Assignment 1 - APML2022

Group 9
Christian Acosta | Josh Bleijenberg | Mischa van Ek | Sjoerd Vink | Robin Wiersma |

Abstract

This paper studies different anomaly detection algorithms which are used to classify outliers in the given data set. An attempt is made to improve the default parameter setting based on the F1-score. The data set which is selected for this study consists of attributes like age, job and personal loan. The main purpose of this paper is to detect the label of the rows. The performance results showed that Decision Tree was the best performing model for this specific data set. It provided the highest scores for accuracy, recall and F1.

1 Introduction

A Portuguese banking institution wanted to decide whether or not a client had access to a bank term deposit. This was done by contacting all clients multiple times, which is very time consuming. The goal of this assignment was to help the banking institution make their decisions in a far less time consuming way, so they can predict faster whether or not a client will subscribe a term deposit. In order to achieve this, several anomaly detection algorithms are used to predict the label. The algorithms used are: Decision Tree, Random Forest, Isolation Forest, Local Outlier Factor and Feature Selection.

2 Data

The provided dataset is related to a marketing campaign of an Portuguese banking institution. The dataset contained 20 input variables, 1 target variable, 41188 instances, and was provided with normalization already applied.

The input variables consist of bank client data (the age, job, marital status, education level, age, credit in default, housing situation and personal loan of the client), contact with the client (communication type, last month and weekday the client has been in contact and the duration the last time there has been contact), other attributes (number of contacts performed during the campaign, number of passed days since last contact, number of contacts performed before the campaign and outcome of the previous campaign) and social economic context attributes (employment variation rate, consumer price index, consumer confidence index, Euribor 3 month rate and number of employees). The target variable is whether or not the client subscribed a term deposit.

Figure 1 describes the data distribution of the client data. Visualized in four bar charts, the first one describing the clients job, the second one their marital status, the third one their level of education and the last one whether or not the client has a housing loan.

3 Methods

Several tree based algorithms have been implemented for classification and outlier detection on the given data set. All tree based algorithms are derived from the Scikit-learn library (version 1.0.1). For evaluation purposes, the data is split into training and testing data with a test-size of 0.25. The model is trained with the training data and the evaluation metrics are calculated with the test data.

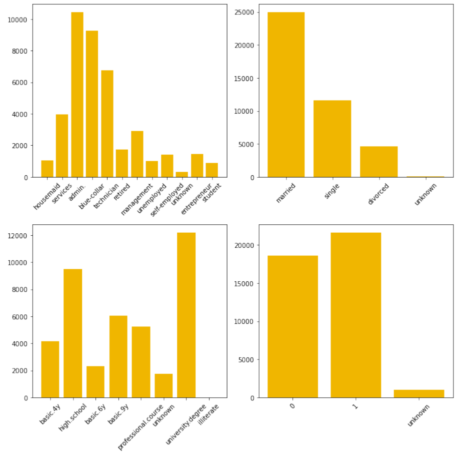


Figure 1: Data distribution

3.1 Preprocessing steps

In order to achieve maximum performance of the algorithms on the data set, several preprocessing techniques are applied. The preprocessing was already done by the data set provider and are listed below.

- Normalization: the data set is normalized in order to bring all the values between 0 and 1.
- One-hot-encoding: the categories in the data set are encoded using a one-hot-encoding technique. This results in one column per category consisting of 0's and 1's.

3.2 Algorithms

3.2.1 Decision tree

For the classification of the class label in the data set, a decision tree has been drawn up. The selection attribute 'Gini' is used to calculate the impurity of the leaves. Based on this, splits can be made to reduce the overall impurity. Care should be taken for overfitting as this is very common when constructing decision trees, especially with default parameters.

The construction of a decision tree with default parameters takes very long, because the tree isn't limited in the depth of the tree. Because of this, the decision tree grows very large and over fits the training data. It is impossible to clearly visualize the tree. The most important features where the decision tree splits the data are displayed in Figure 2.

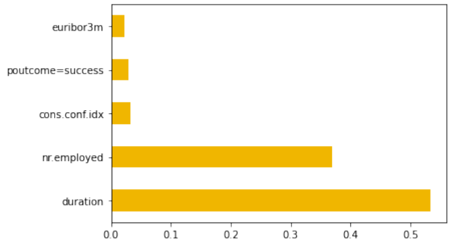


Figure 2: Feature importance of default Decision tree

3.2.2 Random forest

For the classification of the class label in the data set, a random forest has been drawn up. A random forest generally performs better than a decision tree because it is an ensemble learning method. This means that a random forest consists of multiple decision trees. For classification, the idea of most votes count is applied in order to classify a certain data point. In regression the average output of the decision trees in the forest is used as a return value.

3.2.3 Isolation forest

An isolation forest is a very efficient algorithm specifically designed for outlier detection. Intuitively, outliers are easier to separate (isolate) from the rest of the data compared to regular data points. In order to do this, the algorithm generates partitions in the data based on random splits of an attribute.

3.2.4 Local outlier factor

The local outlier factor (LOF) is an unsupervised outlier detection algorithm which computes the local density deviation of a given data point in comparison with its neighbors. It is called local outlier detection because the outlier score depends on how isolated the object is with surrounding data points. Locality is given by k-nearest neighbors. It should be noted that when the number of neighbours increases, the bias increases and the variance decreases.

3.2.5 Feature selection

The data set consists of a lot of dimensions. It should be questioned whether all these dimensions contribute to the prediction. Feature selection is used to select the best amount of features so that the F1-score increases. Less features can be good for the algorithm because some features can be seen as noise.

4 Evaluation and Discussion

In this chapter the experimental setup is described which is used to construct the optimal tree. Also, the results are described and discussed. The F1-score is used to decide which parameter setting offers the best performance on the given data set. In chapter 5 it is explained why the F1 score gives a better result.

4.1 Decision tree

The first algorithm that was used, was a decision tree that classified the bank marketing data set. When looking at the performance of the model, the accuracy of the model on the training data is around 0.89 with a precision of 0.52 and a recall of 0.53. The confusion matrix is displayed in Figure 3.

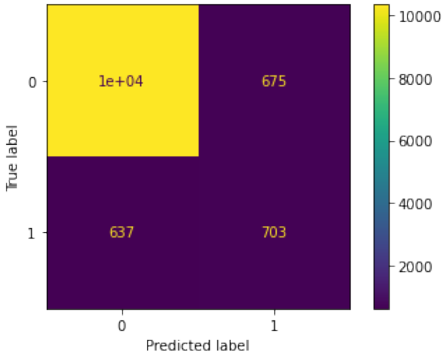


Figure 3: Confusion matrix of default Decision tree

Whenever the maximum depth of the decision tree is minimized to 4, the accuracy increases to around 0.91. For as the recall and precision stays to around 0.53 and 0.52 respectively. The resulting tree is visualized in Figure 4. The resulting confusion matrix is displayed in Figure 5

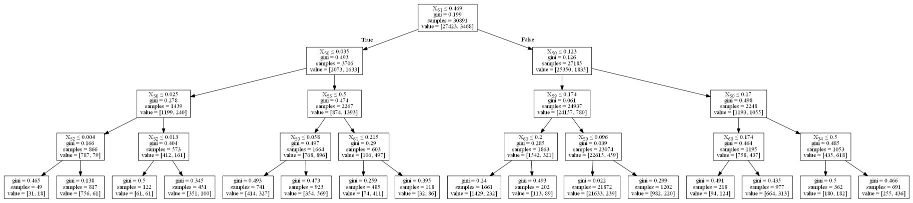


Figure 4: Decision tree with max-depth = 4

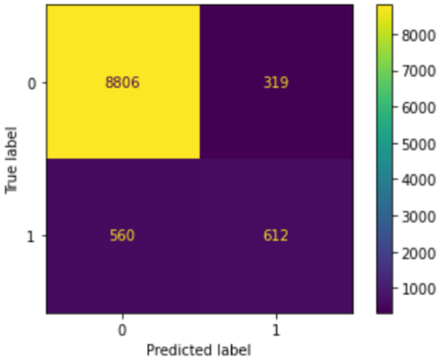
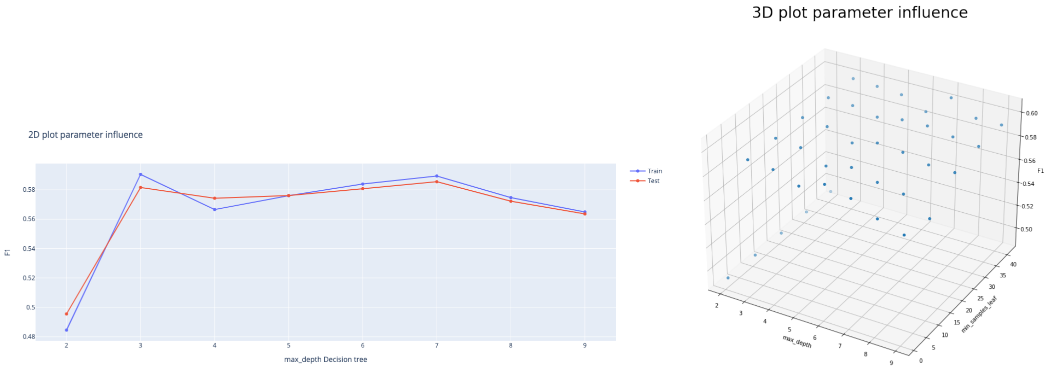


Figure 5: Confusion matrix with max-depth = 4

To assess how the maximum depth in a decision tree could affect training and testing data, a 2d plot was created. The result displayed in figure 6a showed that max depth can increase and decrease the accuracy of training and testing data.



(a) The influence of the tree depth on the F1-score

(b) 3D parameter plot

In order to construct an optimal decision tree, the maximum depth and minimum sample leaves are tuned based on the F1. The search space for these parameters is as follows:

- Maximum depth: The default value for this parameter is ’None’, so the tree can grow very large. In order to contain this and prevent over fitting, the search space is set to a range from 1 to 10 with a step size of 1
- Minimum sample leaves: The minimum number of samples required to be in a leaf node is by default 1. The data set is large so this number needs to be higher in order to improve testing F1-score. The search space is set to a range from 1 to 50 with a step size of 10.

A 10-fold cross validation was used for this to generalize the result. The optimal decision tree has the following parameters:

- minimal samples leaf = 41

- maximum depth = 7

In an attempt to understand the results, a 3d graph was made to visualize the accuracy with adjusted parameters: maximum depth and minimal samples leaf. Interestingly, the Figure 6b shows maximum depth significantly changes accuracy when tuning the parameter. This differs from minimal sample leaves where adjusting the minimal samples leaf shows to barely have any influence on accuracy.

4.2 Random forest

The random forest with default parameter settings result in a forest with an F1-score of around 0.54. The recall is notably lower than the precision, which is 0.46 and 0.68 respectively. The corresponding confusion matrix is showed in Figure 7.

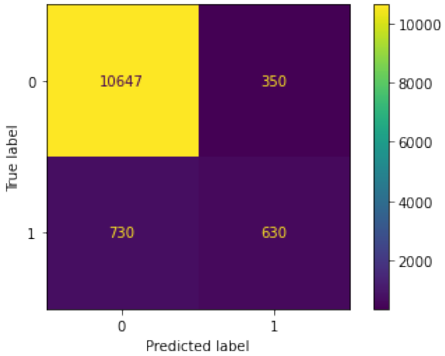


Figure 7: Confusion matrix of random forest with default parameters

In order to construct an optimal random forest, the number of estimators and the maximum features are tuned based on the F1-score. The search space for these parameters is as follows:

- Number of estimators: The default value for this parameter is 100 and is used as a reference for the search space. The range is set from 50 to 150 with a step size of 20
- Maximum features: This is the number of features that is considered in a split. There is no default value for this, but considering the number of columns in the data set the range is set from 1 to 20 with a step size of 5

A 10-fold cross validation was used for this to generalize the result. The model that performed best had the following parameters:

- n-estimators = 130
- max-features = 16

The model with n-estimators = 130 and max-features = 16 had a F1-score of around 0.57. This is a higher score than the model with the default parameters. However, the recall increases and precision decreases relative to the default model. This results in the confusion matrix displayed in Figure 8.

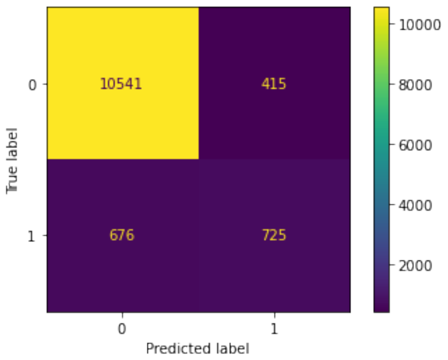


Figure 8: Confusion matrix of random forest with optimal parameters

4.3 Isolation forest

In order to construct an optimal isolation forest, the maximum samples and number of estimators are tuned based on the F1-score. The search space for these parameters is as follows based on the defaults values:

- Maximum samples: The default value for this parameter is 256, and is used as a reference for the search space. The range is set from 150 to 300 with a step size of 50.
- Number of estimators: The default value for this parameter is 100 and is used as a reference for the search space. The range is set from 50 to 150 with a step size of 20
- Contamination: This parameter is set to 0.12, based on the fraction of outliers in the original data set

A 10-fold cross validation was used for this to generalize the result. The model with max-samples=150 and n-estimators=130 performed best with an F1-score of around 0.31. Figure 9 displays the outliers in the data set.

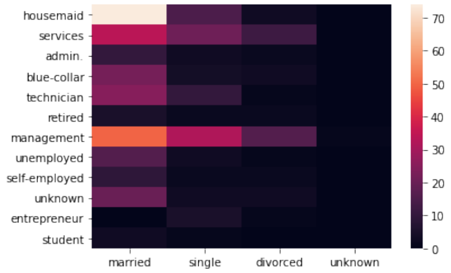


Figure 9: Outlier heatmap with optimal isolation forest: the color stands for the number of outliers in a given category

4.4 Local outlier factor

In order to construct an local outlier factor, the maximum leafs and number of neighbors are tuned based on the F1 score. The search space for these parameters is as follows based on the defaults values:

- Maximum leafs: The default value for this parameter is 30, and is used as a reference for the search space. The range is set from 1 to 100 with a step size of 10.
- Number of neighbors: The default value for this parameter is 20 and is used as a reference for the search space. The range is set from 1 to 100 with a step size of 10.

A 10-fold cross validation was used for this to generalize the result. In the initial trail the search spaces where set smaller based on the default parameter. This only resulted in a very low F1 score. The model with max-leafs=20 and n-neighbors=10 performed best here with an F1-score of around 0.037. When increasing the search space, the optimal parameters tend to be significantly lower (both parameters become 1). When decreasing the number of neighbours the bias becomes smaller, which is the goal. As a side effect, the variance increases however because less neighbours are taken into consideration. The optimal parameters based on F1 is in this case n-neighbors=1 and leaf-size=1 resulting in a F1-score of around 0.14. Figure 10 displays the outliers in the data set.

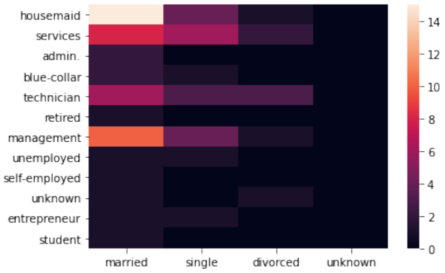


Figure 10: Outlier heatmap of local outlier factor: the color stands for the number of outliers in a given category

4.5 Feature selection

In order to improve the F1 scores of the different models. Feature selection is used. In figure 11 we can see when using 18-20 features, the F1 score is best.

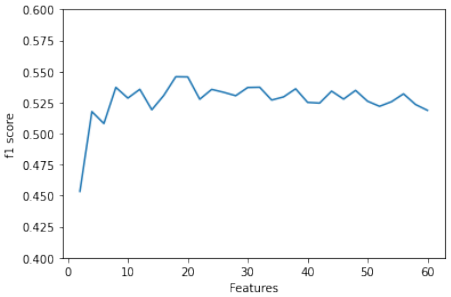


Figure 11: Feature selection

5 Overview and comparison of algorithm performance

In this section we will provide an overview of the performance of each of the three algorithms based on multiple performance measures. The performance measures we will use are accuracy, precision, recall, and F1 score. In this section we will also discuss and reflect on the performance of each model, and finally discuss which algorithm is optimal for the given data.

In the following section we present Table 1, which provides an overview of the performance scores of each model using held-out test data.

Model	Test Accuracy	Test Precision	Test Recall	Test F1 Score
Decision Tree	0.91	0.59	0.57	0.58
Random Forest	0.91	0.52	0.65	0.58
Isolation Forest	0.85	0.33	0.35	0.34

Table 1: Overview of test model performance using various measures

To select the optimal model, we used the F1 score to judge the models. We decided to use the F1 score for a number of reasons. The main reason is that the F1 score takes the distribution of the data into account. In the case of an imbalanced data set, the F1 score will provide a more truthful assessment of model performance. In our specific case, the outliers constitute a small percentage of the total data. If a model were to predict all points as non-outliers, would the accuracy still be very high, as the outliers constitute a very small percentage and do not influence the accuracy score very much. As such, the accuracy measure would give a false image of model performance for this specific data set.

According to the performance results, Decision Tree and Random Forest are both optimal models for this data set. Both models return the highest F1 score during testing on held-out data. These two models also return the same accuracy score. Where they differ from each other is when it comes to the precision and recall scores. Decision Tree returns higher a higher precision score than Random Forest (0.59 versus 0.52), while Random Forest returns a higher recall score than Decision Tree (0.65 versus 0.57). Determining the 'true' optimal model for this data set would depend on the needs of the client. If the client values precision more, than Decision Tree would be the optimal model. If the client values recall more, then Random Forest would be the optimal model. Regardless, in all cases the worst performing model is Isolation Forest, with lower scores for all performance measures.

When it comes to the difference between accuracy and F1 scores for each model, some interesting findings were found. For the Decision Tree model, the accuracy score is much higher than the F1 score (0.91 versus 0.58). For the Random Forest model, again, the accuracy score is much higher than the F1 score (0.91 versus 0.58). The same trend continues for the Isolation Forest model, where the accuracy score is much higher than the F1 score. This time, the difference between the two scores (0.85 versus 0.34) is even bigger than for the previous two models. An explanation for this discrepancy is that the data set is imbalanced. Due to the F1 score being the harmonic mean of precision and recall, it is a class balanced accuracy measure. The accuracy score is biased towards the 'True' classification, which in this data set is the majority of the cases. The few falsely classified cases have little influence on the accuracy score, but do become apparent when the F1 score is calculated.

6 Conclusions

To summarize, the goal of this assignment was to use the Decision Tree, Random Forest, and Isolation Forest models to detect an outlier class. The data set provided was related to a Portuguese bank marketing campaign. The data set contained 20 input variables, 1 target variable, 41188 instances, and was provided with normalization already applied. The performance of each respective model was calculated using various performance measures. Cross-validation was used to assess how the models will perform on an independent data set. Parameters were fine-tuned for each respective model to find the settings that deliver optimal performance. The performance results showed that Decision Tree and Random Forest were tied for the best performing model for this specific data set. They both provided the highest F1 score, which is the performance measure we decided to use to determine which model was the best for this data set.

7 Project evaluation

7.1 Contributions of Group Members

Task 1: Exploring the data set
1.1 Exploratory data analysis - Sjoerd
1.2 Creating Train and Test data sets - Mischa

Tasks 2: Decision Trees
2.1 Decision Tree - Christian
2.2 Confusion Matrix and Accuracy - Josh
2.3 Features to Tree - Robin
2.4 Cross validation - Mischa
2.5 Tree Tuning - Mischa

Task 3: Random Forest - Sjoerd

Task 4: Outlier Detection - Isolation Forest - Sjoerd

Task 5: Report your results and discuss your findings - Christian

Bonus Tasks - Josh

Report
Abstract - Robin
Introduction - Robin
Data - Robin
Methods - Mischa
Evaluation and Discussion - Sjoerd
Overview and comparison of algorithm performance - Christian
Conclusions - Mischa

7.2 Group reflection

Looking back on the progress of the project, this went well. During the first seminar, a clear division of tasks was drawn up with concrete deadlines. This allowed us to immediately start programming.

The following week, a meeting was held for the college to discuss progress. Everyone had completed their tasks for this meeting, which was very nice. Here was also the opportunity to ask teammates for help with difficult tasks. During this meeting a division of tasks was made for the rest of assignment 1. Everyone adhered to the division of tasks, which made the progress of the project smooth.

Communication with the team went well. Direct communication took place via a Whatsapp group, more extensive matters via Microsoft Teams. A Kanban board has also been made on Microsoft Teams for the distribution of tasks. Here we could see not only our own tasks, but also those of others.

During the project we took a close look at each other's results. There was also room to be critical, which improved the end result. In short, the project went well and we are looking forward to assignment 2.