INFOB3APML: Assignment 1

Group 2

Aksel Can Sozudogru Coby Simmons Lorenzo Marogna
Minna Vainikainen Tomás Tavares

November 2023

1 Introduction

This report presents an in-depth exploration and application of three critical machine learning algorithms: Decision Tree (DT), Random Forest (RF), and Isolation Forest (IF). The primary focus is on the utilization of these algorithms to predict whether a potential client will subscribe to a term deposit product offered by a bank as a result of a direct marketing campaign. The motivation for this research is that it will allow the banking institutions to understand their potential clients better and thus increase their profits.

This is a binary classification problem (i.e., the class takes only "true" or "false" as values), indicating that we might be able to use tried-and-tested classification methods like DT and RF. Interestingly, this can also be considered as an outlier detection problem, since the distribution of the class is heavily skewed towards "false." Both approaches are explored in this research.

Additionally, parameter tuning is used to optimize the models, and cross-validation is used to prevent over-fitting. The models are evaluated using standard performance metrics like accuracy, recall, precision, and F1 score so that the performance of different models can be compared.

2 Data

The dataset is from a Portuguese banking institution's direct marketing campaigns. Every instance corresponds to a series of data collected during a phone call with the potential client. The features of the dataset include demographic information about the potential client, as well as general information about the contact that has been made, like the duration of the call and the time since the last call (in days).

Table 5 provides the descriptive statistics for the numeric variables. As seen, the dataset contains 40188 instances and 63 features, which seem to have undergone min-max scaling (justifiable by the range of values that are present). The total storage size of the dataset is approximately 7MB when encoded as a CSV and 20 MB when loaded into main memory by the `pandas` Python package.

The data has been preprocessed using one-hot-encoding, which contributes to many features – only 11 features exist on their own and are not the result of one hot encoding. One-hot-encoding is used to remove categorical features from the dataset and instead encode the columns as multiple binary columns, expanding the range of classification algorithms that can be used.

The univariate distribution of values differs among different features in the dataset (plotted in Figure 2). For example, the number of instances is equally distributed among working days (Monday to Friday); however, this is not true for the month feature – the majority of data last contacts occurred in May, July, August, and June.

The distribution of data in numerical features often does not follow a Gaussian distribution. For instance, the age is clearly skewed to the left and finds its peak around 20 years, meaning that the campaign reached more younger

clients.

It can be seen in the correlation matrix in Figure 3 that `euribor3m` is highly correlated with both `emp.var.rate` (0.97) and `nr.employed` (0.95) are highly correlated with each other.

As previously mentioned, the class attribute is extremely imbalanced. Thus, this task can be approached as an outlier detection problem because, according to Figure 2, the vast majority of the population contacted did not subscribe to the term deposit product, making those subjects who did subscribe outliers.

The variables are highly informative as they are specific to the category they represent. Before starting our data exploration, we expected more singles under 25 than married clients. However, when we explored our data, we found more married clients than single clients under 25 (9240 married, 8749 single).

According to the correlation matrix (Figure 3), the `duration` feature is the most correlated with the class, implying that this feature will be significant in predicting the class. However, the documentation accompanying the data (see references) states that this feature should not be used for predictive modeling since the duration of a call is only known after the call takes places and the class is known.

3 Methods

3.1 Data Pre-Processing & Feature Selection

No data pre-processing was undertaken since the data has been normalized. Regarding feature selection, the `duration` feature (representing the duration of a sales call) was removed from the independent variables since it highly affects the class, yet the duration is not known before the sales call is completed.

3.2 Model Selection

The DT, RF, and IF algorithms (implemented in the `scikit-learn` Python library) were used as candidates for optimal models. These models were selected because tree-based models can perform well on smaller datasets (like this one).

3.2.1 Decision Tree

The first method explored was the Decision Tree (DT). This type of algorithm was chosen as it is quick to train, and the results are interpretable.

To start with, a naïve decision tree was trained (i.e., the default parameters were used). The default parameters do not specify a maximum depth, resulting in a hugely complicated decision tree. Such a decision tree would likely lead to over-fitting – a situation whereby a model fits the training data very well, but does not perform well on new data.

It was decided to tune the maximum depth and minimum samples per leaf node parameters using a grid search with the following ranges:

- max_depth: $\{2, 3, ..., 8, 9\}$
- min_samples_leaf: $\{i^2 : i \in \{1, 2, ..., 12, 13\}$

3.2.2 Random Forest

The second method we explored is the Random Forest (RF) algorithm, an ensemble of decision trees. This model improves on DTs because, as an ensemble model, it is robust against overfitting.

Again, we implemented a baseline model with default parameters. Precisely, the default version of the model uses 100 estimators (i.e., 100 DTs) and sets the maximum number of features in each tree to be the square root of the number of features ($\lfloor\sqrt{\text{total\_features}}\rfloor = \lfloor\sqrt{62}\rfloor = 7$). Following this, a grid search is performed with the following parameters:

- n._estimators: $\{20, 80, 140\}$ Is the amount of learners that will be trained.
- max_features: $\{2, 6, 10\}$ Is the number of features that can be used for the learning of an estimator.

3.2.3 Isolation Forest

The third method explored was the Isolation Forest (IF) which approaches the problem as an unsupervised anomaly detection problem. IF was thought to be a good method for predicting the class in this instance, given that the distribution of the class is heavily skewed towards "false" – thus, positive predictions can be thought of as outliers.

Isolation Forest was applied to the full data set with three selected parameters for tuning during the training phase. *Contamination* is the value of how much of the overall data we expect to be considered as an outlier, which we set to a search space between 0 and 0.5. The second parameter is *max samples* which is the number of samples to draw from the training data to train each base estimator. The search space for this parameter is set to the range from 100 to the number of observations. The range of the parameters explored is presented as follows:

- contamination: $\{0, 0.05, 0.1, ..., 0.5\}$
- max_samples: $\{100, 600, 1100, ..., 4000\}$

4 Evaluation & Discussion

To validate the supervised models, we used k-fold cross-validation with 5 folds (a standard choice). This is a robust method for evaluating the performance of machine learning models as it ensures that every observation from the original dataset has the chance to appear in the training and test set. The key advantage of k-fold cross-validation is its ability to mitigate the problem of a model's

performance being dependent on a particular random split of the data, thus preventing overfitting.

Throughout this research, the F1-sore is used to optimize and tune models, as well as to compare performance between models. The F1-score is the harmonic mean of precision and recall. We chose this metric for this particular research domain because we aim to minimize the presence of both false positives and false negatives since both are equally detrimental. Therefore, there is no advantage in optimizing precision over recall or vice versa.

4.1 Decision Tree

It was found that increasing the maximum tree depth parameter contributed to an increase in the training accuracy but a decrease in the validation accuracy. This lends evidence to the earlier assumption that increasing maximum tree depth results in overfitting, as shown in Figure 1.

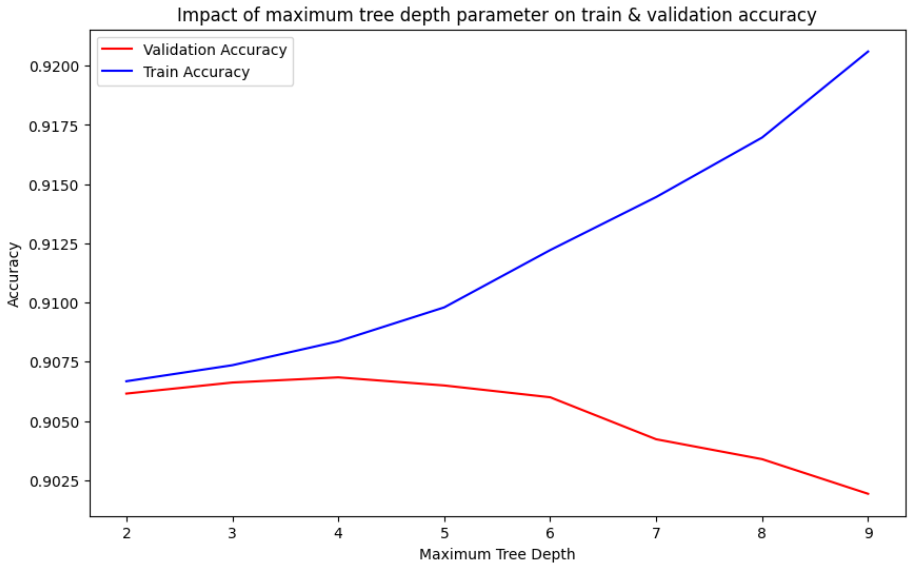

Figure 1

Parameter tuning (which aimed to optimize the F1 score) resulted in slight increases in accuracy, a significant increase in precision, and decreased recall compared to the naïve model.

4.2 Random Forest

It is found that lowering the `number of estimators` and the `max features` leads to worse results, but increasing them more than the default parameters only increases the training time.

Max. Depth	Min. Samples Leaf	Accuracy	Recall	Precision	F1
None (default)	1 (default)	0.85	0.34	0.3	0.32
4	1 (default)	0.91	0.21	0.69	0.32
7	16	0.91	0.23	0.64	0.34

Table 1: Performance metrics for different parameter configurations

We then tested the model on unseen data to evaluate the results:

N. Estimators	Max. Depth	Accuracy	Recall	Precision	F1
100 (default)	$\lfloor\sqrt{62}\rfloor = 7$ (default)	0.9	0.29	0.54	0.37
140	10	0.9	0.29	0.54	0.38

Table 2: Performance metrics for different parameter configurations

The parameter tuning resulted in a model with `max_features = 10` and `n_estimators = 140`, but the improvement in the F1 score is imperceptible.

The performance of the optimal RF model has a higher F1-score than that of the DT; however, the F1-score is still not optimal.

4.3 Isolation Forest

When optimising the IF model for F1-score, the best model had a 0.15 contamination and 2100 max samples. The resulting F1-score of 0.33 is a relatively low value and indicates that the performance of the model is not optimal. Tuning the parameters did not result in significant increases in the training accuracy and precision. However, there was a larger increase of the recall.

contamination	max. samples	Accuracy	Recall	Precision	F1
0	100	0.89	0.21	0.43	0.28
0.15	2100	0.89	0.77	0.43	0.32
0.4	3500	0.56	0.77	0.16	0.26

Table 3: Performance metrics for different parameter configurations

Given that this is the only unsupervised method that was used, we can expect the performance to be worse than that of the supervised methods (i.e., DT, RF).

In order to better understand the characteristics of the outliers, we plotted the distributions of each feature, stratifying the data by whether or not the optimal IF model predicted it to be an outlier. This is displayed in Figure 4.

4.4 Performance Comparison

In this section, we'll look at the results obtained in sections 4.1, 4.2, and 4.3 and compare them.

Table 4 provides a summary of the findings from the previous sections; analyzing such results, we conclude that random forest is the optimal model since it has the highest F1-score of all (38%).

Despite the availability of multiple metrics to compare models, such as accuracy score, recall, precision, and F1-score, we have consistently used the F1-score as our comparison metric throughout this assignment.

This is mainly influenced by the imbalanced nature of our dataset, making the accuracy score misleading in this context because it does not consider the class distribution. If the model is predicting the majority class well, while it may perform poorly on the minority class, this makes the resulting accuracy high. On the other hand, it doesn't affect the F1-score as much since it considers the false positives (FP) and false negatives (FN). Since the dataset is imbalanced, this is exactly what is happening, and it can be observed by the difference between the F1-scores and the accuracy in each model: 57% for the decision tree, 52% for the random forest, and 50% for the isolation tree.

It's also worth mentioning that comparing the performance of supervised algorithms like Decision Trees (DT) and Random Forests (RF) with an unsupervised algorithm such as the Isolation Forest may not be a fair comparison. This is evident from the noticeable difference in performance. The main reason for this distinction lies in the fact that supervised algorithms have more information, enabling them better to fit the data instances to the actual results. Unsupervised algorithms, on the other hand, can only work with patterns in the data to achieve this goal.

Additionally, it should be noted that while IF and RF models have greater potential to grow (given increased computing power) and are generally more robust, DTs are easier to interpret and faster to train. For this use case, the latter might be better, given its efficiency.

Model	Val. Acc.	Test Acc.	Val. Recall	Test Recall	Val. Precision	Test Precision	Val. F1-score	Test F1-score
DT	0.91	0.91	0.23	0.25	0.64	0.71	0.34	0.37
RF	0.9	1	0.29	0.96	0.54	0.99	0.38	0.98
IF	0.89	0.82	0.77	0.36	0.16	0.25	0.32	0.3

Table 4: Summary of Results

5 Conclusion

It was found that no model could predict the class particularly well. There were minor improvements in using RF over DT. Given that RF is an ensemble of DTs, this was to be expected as an ensemble of models aids in reducing overfitting. IF had the worst validation F1-score; however, an advantage of IF is that it is

an unsupervised learning algorithm. Given that we have labeled data, there is no reason to use an unsupervised learning algorithm. Thus, we can conclude that RF is the best model for this use case.

In the future, we might be able to invest in greater computing power in order to test more parameters and a greater range for each parameter, which might lead to improvements in our models. Additionally, we might be able to engineer more useful features in order to achieve better results.

6 Contributions of Group Members

- **Task 1 & 2 - EDA and Decision Tree** - Coby Simmons
- **Task 3 - Random Forest** - Lorenzo Marogna
- **Task 4 - Isolation Forest** - Minna Vainikainen
- **Task 5 & Bonus Task** - Tomás Tavares
- **Report** - ALL

7 Team members

- **Lorenzo Marogna**
- **Coby Simmons**
- **Aksel Sozudogru**
- **Tomás Tavares**
- **Minna Vainikainen**

8 References

Moro, S., Rita, P., and Cortez, P.. (2012). Bank Marketing. UCI Machine Learning Repository. https://doi.org/10.24432/C5K306.

9 Appendix

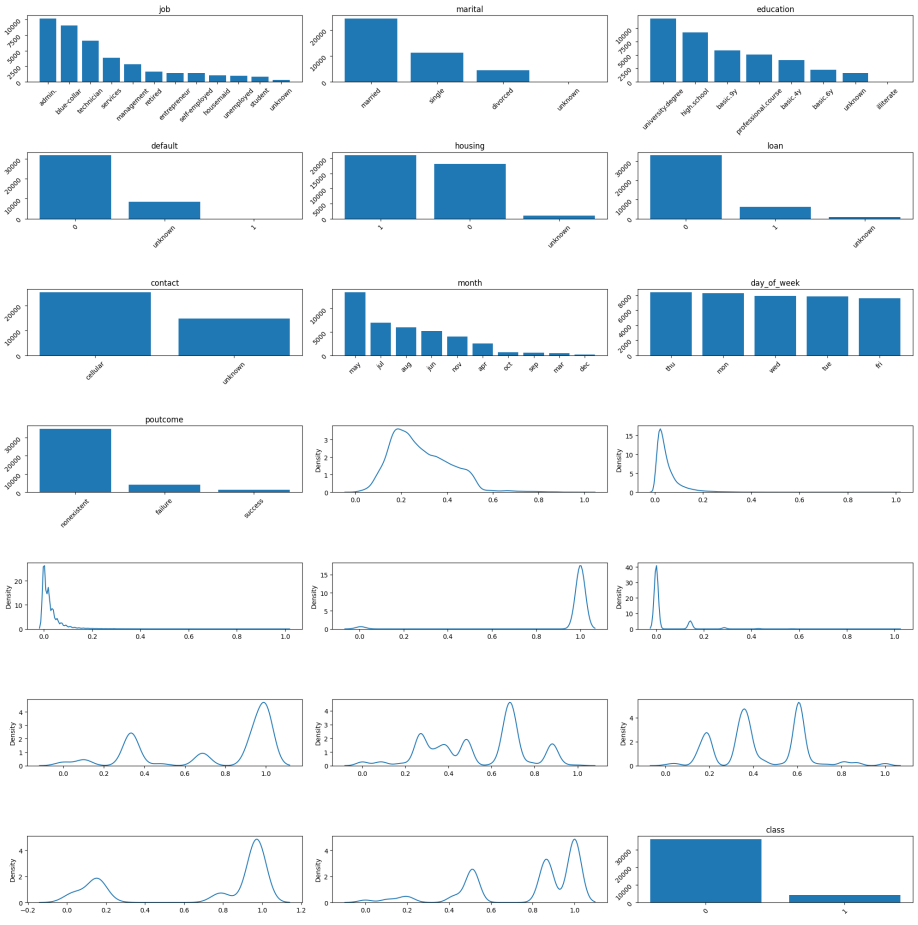

Figure 2: Univariate Distributions of all variables

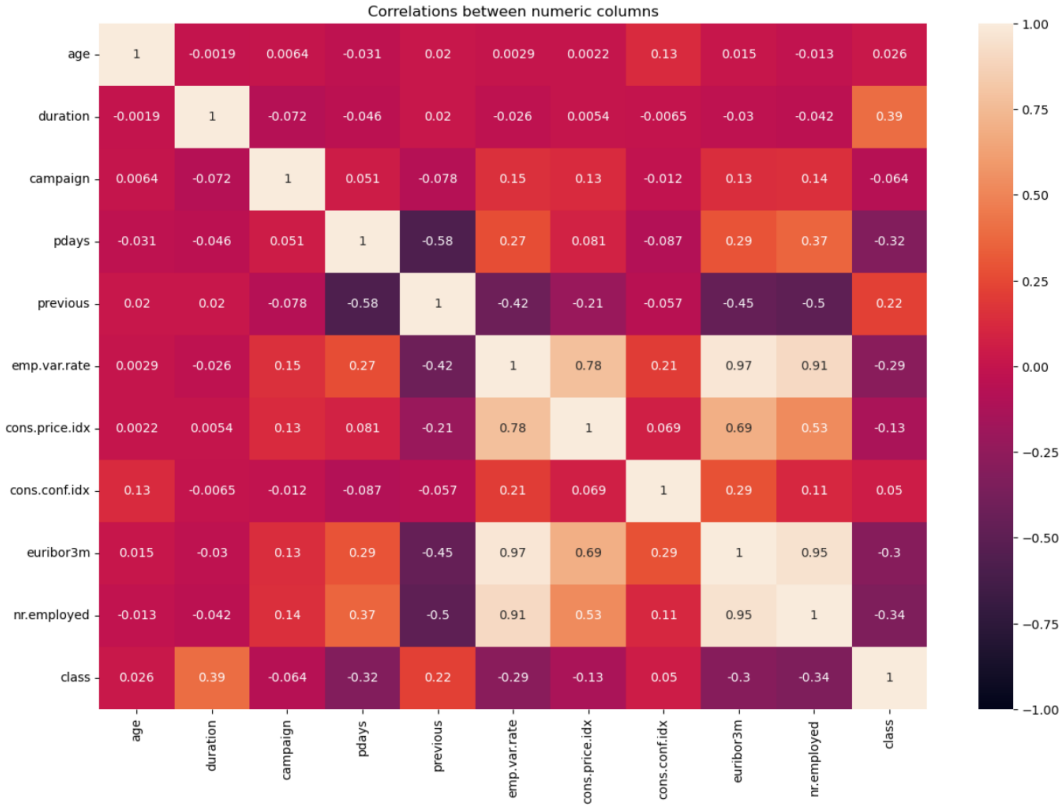

Figure 3: Correlation Matrix

	age	duration	campaign	pdays	previous	emp.var.rate
count	40188.0	40188.0	40188.0	40188.0	40188.0	40188.0
mean	0.28	0.05	0.03	0.97	0.02	0.73
std	0.13	0.05	0.05	0.18	0.07	0.33
min	0.0	0.0	0.0	0.0	0.0	0.0
25%	0.19	0.02	0.0	1.0	0.0	0.33
50%	0.26	0.04	0.02	1.0	0.0	0.94
75%	0.37	0.06	0.04	1.0	0.0	1.0
max	1.0	1.0	1.0	1.0	1.0	1.0

	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	class
count	40188.0	40188.0	40188.0	40188.0	40188.0
mean	0.54	0.43	0.68	0.77	0.1
std	0.22	0.19	0.39	0.27	0.3
min	0.0	0.0	0.0	0.0	0.0
25%	0.34	0.34	0.16	0.51	0.0
50%	0.6	0.38	0.96	0.86	0.0
75%	0.7	0.6	0.98	1.0	0.0
max	1.0	1.0	1.0	1.0	1.0

Table 5: Descriptive statistics for numeric variables

Figure 4: Distributions of all features, stratified by whether a data point is a predicted outlier