tools throughout this analysis: Microsoft Excel, JMP, SAS Enterprise Miner, and R-Studio. Unlike JMP and SAS, the benefit of using Excel for data visualization and pre-processing is its broad availability across industries, making it a cost-responsible tool requiring minimal training. Likewise, as an open-source tool, R-Studio and the Base-R language are non-cost prohibitive. However, unlike Excel, coding in R requires a steep investment of time, particularly at the beginning of the learning curve. As with Excel, there are an abundance of user-community resources available on the internet to support self-directed learning. Unlike Excel, R is flexible and mobile. With the use of packages, one may code in R across a variety of platforms, including cloud servers.

Although essentially a SAS product, JMP stands on its own as a robust statistical-analytics software. Like Excel, JMP is light-weight enough to reside contentedly on a laptop. JMP requires a licensing fee, and it is supported by an abundance of help and learning resources. Statisticians trained in other software can quickly learn to navigate the capabilities in JMP. SAS Enterprise Miner is one tool integrated within the SAS family. The cost for our team to acquire the SAS family, version 14, was minimal; however, at an enterprise-level, the licensing of SAS may be cost-prohibitive. As with JMP, the abundance of SAS documentation for Enterprise Miner supports an introductory level of skill competence. For the most recently added "high performance" nodes, SAS offers exclusive training and certification, for a price.

SAS uses SAS-language; however, packages are available that integrate R into SAS. Additionally, SAS EM exports summary statistics to .csv format, which can be manipulated in our light-weight Excel tool. Although highly robust, SAS is essentially a "heavy" tool, occupying a large amount of space, and requiring specific graphic and RAM parameters. Whereas JMP and Excel may struggle to read-in larger data-sets, SAS 14 offers a capability supporting distributed and in-memory computing – in the Age of Big Data, these capabilities are increasingly essential to supporting enterprise-level analytics. Our team found that the processing requirements for manipulating large-data in SAS almost require its installation on a stand-alone computer. It does not perform well on a virtual machine.

In addition to these software capabilities of SAS and BaseR/R-studio, our team explored the potential in Amazon Web Services machine learning tools. Specifically, we explored the application Amazon's Machine Learning (AML) tool to make useful classification predictions. Unfortunately, this tool is 100% "black box" – not only do we not know how/why the decisions of the model are being weighted, we're also not informed about the learning algorithms applied or determined to be the best. A user can upload a training and test dataset into the AML tool, and specify a target variable. The AML tool then analyzes both datasets and applies a variety of machine learning algorithm to your datasets. AML than picks what it calculates as the best prediction model for your dataset to deliver a prediction. Because this tool was not in keeping to this paper's aim of critiquing model performance, we do not include this tool's results in the discussion. However, the team remains curious about how well these user-friendly tools perform compared to best performing models, and the reader can expect a follow-up this summer.

1.3 Approach Overview

Our team guided this research using both the CRISP-DM and SEMMA methodologies. The CRISP methodology functions as a cyclical flow, which enables a continually iterative process of business-centered data analytics, whereas the linear SEMMA process moves from start to end of each analysis project in isolate. Together, the steps of our analysis may be articulated as follows: (1) Gather Business Knowledge; (2) Gain Data Understanding – *sample, explore*; (3) Prepare Data -- *modify*; (4) Construct Data Models -- *model*; (5) Evaluate Model Performance -- *assess*; and (6) Deploy Model Capability. Although this particular assessment primarily articulates the SEMMA steps – *sample, explore, modify, model, assess* – the additional layers of gathering business knowledge and deploying capabilities aligned to business requirements are held as a higher perspective guiding our analysis. Without this high-level perspective, we are merely performing analysis for analysis' sake.

The organization of the following sections mirror the SEMMA flow of our approach to this analysis. In the discovery phase, our team performs an *exploratory analysis* to gain an understanding of the data, a measure of its integrity, and an intuitive sense of the most significant variable interactions (Section 2). Also during this phase, we perform the foundational pre-processing required of the modelling phase. In the second phase, our team performs three tiers of binary-classification *model refinement*, each with specific benefits and drawbacks (Section 3). Here, we tune each of four best-performing models to their optimal performance. Finally, the third phase *evaluates the performance* of these top-performing models using both the misclassification rate and the AUC measure, and summarizes the key insights gained from these techniques (Section 4). Additionally, this final section provides a reflection both of the software and of the techniques applied during this analysis.

2. Data Discovery

In the first phase of Data Discovery, the team performed a thorough and time-intensive exploration of the data using a variety of tools. Originally in .csv format, we opened the data in Excel to perform a quick assessment of the data quality. The data was also imported into JMP, where the team replaced missing variables, structured dummy variables for categorical data, and assigned correct data types for each variable. Much of this same pre-processing can be completed using the SAS Enterprise Miner (EM) Replacement Node; however, JMP has the additional attribute of exporting data directly into SAS format (.sas7bdat), making it EM-import ready. Within Enterprise Miner, the team leveraged the Stat Explore node to gain an understanding of the variable frequency distributions. As our pre-processing advanced to the step of variable selection, we returned to JMP to perform a Principle Components Analysis. Next, we interpreted the PCA results in Excel, finally returning to SAS to test some initial models using baseline parameters. From these data discovery techniques, the team determined the following key points:

- Both *P-days* and *Euribor3m* contained many missing values;
- *P-days, Previous, Campaign, Duration* and *Nr-employed* show significant skew, indicating the presence of outlier values;
- Of these, *P-days, Previous* and *Nr-employed* show significant predictive "worth", as calculated by SAS EM;
- *Euribo3m, Day-of-the-week, Housing* and *Loan* emerge as having the least predictive worth.

2.1 StatExplore

To understand the significance of each variable's relationship to the target variable, the team leveraged the Stat Explore node within SAS EM. In order to understanding the distribution of each variable as it relates to the target outcome, the team compared summary statistics of one target grouping to the summary statistics of the other grouping, for each variable. *Figure 1* summarizes the variables differing most strongly between yes/no outcomes, captured in terms of a "skew". Skew offers a standardized metric measuring the direction and strength of a variable's shift in mean away from the median value. In addition to detecting the presence of outliers, differences between yes/no group skew may help identify potential target-group indicators.

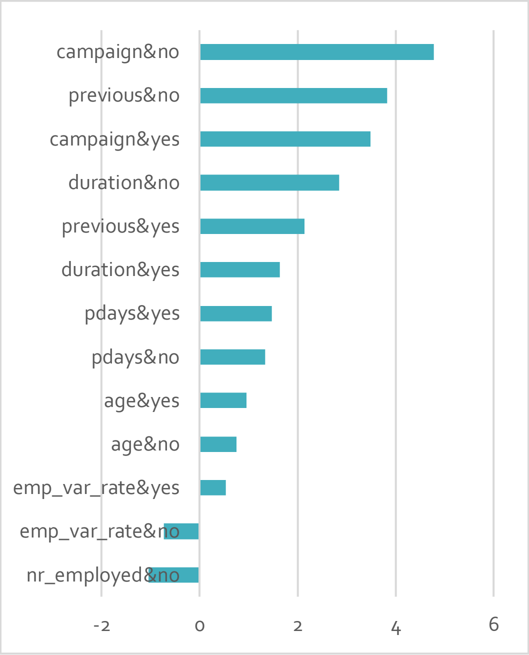

Figure 1: Skew of Variable Distributions, per Target Classification

Within this figure, we see can determine that Campaign and Previous, and Duration all display both strong skew difference from among all variables, and strong skew differences within their target groupings. Of these potential indicator variables, Duration must be discarded from future inclusion in the modelling phase. Unlike the other input variables, the duration of the marketing call is not pre-established. Rather, the duration of a marketing call is known only at the end of the call—along with its outcome. However, as recommended by the resource material accompanying this data's original set, the use of duration may serve as a baseline, for identifying other potential indicator variables.

2.2 Principle Components Analysis

Principle Components Analysis is a variable reduction technique that combines variables into components of diminishing influence, which together account for the total variation in the data. In Principle Components Analysis, we isolate a primary vector along which most the of the variance in the data is explained – the standardized linear combination along this is our first principle component. From there, we select a second vector orthogonal (perpendicular) to the first, along which the next-most variance in the data is explained. Each subsequent vector is orthogonal to the last, and explains the unexplained variation of the previous vector. The total number of components expands to the original dimensionality of the data set. Even so, it is possible to consider only a few principle components, which together explain most of the original variation.

Using PCA as a data exploration technique, we identified (1) the number of components most significant in explaining the variation of the data in the total set and (2) the interactions of the variables most significant within each principle component. In order to identify the variables having measures of the greatest importance, the team assessed the first 10 principle components within the loadings matrix. Because the fourth and fifth components almost exclusively represented single variables (rather than interacting variables), the team then reduced their examination to the first 3 components. To achieve a ranking of variables by total importance, the absolute value of each loading was totaled and assessed at the 3rd dimensions (*Figure 2*).

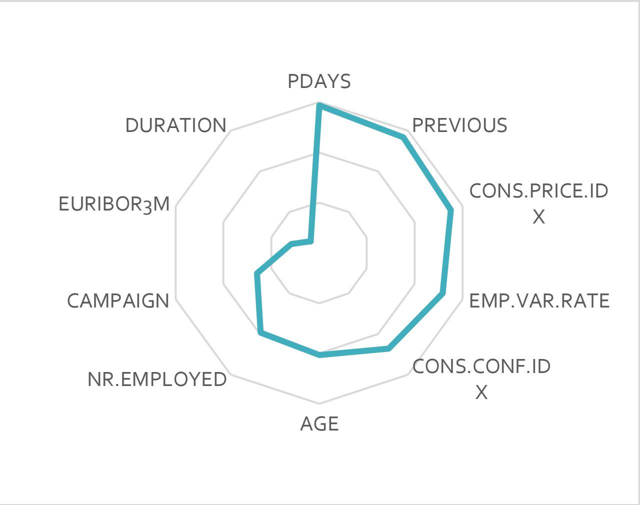

Figure 2: Total Variable Importance, Three Dimensions (descending)

2.3 Pre-processing Overview

The analysis team relied on five key pre-processing steps to optimize the quality of the performance data: (1) Decisions weights to balance the target-group observations; (2) Replacement of missing values and of +/- 3-sigma outlier values; (3) Variable Selection to include only variables with the highest correlation to the outcomes; and (4), for the less-robust of our proposed models, Imputation of missing values. Most importantly, we prepared two separate sets of raw-training data in order to accommodate the differing abilities of each of our models to process categorical variables. To best support logistic regression and neural network, the first applies dummy encoding to all categorical variables, essentially making them numeric. For decision trees and support vector machines, we present the categorical variables as-is.

3. Model Refinement

As an initial step in our data modelling phase, we needed to test our assumptions that the hypothesized binary classifiers would perform as top prediction models. Using SAS EM, the team then constructed two diagrams, one for each training set. In the first diagram, we use BankTrain_2 to refine both Logistic Regression and Neural Network models. In the second, we use BankTrain_1 to build and refine both Support Vector Machine and Decision Tree models. Using this foundation, our team constructed a variety of permutations to assess and identify the best-performing model for each model class.

To assess model performance, we examined the Test-partition ROC index value as the primary selection criteria, seconded by the Misclassification Rate for each. We also leveraged two different data-mining tools to build and score the model performances – SAS EM and Base-R/R-Studio. Further, our team explored several validation techniques to validate the model performance. After optimizing the tuning of these models, we then increased the pre-processing refinement of our data, using the Variable Selection and the Transform Variables nodes. This final step sets the stage for the final Model Evaluation phase in Section 4, where the performance of each model is then assessed in context to the other models.

3.1 Logistic Regression

Using SAS EM, the team assessed three regression node options to fit logistic regressions: Regression, Dmin Regression and HP Regression. Both Regression and HP Regression support stepwise, forward, and backward selection methods. The Dmine Regression node computes a forward stepwise least-squares regression model; in each step, an independent variable is selected that contributes maximally to the model R-square value. For each regression model, both the Impute node and the Transform Variable node proved essential to optimizing model performance.

As depicted in *Figure 3*, the best performing logistic regression model was the simple Regression using a Stepwise variable selection using Akaike Information Criterion selection measure, followed by the Dmine Regression and the HP Regression.

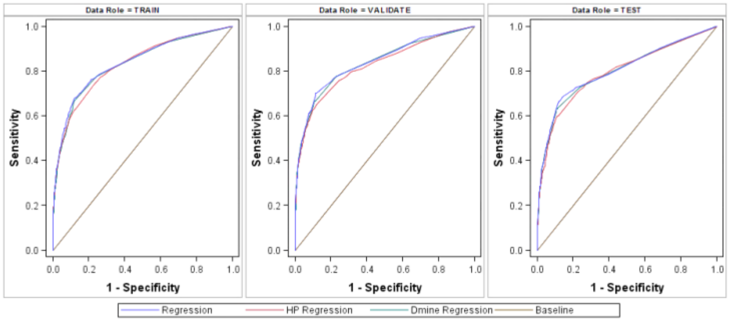

Figure 3: ROC Curves of Logistic Regression Models

3.2 Neural Network

Using SAS EM, the team assessed four neural node options: Neural Network; AutoNeural, DM Neural and HP Neural. Because the Neural Network node trains a specific neural network configuration, this node is best applied when the analyst knows a lot about the structure of the model that they want to define. Contrastingly, the AutoNeural node searches over several network configurations to find one structure best describing the relationships in a data set, and then automatically trains that network. The DMNeural node to fit an additive nonlinear model, which uses bucketed principal components as inputs to predict the target variable. The DMNeural algorithm works to overcome the common problems experienced by neural networks, including multi-collinearity. Finally, the HP Neural Node is optimized for distributed computing, making it an essential choice for big-data analytics. All of these options create multilayer neural networks, which functionally pass information from one layer to the next, mapping an input to a predicted value.

As depicted in *Figure 4*, the best performing neural modal was the HP Neural, followed by the DMNeural, then the AutoNeural.

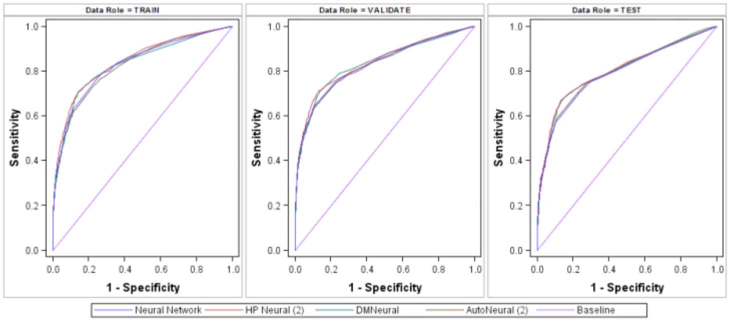

Figure 4: ROC Curves of Neural Network Models

3.3 Support Vector Machines

Using SAS EM, the team assessed countless permutations of kernel rotations, penalties and error tolerance. Of these the team found that the linear kernel performed best, and discovered an optimal performance at a penalty of 100. Next, the team increased the number of iterations from 10 to 25, to further validate the model. Because we tested so many Support Vector Machine configurations, our summary in *Figure 5* depicts only the two top-performing linear kernel models at varying validations.

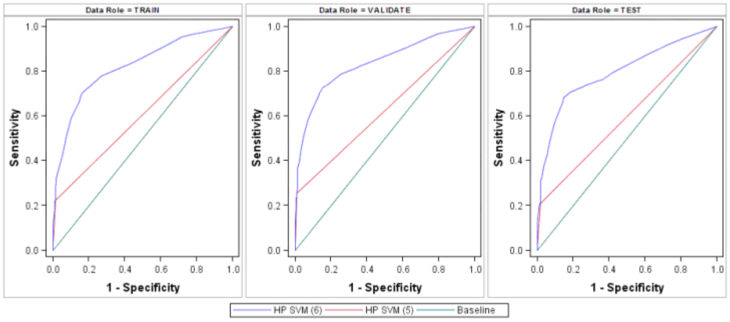

Figure 5: ROC Curves of Support Vector Machines

3.4 Decision Trees

The team constructed a decision tree using both Base-R/R-studio and SAS EM. Because the initial outputs of this model differ slightly from those considered for this final discussion, the complete R-markdown analysis is included as in Appendix A for further reference. Despite their diverging outputs, the team's method for constructing the decision tree follows the same SEMMA methodology, whether it be in R, or in SAS: the data must be (1) sampled, (2) explored, (3) modified, (4) modelled, and (5) assessed. The differences in outputs between SAS- and R-derived models primarily reflect the less-refined knowledge of the data possessed of our team in the earlier phases of this research[1].

As a step in model refinement, the team created an alternative model using SAS EM. We then explored the additional nodes of HP Tree and HP Forest. The HP Forest node applies a Random Forest ensemble method to construct and train multiple decision trees. As a method of "bootstrap aggregating", the Radom Forest model works to compute model averages, reduces variance and avoid over-fitting, rendering it a highly stable data mining technique. Although not considered in the team's initial research assumptions, it's excellent performance in the model discovery phase warrant further exploration.

As depicted in *Figure 6,* the best performing tree node is the Random Forrest, followed by the HP Tree, and the Decision Tree.

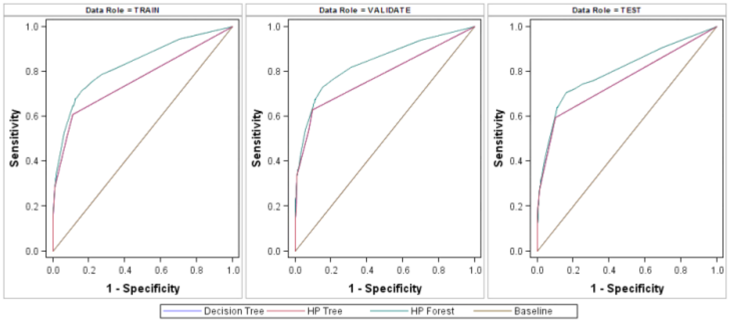

Figure 6: ROC Curves of Decision Trees

[1] We neglected to reject *Duration* as an input variable (as per Section 2.1)

4. Model Evaluation

In the final model evaluation, the team assessed the comparative performance of the best-of-class models, as a whole. For this evaluation, the team considered both the ROC Area-Under-the-Curve (AUC) index as well as the model's misclassification rate: the best model will have the highest AUC and the lowest misclassification, and will lend itself to business interpretation. Of the twelve models assessed, four emerge as having the highest value to predicting bank subscriptions. These, in ranked order, include the following: HP DM Forest, DM Neural, Regression, and HP Neural Network (*Figure 7*).

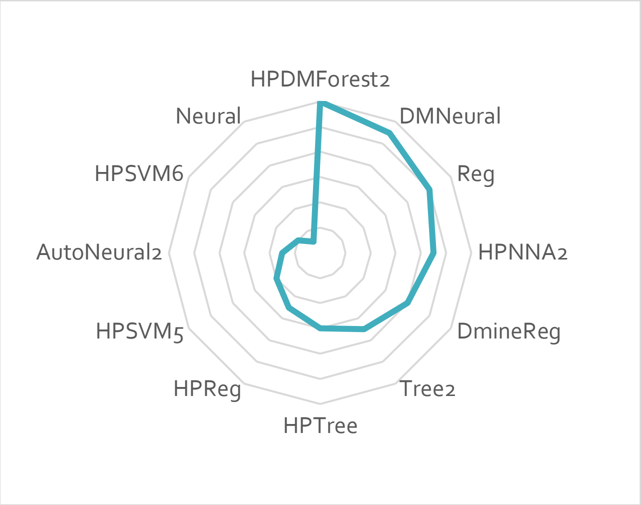

Figure 7: Model by Performance Ranking (inverse, descending)

4.1 Review of Models

All four of the following models produce AUC indexes between 80.0 and 80.2%. However, as may be noted in *Figure 8*, the models display some variation in misclassification rates, which range from between 9.4 and 9.7%. By maximizing the AUC index and minimizing the misclassification rate (MR), we arrive at the following ranking. However, because Neural Networks do not lend themselves well to business interpretation, we can further reduce our best-performing models to (1) The Random Forest, with an AUC of 80.2% and MR of ~9.39%; and (2) the Logistic Regression, with an AUC of 80.1%, and a MR of ~9.55%.

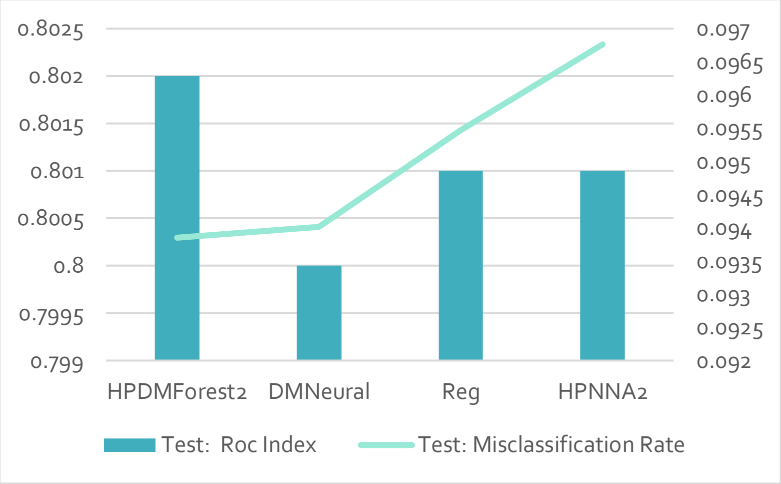

Figure 8: Top-four Models by Ranking (ascending)

Both the Random Forrest and the Logistic Regression lend themselves to business interpretability as well as significantly accurate forecasting. The Logistic Regression in particular provides insights into the relationship of variables to the desired increase in term-deposit subscriptions. However, both Regression and Random Forrest models represent supervised methods of machine learning – as the marketing approach is refined, these models will need to be manually re-tuned to reflect emerging data.

As the only unsupervised method of machine learning explored, the beauty of the Neural Network is that it moves through our entire data analysis process nearly autonomously:

- It standardizes input variables;
- It creates functions for every possible relation between each and all variables (within its hidden nodes);
- It uses these functions to determine the variables requiring decision weights;
- It iterates through data partitions to validate and tune model performance;
- And it compares results to the Y variable to come up with an Error.

Despite this innate intelligence, however, one problem inherent in using Neural Networks is that its functions are very complex, which do not lend well to interpretability. It is useless to inform the bank that $X1_4+X2X5-(X9_{12}/-X3)$ needs to be improved in order to target more subscriptions. The application of Neural Networks might best apply to problems not seeking to understand "why" in order to change a behavior, but rather seeks to reliably and autonomously predict outcomes.

4.2 Recommendations

The team recommends a hybrid model-building approach to the banking institution, which uses either the Random Forrest or Logistic Regression to identify those criteria most influencing successful subscriptions for the most-recent campaign, as well as an unsupervised binary classification model (such as Neural Networks) to spontaneously offer recommendations to managers when client status's indicate an above-threshold likelihood to subscribe. These clients might then be "groomed" in preparation for the next direct-marketing campaign, or digitally incentivized through his or her personal on-line dashboard. Thus, the combination of both supervised and unsupervised machine learning enables both continuous customer targeting as well as improved direct-marketing campaign performance.