# Lab 2 Assignment 2

## Task 1.

Bank.csv file is read, and the variable duration is removed as instructed. The data is portioned, with the code given by lecture 2a, into training, test and validation data sets. With the split 40/30/30.

The bank file contains information about bank clients which is to be used for a marketing campaign with the goal to get as many as possible to subscribe to the banks term deposits.

## Task 2.

The R package tree was used for fitting the decision trees to the training data. Once that had been done confusion matrixes were created and used to calculate the misclassification rates for the three trees.

|  |  |  |
| --- | --- | --- |
| Decision Tree | Misclassification Rate Train | Misclassification Rate Valid |
| Default | 0.1142446 | 0.1203274 |
| Minimum Node Size 7000 | 0.1142446 | 0.1203274 |
| Deviance Minimum = 0.0005 | 0.1045676 | 0.1448242 |

Deviance tree = 0,0005 overfitted more than the other two trees to the training data, so it is not the best model. The other two however have the same misclassification rate, so judging by that metric they are equally good.

Based on the misclassification rates we acquired it seems that the tree with higher deviance fit better to the training data but worse to the validation data set. While decision trees with the highest allowed node size at 7000 didn’t make any difference compared to the default tree.

## Task 3.

The dependence of deviances for training and validation data on the number of leaves. 


*Dependence of deviances for the training and validation data on the number of leaves. Blue is validation and red is training.*

First a little background, y-axis, trainscore shows deviance, which shows amounts or errors, high deviance equals high error rate. X-axis shows number of leaves on the decision tree which corresponds to how complex the decision tree model is. Because the complexity of a decision tree is determined by how deep it is. Depth is how many nodes there are from the root (start node) to leaf (end node). So, more leaves mean a longer path in a binary tree, where each node only have two paths.

In the graph we can see that the model learns the training data better and better so that the error goes down as complexity of the model increases. But the error in the validation set goes up, indicating increased overfitting and higher variance.

We can see high bias in the graph in the beginning, most likely because it is not a complex model at that point and we can see high error.

Bias-variance tradeoff means, as the name suggests, that as bias shrinks the variance in the model will increase and vice versa. Short description is that bias can be described as accuracy, how close we are to the actual values. While variance can be described as precision, how close each data point is to other data points. The goal with bias-variance tradeoff is to select a model where the total error is as small as possible. The model is on target and not to spread out.

This can be done by selecting a model that has a good enough amount of complexity, low complexity models give high bias, while high complexity models can overfit to the training data, meaning high variance.

The optimal amount of leaves seems to be from the graph 20 based on that for that amount of leaves the deviance, the errors are the lowest for the validation data.

Running summary on the tree with 50 leaves we get to know that the variables actually used in tree construction are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| pdays | age | balance | campaign | day |

Diagram, schematic

Description automatically generated

*Tree structure for task 3.*

Based on the structure as seen in the image above, there are some redundant branches, pdays<83.5 after pdays<8.5, same with a few other pdays. The model might be a bit to complex.

## Task 4.

Optimal model was selected to the pruned model from the previous task. It was named bestTree.

The estimated confusion matrix for bestTree. Row no yes is prediction, column no yes is actual value.

predictionTest

no yes

no 11764 215

yes 1354 231

|  |  |
| --- | --- |
| Accuracy | 88.43262% |
| F1-score | 22.74742% |

Going by accuracy the model has good predictive power, while going by F1-score the model has poor. Worth mentioning is that there are roughly 16000 occurrences of class “yes” while there are 2000 of class “no” in the training data set, so the data set is imbalanced between its classes.

F1-score takes into account imbalances between classes while accuracy does not so the F1-score is the better metric for our model, so the model has poor predictive power.

## Task 5.

|  |  |
| --- | --- |
| Accuracy | 85.21085% |
| F1-score | 38.04818% |

Accuracy got slightly worse, it decreased with 3 percentage units, while F1-score increased by 16 percentage units. So, the models prediction power has gotten better.

What we did was that we weighted our probability table in a direction in order to make up for the existing imbalance between the two classes. So F1-score improved because we balanced our prediction by applying weights.

## Task 6.

Chart, line chart

Description automatically generated

*ROC curve. Red is the logistic regression model and blue is the decision tree.*

True positive rate and false positive rate is calculated using a confusion matrix with the following formulas:

TPR = TP/P True Positive Rate = True Positive / Positives

FPR = FP/N False Positive Rate = False Positive / Negatives

The rates are used to evaluate a model, high TPR is good while low FPR is good, given the same FPR a higher TPR shows the better model.

These rates can be plotted in an receiver operating characteristics curve, ROC curve, which shows the performance for a classifier. Where the bigger the area under the graph the better.

In our ROC curve we can see that the logistic regression model is the better predictor since it has more area under the graph.

There exists an alternative to the ROC curve, the Precision Recall Curve, which is more informative for imbalanced problems, such as ours. While the Receiver Operating Characteristics, ROC curve is better for balanced problems.