Lab02_A2

2022-12-02

Assignment 2. Decision trees and logistic regression for bank marketing

Provided description of data: The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

Objective: The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y). The class "yes" will thus be considered the positive class.

2.1 Import data and divide into training and test set

Upon reading the data to R, it was noted that the variable "balance" that is included in the dataset is not included in the provided description of the variables. Also, the variable "day" does not correspond to the variable "day_of_week" in the variable list, the variable "day" in the data appears to refer to the day of the month. Since there is no information on year, this variable cannot be converted to information regarding day of the week. It was further noted that some of the variable categories and values in the data differ from those in the description, e.g. in the description for the variable "pdays" it is noted that the value 999 means client was not previously contacted, however the maximum value of the variable pdays in the data is 871, whereas 82% of the observations have the value -1, which likely is the coding for the client was not contacted, or that the information is unknown. Modelling this value as a numeric (low) value could have an impact on the results. The variable "duration" was removed according to the instructions.

There are six numeric input variables: age, balance, day, campaign (number of contacts performed during this campaign), pdays (number of days that passed by after the client was last contacted from a previous campaign) and previous (number of contacts performed before this campaign),

and nine categorical input variables: job, marital (marital status), education, default (has credit in default), housing (has housing loan), loan (has personal loan), contact (contact communication type), month (last contact month of year), and day (day of the month).

As can be seen from table 1, there is imbalance in the class rates.

Table 1: Frequency table of target variable.

y	n	%
no	39922	88.3
yes	5289	11.7

2.2 Fit decision trees to the training data (three different models)

Three different models were fitted to the training data:

m1a - decision tree with default settings.
m1b - decision tree with smallest allowed node size equal to 7000.

m1c - decision tree with minimum deviance equal to 0.0005.

Default settings for controlling tree growth with respect to minimum nodesize and deviance is for nodesize equal to 10 and for deviance equal to 0.01. The interpretation of the minimun deviance is that the within-node deviance must be at least this parameter (mindev) times that of the root node for the node to be split.

Misclassification rates were calculated for the training and validation data sets, see table 2 for results. Confusion matrices for the validation data sets were also calculated, see table 3 for those results. Models m1a and m1b have the lowest misclassification error for the validation sets, however model m1c is somewhat better at classifying the positive class. By following the principle of selecting the model with the lowest misclassification error for the validation set with the least complexity, model m1b would be selected, as it is less complex than model m1a (see table 4 below).

Table 2: Misclassification rates (%) for the three models.

model	err_train	err_valid
m1a	10.48	10.93
m1b	10.48	10.93
m1c	9.40	11.19

Table 3: Confusion matrices for the three models for the validation dataset.

model		no	yes	%
m1a	no	11772	153	98.7
	yes	1329	309	18.9
m1b	no	11772	153	98.7
	yes	1329	309	18.9
m1c	no	11715	210	98.2
	yes	1308	330	20.1

The trees were further investigated with respect to how changing the deviance and node size affected the size of the trees. Table 4 displays the size and deviance of the trees, as well as the minimum size and minimum deviance for the internal nodes. Adding the criterion smallest allowed node size equal to 7000 decreased the size of the tree by one leaf compared to using the default settings. This can be understood by inspecting the tree structure (code in the Appendix), where there was one internal node of size below 7000, that in model m1b had to be turned into a terminal node.

Changing the criterion of the minimum deviance from the default (0.01) to 0.0005 increased the tree considerably, to 122 leaves. The reason that the model m1c tree is larger is because internal nodes with smaller deviance are allowed. It can be seen in table 4 that the minimum node size and deviance for the internal nodes are much smaller for the model m1c tree than for the other trees.

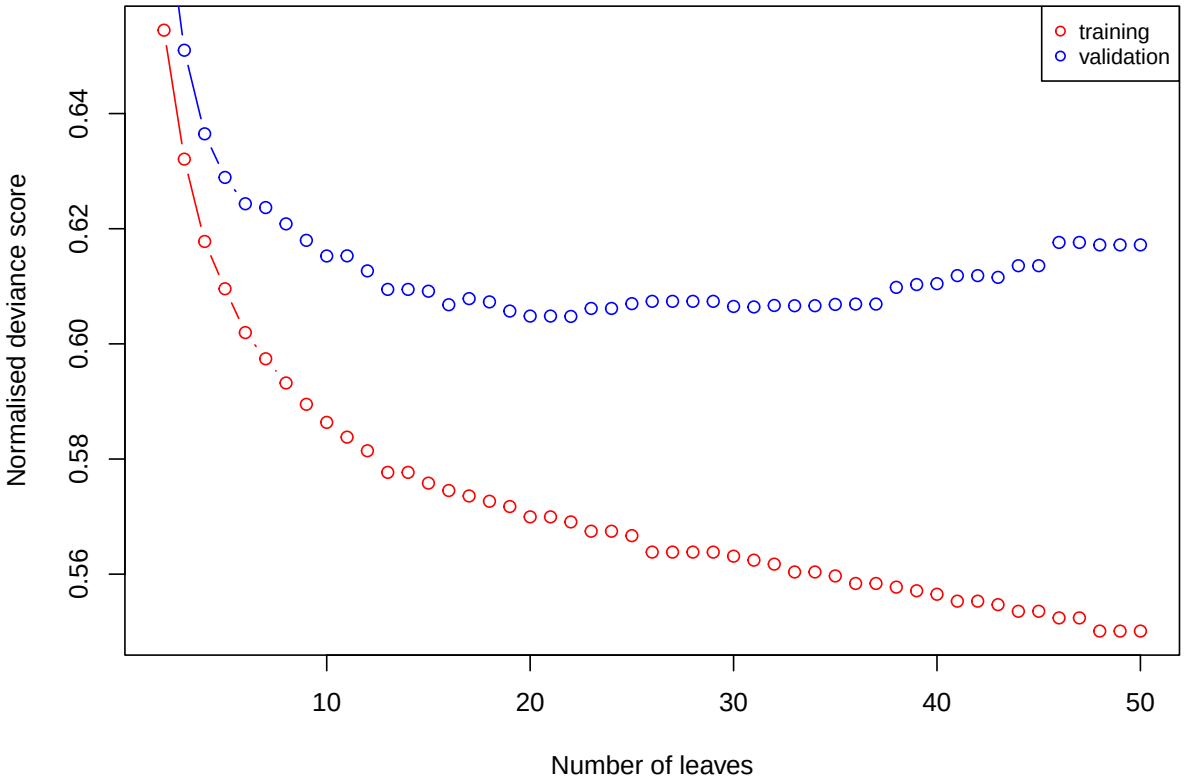
Table 4: Tree size and deviance, and minimum internal node size and deviance.

Model	Tree size	Deviance	Min int. node size	Min int. node deviance
m1a	6	10885.9	2414	2262.1
m1b	5	11023.5	11698	7913.8
m1c	122	9363.3	13	15.0

2.3. Choose the optimal tree depth in the model c in step 2 (model m1c)

The training and validation sets were used to choose the optimal tree depth in the model in step 2 (model 1c). Models were fitted with the training data for up to 50 leaves, and the validation set was used for prediction. Deviance was extracted for the training and validation sets in the models respectively, adjusted for the number of observations in each dataset. Figure 1 shows the deviances for the training and the validation data versus the number of leaves.

Figure 1. Deviances for the training and the validation data vs number of leaves



To the left in the graph, with the smaller amount of trees, are the least complex models that have a higher bias, as can be seen by that both the training and validation sets have a higher deviance. As the models become more and more complex, the deviance decreases for the training set, and for the test set also up till around 20 trees, the bias decreases, but with increasing complexity comes increased variance, which is noted by that the deviance decrease for the validation set levels off, and its deviance starts to also increase gradually from the lowest point at around 20 trees.

The optimal amount of leaves were chosen as the tree with the lowest validation error (and if there had been several trees that had an equally low validation error, the least complex tree would have been chosen).

```
#> Optimal amount of leaves
```

```
#> [1] 22
```

Inspecting the structure of the tree (code in Appendix), shows that the variables used in the tree are poutcome, month, contact, pdays, age, day, balance, housing and job. Of these, poutcome, month, pdays, job and contact seem to be the most important.

There were three decision pathways that lead to the classification "yes": if the outcome of the previous marketing campaign (poutcome) was success and the days since the previous campaign (pdays) was either less than 94.5, or if the previous days were equal to or more than 94.5 days and the job was housemaid, management, retired, self-employed, student, unemployed or unknown the classification was "yes". The third pathway that lead to the classification "yes" was if the outcome of the previous marketing campaign (poutcome) was failure, other or unknown, the month was any of January, May, July, August or November and contact was with cellular or telephone and the previous days were equal to or more than 383.5.

2.4. Estimate the confusion matrix, accuracy and F1 score for the optimal model

The confusion matrix, accuracy and F1 score were calculated for the test data by using the optimal model from step 3 (model m2c). The accuracy is the fraction or percentage of correctly classified observations, while F1 score is defined as:

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

where precision is the proportion (or percentage) of the positive identifications that was actually correct and recall is defined as the proportion (or percentage) of the actual positives that was identified correctly.

Tables 5-7 below show the different metrics.

Table 5: Error rates for model m2c (optimal) model.

err_train	err_valid	err_test
10.4	11.2	10.9

Table 6: Confusion matrix for model m2c model, test dataset.

	no	yes	%
no	11872	107	99.1
yes	1371	214	13.5

Table 7: Accuracy and F1 score for model m2c for test data.

Accuracy	F1
89.1	22.5

In terms of the total percentage of observations that gets correctly predicted, the performance could be considered quite good with 10.9% misclassification error, or 89.1% accuracy, for the test data. However, the prediction rate for the positive class is only 13.5%, which cannot be considered good. The performace of the model is better reflected by the F1 score, which has a value of 22.3.

2.5 Perform a decision tree classification with loss matrix

A decision tree classification using model m2c was performed with a loss matrix, that weighted the positive class 5:1 to the negative class. The confusion matrix and the accuracy and F1 score are shown in tables 8-9 below.

Table 8: Confusion matrix for the loss matrix classification.

	no	yes	%
no	11030	949	92.1
yes	771	814	51.4

Table 9: Accuracy and F1 score for model m2c for test data using loss matrix.

Accuracy	F1
87.3	48.6

The prediction rate has now increased for the positive class, 51.45% compared to 13.5% for the model without the loss matrix, and decreased somewhat for the negative class, 92% compared to 99.1% for the previous model. This is because the loss matrix shifts the r value so that more observations are classified as postive, The accuracy is now 87.3% and the F1 score is 48.6.

2.6 ROC curves for optimal tree and logistic regression

ROC curves were calculated for the optimal tree (model m2c) and a logistic regression model (model 3a). Precision and recall were also calculated and plotted. Confusion matrix and accuracy and F1 for the default r value (0.5) were also calculated for comparison, see tables 10 and 11. The prediction rate for the positive class was a few percentage points higher for the logistic regression model (16.5%) compared to the optimal tree (13.5%).

Table 10: Confusion matrix for logistic regression model (m3a).

	no	yes	%
no	11837	142	98.8
yes	1323	262	16.5

Table 11: Accuracy and F1 score for the logistic regression model (m3a).

Accuracy	F1
89.2	26.3

Figure 2a. ROC

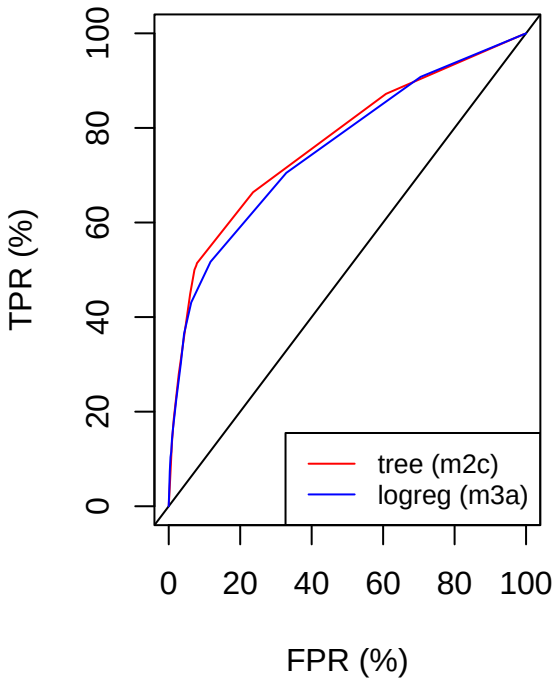
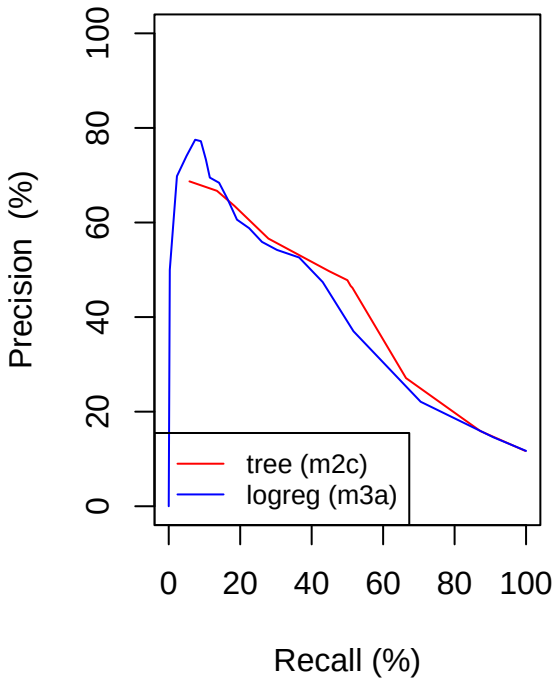


Figure 2b. Precision−Recall



The ROC and precision-recall curves are quite similar for the optimal tree and the logistic regression model, however the decision tree is somewhat better over some ranges of the performance metrics. From the ROC curve, it can be seen that to obtain around 80% true positive rate, almost 60% false positive rate has to be accepted. From the precision-recall curve, it can be seen that when an 80% rate of the true positives/recall is selected, only around 20% of the selected observations will be true positives. The precision-recall curve gives a better idea of the benefits versus the disadvanatges with selecting a certain recall/ true positive rate (or rather, selecting an r value corrsponding to a certain recall).