Keio University graduate school of science and technology

Keio University

1858

CALAMVS GLADIO FORTIOR

Applied Statistical Analysis

Problem 3
Output variable prediction with logistic regression, classification tree, random forest, and neural network models

GROUP

Authors:	*Student Number:*
Vlad Stefan	82123434
Jorge Guerra	92255133
Jun Tateiwa	82218795
Axel Fridell	92255146
Jean-Baptiste Joannic	82223653

January 12, 2023

Contents

1 Introduction

The purpose of this project [2] is the creation of a model capable of solving a binary classification problem. In this case, the model is meant to determine whether the client of a bank has subscribed to a term deposit.
In the rest of the report, the dataset and the four models implemented with the R language are detailed. Furthermore, the performances of all models are analysed and compared to determine which one is most adapted to the situation.

2 Data Description

The dataset [1] is composed by the information of 10,000 people that have been registered at the bank and contains 19 explanatory variables of different types such as the age or the job of the registered client. The output variable y is binary and expresses whether the client has subscribed to a term deposit. The aim of the implemented solutions is thus to predict y by searching its relation with the input features. The dataset was manually split into a training set and a test set containing 7,500 samples and 2,500 samples respectively.

3 Logistic Regression Model

Logistic regression is a type of statistical model that is used to predict the probability that an event will occur, where the event can only take two values, e.g., ”yes” or ”no”. In logistic regression, the response variable y is therefore binary, 0 or 1, and it is predicted using n explanatory variables. The following formula represents the probability that the response variable will be equal to 1, given a set of explanatory variables represented by x_i, where i ranges from *1* to n:

$$P(y=1) = p = \frac{1}{1+\exp\{-(\beta_0+\beta_1 x_1+...+\beta_n x_n)\}} \tag{1}$$

The term in the denominator is the power of the linear combination of the regression coefficients β_0, β_1, $\dots$, β_n and the explanatory variables x_1, x_2, ..., x_n. The regression coefficients can be estimated by maximum-likelihood estimation, which is a method for finding the set of coefficients that maximizes the likelihood of correctly predicting the outcome. To make a prediction based on the output of the logistic function, we can set a threshold value T. If $P(T)$ is greater than or equal to the threshold value, then the event is predicted to occur. If $P(T)$ is less than the threshold value, then the event is predicted not to occur. The threshold value is often set to 0.5, but it can be adjusted depending on the specific needs of the model.
The logistic regression model was implemented with the `glm` function from the `stats` R package. The ROC curve obtained on the test set is shown in figure 1. Overall, the model achieves a pretty good performance with an AUC of 0.7875.

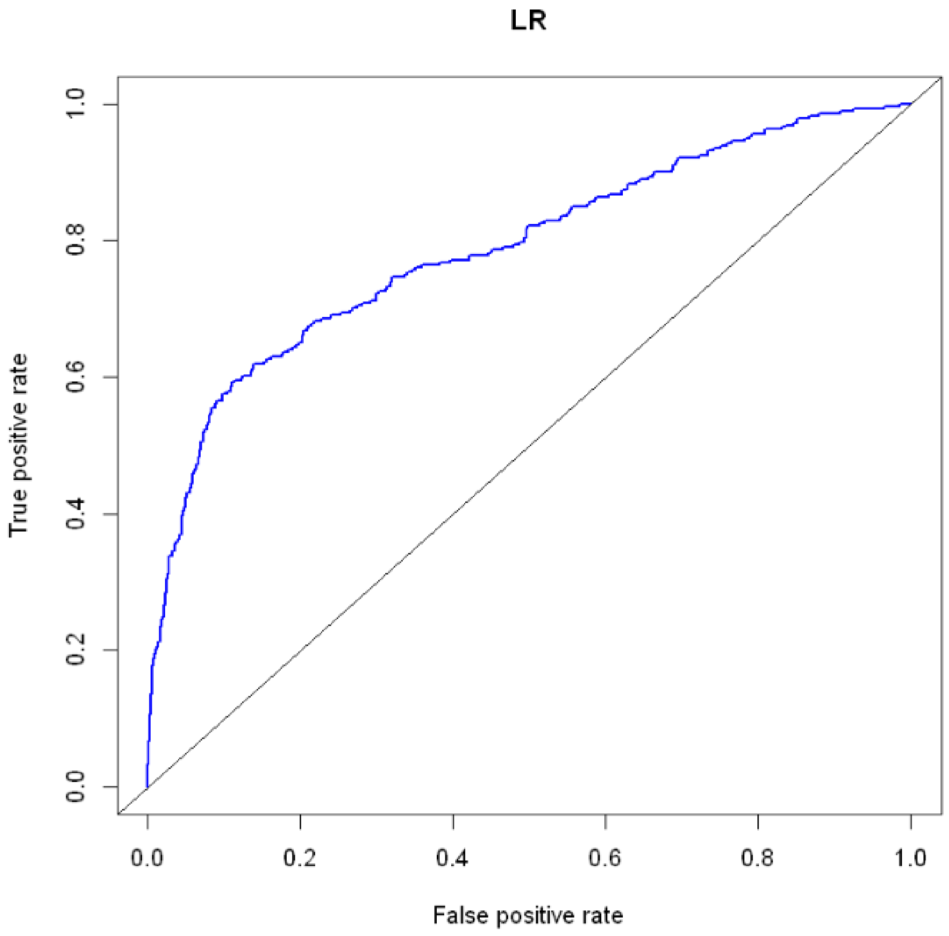

Figure 1: Prediction logistic regression model

4 Classification Tree Model

Classification trees are a type of non-linear regression analysis that can be used to predict a categorical response value. They work by partitioning the set of explanatory variables n and fitting a simple model in each partition. Classification trees are conceptually simple yet powerful tools for predicting categorical response values, such as predicting whether a patient has a certain disease (disease/non-disease). A classification tree model is therefore a tree-like model which consists of a series of decision nodes, branches, and leaf nodes. They are particularly useful because they can be easily visualized and interpreted. However, they can also be prone to overfitting if the tree is allowed to grow too deep or has too many branches. It is thus often necessary to prune the tree by removing unnecessary branches or limiting its depth.
The ROC curve of the classification tree is shown in figure 2. The corresponding AUC is 0.6815. Overall, the decision tree does not perform as well as the logistic regression.

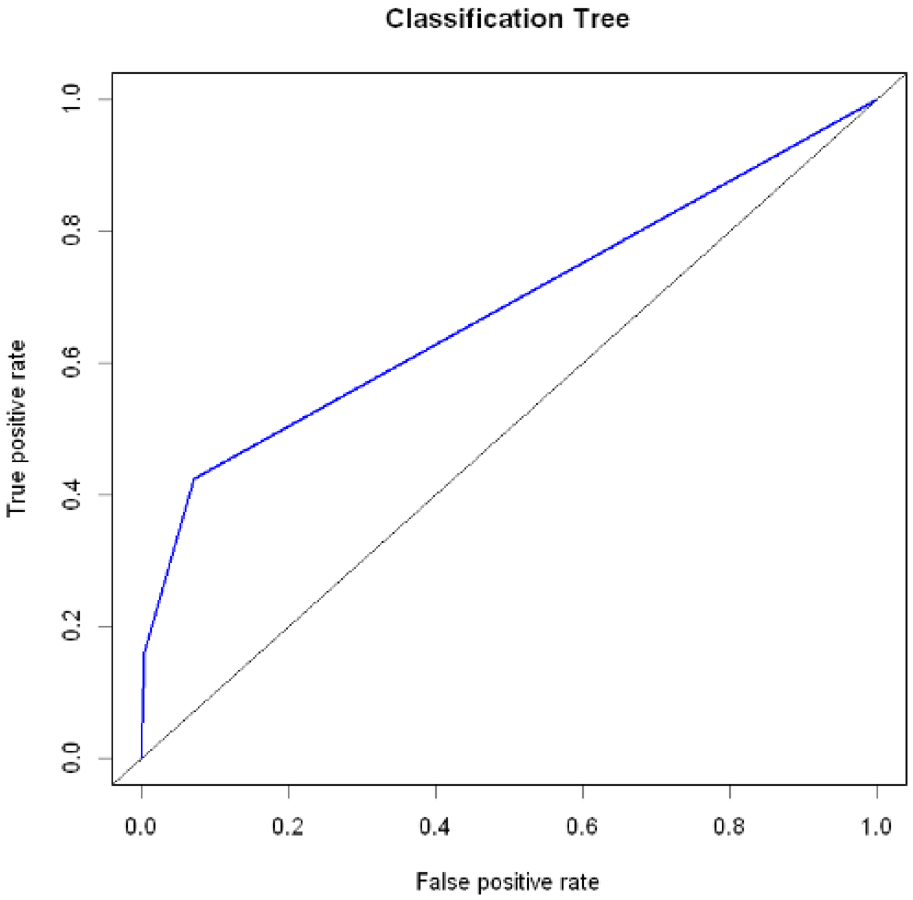

Figure 2: Prediction classification tree model

5 Random Forest Model

A random forest is a model that is made up of multiple decision trees, where each decision tree is trained on a random subset of the data and makes predictions based on certain features of the data. To make a prediction using a random forest, the data passes through each decision tree and the predictions made by each tree are then combined. The final prediction can, for instance, be the mean of the predictions made by each tree. The idea behind random forests is to create a diverse set of decision trees that can make predictions with a higher degree of accuracy and generalization than a single decision tree. This is achieved by training each decision tree on a random subset of the data and considering a random subset of features at each split in the tree. While random forests are resistant to overfitting, they can be computationally expensive to train, especially when dealing with large datasets.

The performance of the random forest on the test set is illustrated by the ROC curve in figure 3. The AUC achieved by the random forest is 0.7752. The random forest seems to be about as effective as the logistic regression. This is unusual, since random forests are more expressive than logistic regression and often outperform it. Maybe the dataset was too small and the number of features too large for the random forest to fully capture the relation between the response and the explanatory variables. It is also possible that the hyperparameters of the model were poorly chosen for the problem.

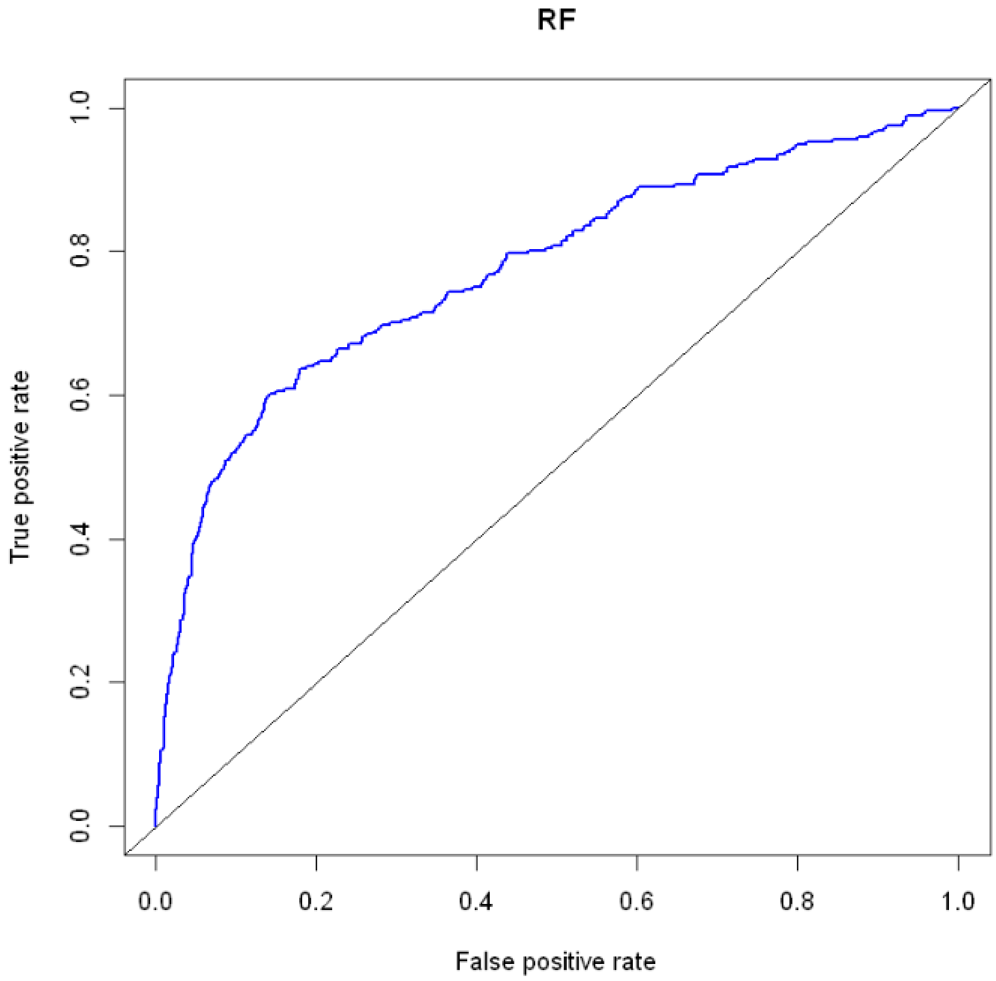

Figure 3: Prediction random forest classification model

6 Neural Network Model

Finally, a neural network is a type of machine learning algorithm which uses layers of neurons to process data and give a result. Neurons can be seen as a simple linear classifier that weigh its input data to produce a sum as its output. By passing through multiple layers of neurons, the input data will produce a prediction. The weights of all neurons are optimized by reducing a loss function. In the case of supervised machine learning, the loss function can be the sum of squared errors on the input data. Like in other models, the data is split between the train data and the test data. The algorithm will optimize a neural network on the train data and evaluate its performances on the test data. Neural networks offer very good performances for a wide range of problems because they can transform data and features to solve non-linear problems. However, they tend to perform worse on simple problems, since an error in the choices of hyperparameters, such as the number of hidden layers, can cause overfitting. Moreover, the computation time can be very long even if the libraries used are optimized.

The neural network was created with the `neuralnet` package. Before being given to the network, the input data was pre-processed. First, categorical variables were converted to numerical or boolean. Then, it was also important to normalize the data to ensure that each feature has as much importance as the others. As such, the data was normalized so that each feature has a mean of 0 and a standard deviation of 1 along all the input people. This action is not mandatory since the neural network can also normalize the data through a hidden layer over the iterations, but it would be slower, and would need more layers. After experimenting with multiple configurations of hyperparameters, mostly by changing the number of layers and neurons, the performance of the model was quite disappointing. The best result was obtained with one hidden layer of five neurons. The performance of that model is shown in figure 4, with a resulting AUC of 0.7323. It might be possible to improve the effectiveness of the model by applying feature selection before feeding input data to the model. It is also possible that the current choice of hyperparameters is not optimal, and a different neural network configuration would yield better results.

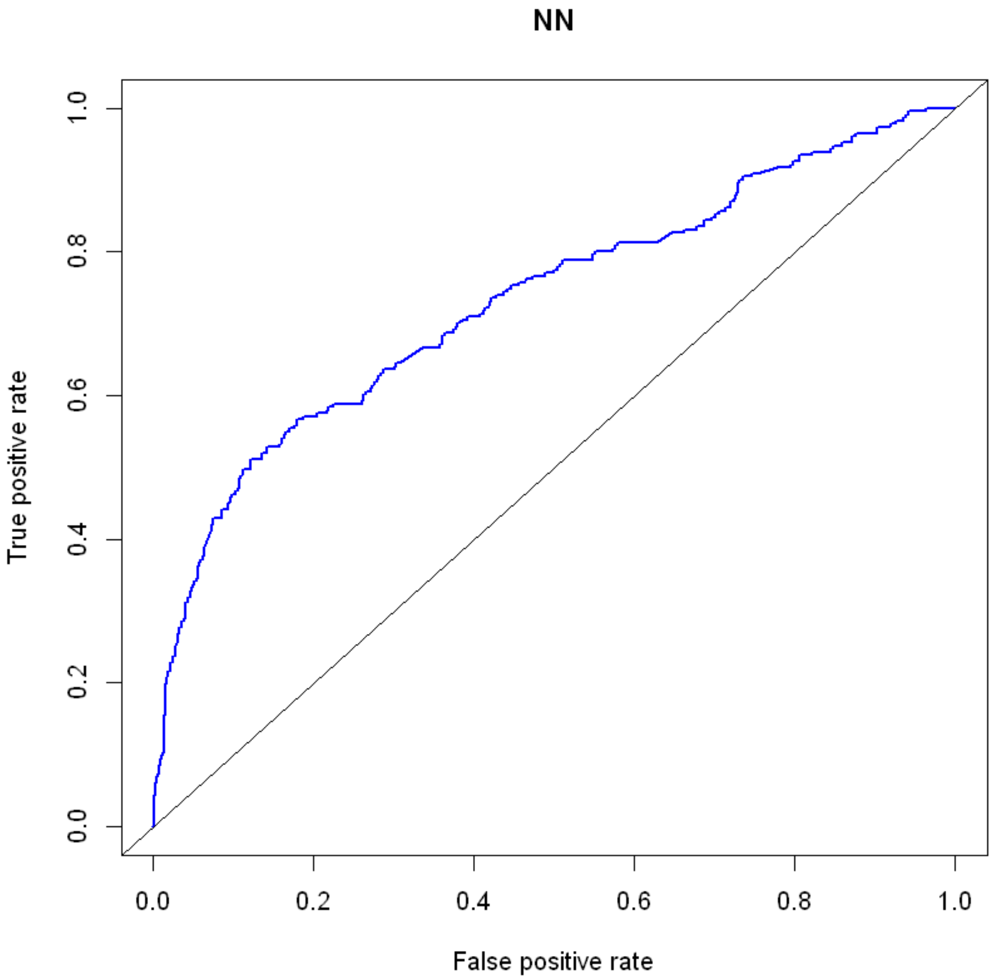

Figure 4: Prediction neural model (with hidden_nodes = 5)

7 Best Model

7.1 Input Variable Importance

The logistic regression has achieved the best AUC as shown in section 3. To see which of the input variables is most significant for the model, it is possible to look at the output of the glm function. Indeed, glm provides the p-values of the likelihood-ratio test performed on the coefficients of the input variables. The results are shown in list 1.

```
Coefficients: (1 not defined because of singularities)
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)               -1.528e+02 7.968e+01 -1.917 0.055233 .
age                       -6.505e-03 4.932e-03 -1.319 0.187193
jobblue-collar             5.451e-02 1.651e-01  0.330 0.741259
jobentrepreneur            2.549e-01 2.384e-01  1.069 0.284883
jobhousemaid               3.099e-01 2.753e-01  1.126 0.260274
```

```
jobmanagement                  1.883e-01 1.779e-01  1.058 0.289858
jobretired                     5.812e-01 2.189e-01  2.655 0.007941 **
jobself-employed              -3.015e-01 2.787e-01 -1.082 0.279333
jobservices                   -1.910e-02 1.774e-01 -0.108 0.914231
jobstudent                     6.572e-02 2.519e-01  0.261 0.794184
jobtechnician                  2.343e-01 1.478e-01  1.586 0.112804
jobunemployed                  4.820e-02 2.488e-01  0.194 0.846383
jobunknown                     7.819e-02 4.351e-01  0.180 0.857397
maritalmarried                 1.889e-01 1.456e-01  1.298 0.194385
maritalsingle                  9.063e-02 1.659e-01  0.546 0.584938
maritalunknown                 1.022e+00 7.905e-01  1.293 0.195992
educationbasic.6y              1.390e-01 2.396e-01  0.580 0.561666
educationbasic.9y             -1.087e-01 1.996e-01 -0.545 0.585914
educationhigh.school           1.525e-01 1.906e-01  0.800 0.423672
educationilliterate            1.129e+00 9.013e-01  1.252 0.210473
educationprofessional.course 1.169e-01 2.095e-01 0.558 0.576864
educationuniversity.degree 1.958e-01 1.910e-01  1.025 0.305256
educationunknown               2.144e-01 2.465e-01  0.870 0.384400
defaultunknown                -1.173e-01 1.339e-01 -0.876 0.381233
defaultyes                    -8.687e+00 1.970e+02 -0.044 0.964823
housingunknown                 3.065e-01 2.581e-01  1.187 0.235070
housingyes                    -1.331e-01 8.560e-02 -1.554 0.120104
loanunknown                           NA        NA     NA       NA
loanyes                       -9.707e-02 1.198e-01 -0.810 0.417879
contacttelephone              -5.867e-01 1.610e-01 -3.644 0.000269 ***
monthaug                      -4.684e-02 2.521e-01 -0.186 0.852566
monthdec                      -6.459e-02 4.416e-01 -0.146 0.883722
monthjul                      -6.263e-02 1.991e-01 -0.315 0.753080
monthjun                      -5.448e-01 2.664e-01 -2.045 0.040831 *
monthmar                       1.107e+00 3.038e-01  3.642 0.000270 ***
monthmay                      -7.721e-01 1.728e-01 -4.469 7.84e-06 ***
monthnov                      -5.144e-01 2.468e-01 -2.085 0.037112 *
monthoct                      -2.980e-01 3.221e-01 -0.925 0.354793
monthsep                      -3.669e-01 3.693e-01 -0.993 0.320477
day_of_weekmon                -2.617e-01 1.372e-01 -1.907 0.056508 .
day_of_weekthu                 1.804e-01 1.290e-01  1.398 0.162139
day_of_weektue                 7.061e-02 1.354e-01  0.521 0.602160
day_of_weekwed                 1.307e-01 1.343e-01  0.973 0.330327
campaign                      -2.097e-02 2.126e-02 -0.987 0.323806
pdays                         -8.797e-04 4.283e-04 -2.054 0.040003 *
previous                       6.235e-02 1.398e-01  0.446 0.655737
poutcomenonexistent            6.775e-01 2.155e-01  3.143 0.001670 **
poutcomesuccess                1.256e+00 4.255e-01  2.951 0.003166 **
emp.var.rate                  -1.079e+00 2.948e-01 -3.661 0.000251 ***
cons.price.idx                 1.441e+00 5.255e-01  2.742 0.006103 **
cons.conf.idx                  4.066e-02 1.682e-02  2.418 0.015624 *
euribor3m                      8.892e-02 2.711e-01  0.328 0.742950
nr.employed                    3.415e-03 6.471e-03  0.528 0.597720
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Listing 1: Logistic Regression coefficients

7.2 Confusion Matrix

The model showing the best performance is the logistic regression. Since the model is evaluated with the F-measure, the `performance` function from the `ROCR` package is used to compute the F-measure for multiple thresholds τ on the test set to determine the optimal τ. The highest F-measure is found for $\tau = 0.206272836493452$. The confusion matrix on the test set is shown in table 1. The corresponding precision, recall, and F-measure are shown in 2, 3, and 4 respectively.

		Predicted	
		Positive	Negative
Real	Positive	162	130
	Negative	184	2024

Table 1: Confusion matrix on the test set

$$\text{Precision} = \frac{TP}{FP + TP} = \frac{162}{184 + 162} = 0.4682 \tag{2}$$

$$\text{Recall} = \frac{TP}{FN + TP} = \frac{162}{130 + 162} = 0.5548 \tag{3}$$

$$\text{F-measure} = 2\frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} = 2\frac{0.4682 \times 0.5548}{0.4682 + 0.5548} = 0.5078 \tag{4}$$

8 Conclusions

The purpose of this project was predicting if clients of a bank had subscribed to a term deposit by using several informative variables. To solve this problem, four different models were created: logistic regression, classification tree, random forest, and neural network. The performance of the models were measured and compared using the ROC curve and AUC, showing that the most effective model is the logistic regression.

References

[1] Suzuki Hideo. *Data set Handout.* 2022.

[2] Suzuki Hideo. *Homework 3 instructions Handout.* 2022.