Data Science - Bank Marketing Campaign Report

Table of Contents

1. Group Information

Group Name: Datalux

Group Members: 3

Name	Email	Country	Uni/Company
Huu Thien Nguyen	nguyenhuuthien27296@gmail.com	Sweden	Skövde University
Asmaa Alqurashi	asmaa.idk@gmail.com	Saudi Arabia	Taif University
Deepak Rawat	deepakrawat68@gmail.com	Ireland	Dublin Business School

Note: Deepak decided to drop the internship in the last week

Specialisation: Data Science

Submitted to: Data Glacier canvas platform

Internship Batch: LISUM09

2. Problem Description

2.1. Introduction

A common approach for increasing business is to run marketing and selling campaigns. Companies use direct marketing to reach certain categories of clients to achieve a specific goal. Customer distant interactions may be centralised in a contact centre, making campaign administration easier. Technology allows us to reimagine marketing by optimising customer lifetime value through analysing accessible data and customer KPIs, allowing us to develop longer and closer relationships in line with company needs.

The ABC Bank wants to market its term deposit product to clients in this project. Before doing so, they want to construct a machine learning model that will assist them in determining whether a particular consumer would buy their product based on the customer's previous interactions with the bank or other financial institution.

To solve the problem mentioned above, the bank wants to use machine learning modelling to identify the customers who are more reluctant to buy their services so that their marketing channels will only focus on these customers, which in turn will save time and resources and finally leads to optimised cost for this campaign.

2.2. Background

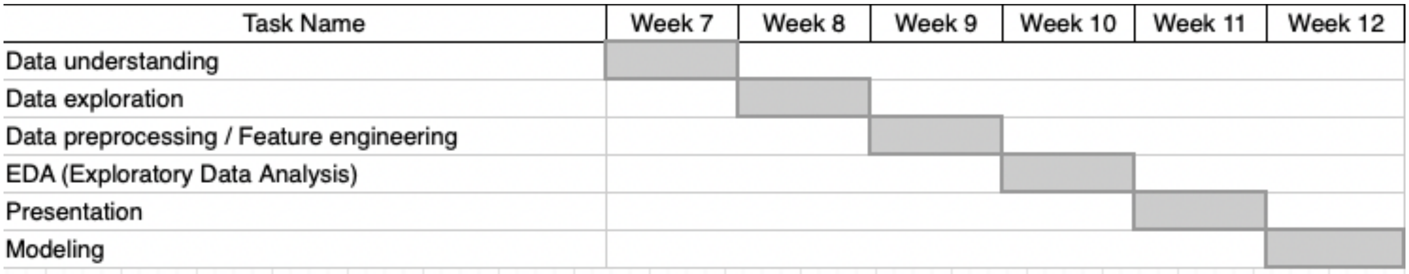
ABC Bank wants to sell its term deposit product to customers. Before launching the product, they want to develop a model which helps them in understanding whether a particular customer will buy their product or not (based on the customer's past interaction with the bank or other Financial Institution). Bank wants to use the ML model to shortlist customer whose chances of buying the product is more so that their marketing channel (telemarketing, SMS/email marketing etc.) can focus only to those customers whose chances of purchasing the product is more.

3. GitHub Link / Data / Project cycle

The link for GitHub: https://github.com/AndrewNguyen27296/DataGlacier

The data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact with the same client was required to assess if the product (bank term deposit) would be ('yes') or not ('no') subscribed.
Size: 41188 records, 20 explanatory variables, and one binary response variable

The project's general view, along with the deadline, is described in the table below. The deadline is added accordingly to the requirements from Data Glacier's canvas page.



Task Name	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Data understanding	▨					
Data exploration		▨				
Data preprocessing / Feature engineering			▨			
EDA (Exploratory Data Analysis)				▨		
Presentation					▨	
Modeling						▨

Table 1: The project's timeline

4. Methods

In order to choose the appropriate model to provide the prediction for the term deposit, our methods followed the four steps to identify the correct techniques to fulfil the requirements. It included data exploration, feature engineering, model deployment, and predictive evaluation. This section aimed to be transparent, concise, and detailed for readers from non-technical perspectives to experts can comprehend and reproduce similar results.

A list of white-box ML models (logistic regression, a simple decision tree, and a Naive Bayes algorithm) and black-box ML models (ridge classifier, SVC, k neighbours classifier, gradient boosting, random forest, and neural network) was implemented to compare which model performs the best on this particular dataset

Multiple classification metrics were utilised to examine the model. It included accuracy, recall, and ROC-AUC. F1 scores were also included.

Accuracy is the metric that evaluates a classification machine learning model's performance by using the number of accurate predictions divided by the total number of predictions. It is the most frequently used statistic for assessing classifier tasks since it is simple to compute and apprehend. In order to understand the evaluation metrics of a classifier model, there are four indicators that readers need to comprehend.

TN / True Negative: the outcome was negative(0) and predicted negative(0)

TP / True Positive: the outcome was positive(1) and predicted positive(1)

FN / False Negative: the outcome was positive(1) but predicted negative(0)

FP / False Positive: the outcome was negative(0) but predicted positive(1)

The efficiency of a classifier to accurately detect all positive cases is measured by the recall, which is a metric of its correctness. It is calculated as the ratio of true positives to the sum of true positives and false negatives for each class.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

The Receiver Operator Characteristic (ROC) is a binary classification problem evaluation metric. It's a probability curve that displays the true positive rate against the false-positive rate at different threshold levels, separating the signal from the noise. The Area Under the Curve (AUC) summarises the ROC curve that measures a classifier's ability to differentiate between categories. The AUC reveals how sufficiently the sample differentiates between positive and negative classes. The more significant the AUC, the better.

Predictive validity is assessed by the F1-score, which balances precision and recall, and is dependent on the accuracy of the model's prediction of consumer sentiment rating. The F1 score is a weighted harmonic average of precision and recall, with 1.0 being the highest and 0.0 being the lowest. Because precision and recall are factored into F1 scores, they are lower than accurate measurements. When comparing classifier models, utilise the weighted average of F1 rather than global accuracy as a rule of thumb.

$$\text{F1 Score} = 2^*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

Since the data has an imbalance output, the technique of under-sampling was implemented to provide a robust classification model.

One method to validate the chosen mode is to split the data into train-valid-test where a few candidate models are trained on the training set then the validating set is used to evaluate the candidate models. After choosing the best-scored model, we save the model and use it in the testing set. This method is used to prevent data leakage.

As we are evaluating the model or hyperparameter, the model has to be trained from scratch, each time, without reusing the training result from previous attempts. We call this process cross-validation.

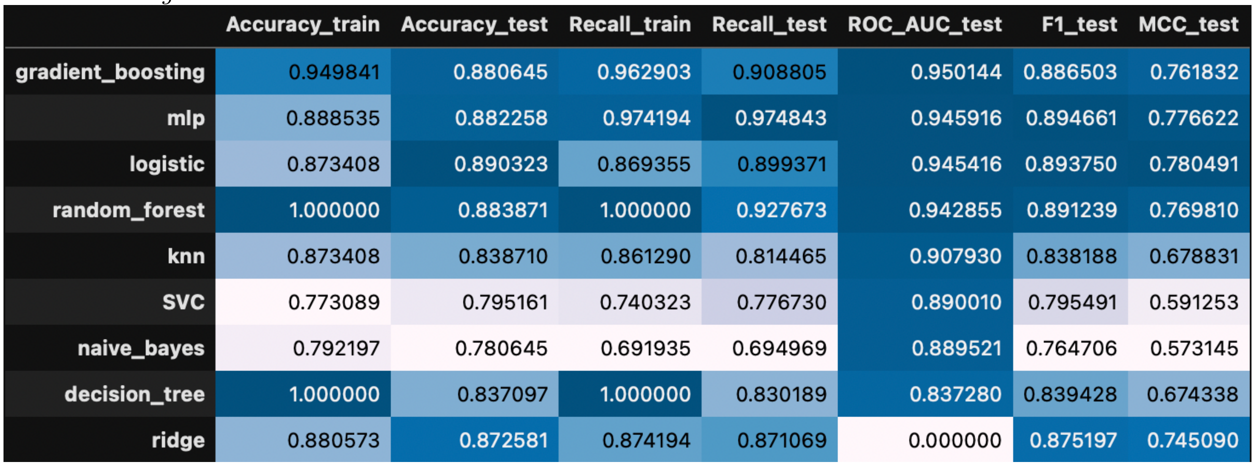
5. Results

Thien

Comparing models – Undersampling – Feature Importance

The detailed metrics of white-box and black-box models are shown in the table below, in which the AUC_test is sorted in decreasing order. The black box functioned similarly to the white box in this case. However, it has a critical flaw: the core algorithm is incomprehensible. The inputs were undersampled to balance out the dependent variable.

Table: The result of the black-box and white-box classifications model applied in the project was sorted by the descending AUC test.



	Accuracy_train	Accuracy_test	Recall_train	Recall_test	ROC_AUC_test	F1_test	MCC_test
gradient_boosting	0.949841	0.880645	0.962903	0.908805	0.950144	0.886503	0.761832
mlp	0.888535	0.882258	0.974194	0.974843	0.945916	0.894661	0.776622
logistic	0.873408	0.890323	0.869355	0.899371	0.945416	0.893750	0.780491
random_forest	1.000000	0.883871	1.000000	0.927673	0.942855	0.891239	0.769810
knn	0.873408	0.838710	0.861290	0.814465	0.907930	0.838188	0.678831
SVC	0.773089	0.795161	0.740323	0.776730	0.890010	0.795491	0.591253
naive_bayes	0.792197	0.780645	0.691935	0.694969	0.889521	0.764706	0.573145
decision_tree	1.000000	0.837097	1.000000	0.830189	0.837280	0.839428	0.674338
ridge	0.880573	0.872581	0.874194	0.871069	0.000000	0.875197	0.745090

As indicated, the gradient boosting provided the highest accuracy train, test, and the ROC score. Additionally, the F1 score showed a harmonic between the recall and precision of the model. However, the classification contains a weakness of a black-box model. It cannot explain the outcome of a prediction. Therefore, if users strive for interpretability, logistic regression can be a suitable replacement.

In order to evaluate the influence of each feature on the classification model. The bar chart below was created to depict the feature important. The euribor3m, poutcome, and marital showed a highly effective score. The model was evaluated with all independent features without pre-processing. Therefore, with the appropriate feature engineering by using domain knowledge, a more powerful classification can be made for this project.

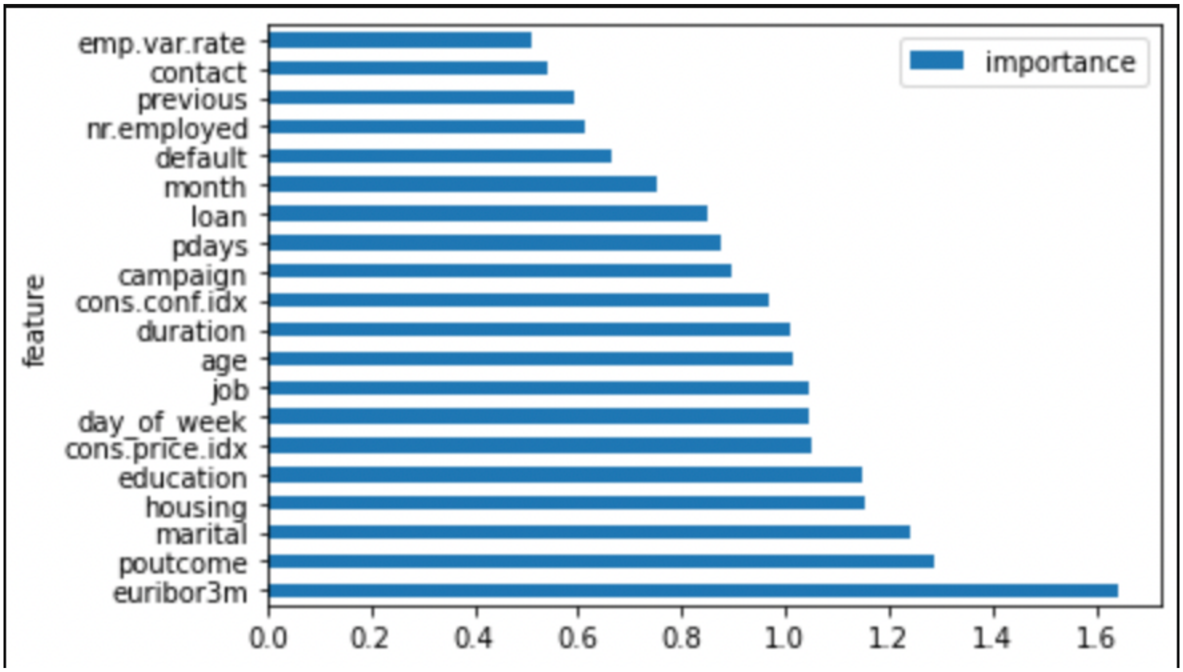


Figure: Feature importance comparison

Asmaa

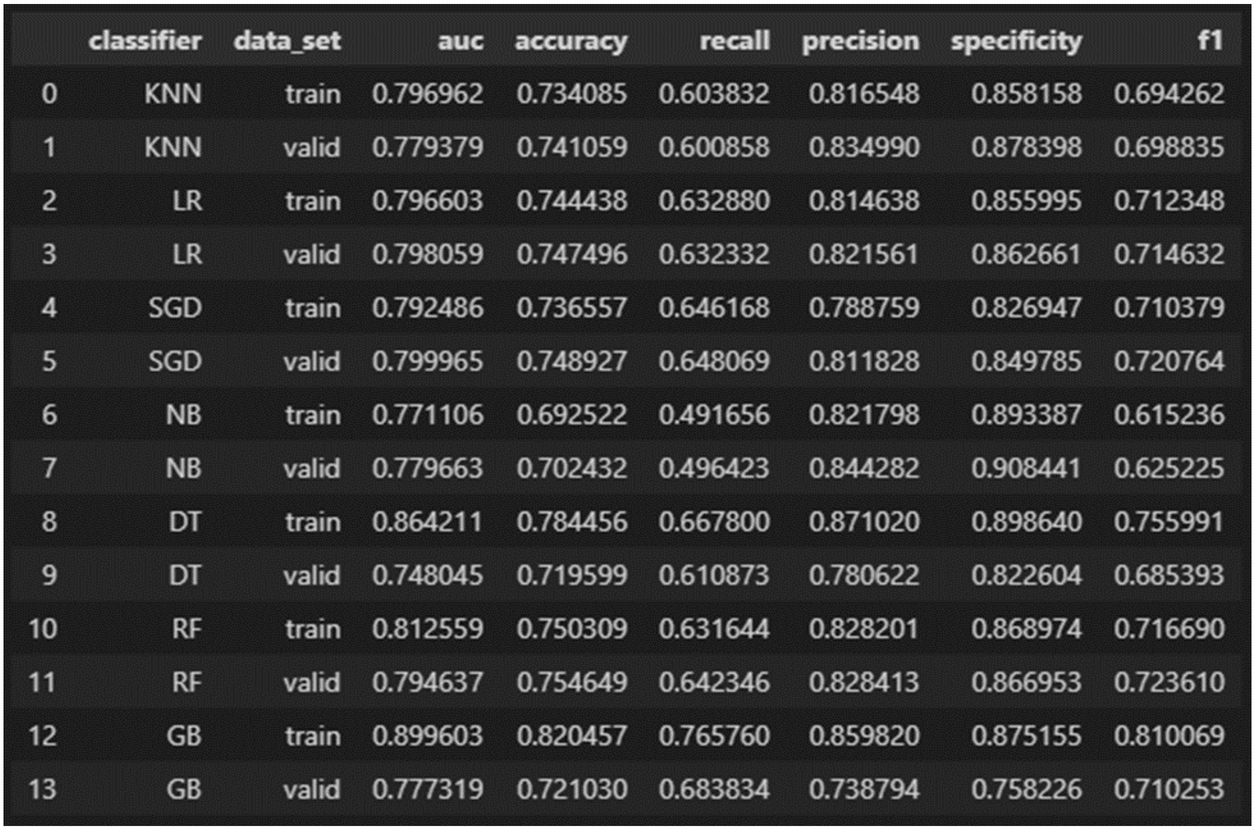
Excluding "Duration" Feature - Undersampling - comparing Models Scores

Choosing the features in the model excluding "duration" was suggested in the requirement since it has a high impact on the results and the call doesn't happen until the targeted clients are decided by the model.

As stated, the data are unbalanced and here the method choosing sen was also undersampling. After shuffling the samples and splitting to 70% train, 15% validation, and 15% test. Each set was balanced and here are the scores of each chosen model.

The function to evaluate the models calculate: AUC, accuracy, recall, precision, specificity, F1

The models chosen are: KNN, Logistic Regression, Stochastic Gradient Descent, Naive Bayes, Decision Tree Classifier, Random Forest, Gradient Boosting Classifier

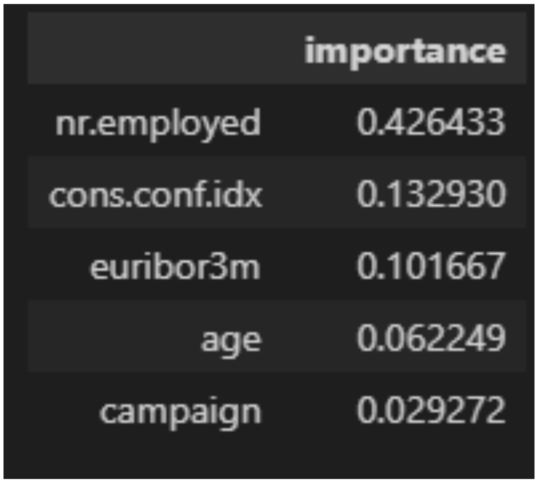


	classifier	data_set	auc	accuracy	recall	precision	specificity	f1
0	KNN	train	0.796962	0.734085	0.603832	0.816548	0.858158	0.694262
1	KNN	valid	0.779379	0.741059	0.600858	0.834990	0.878398	0.698835
2	LR	train	0.796603	0.744438	0.632880	0.814638	0.855995	0.712348
3	LR	valid	0.798059	0.747496	0.632332	0.821561	0.862661	0.714632
4	SGD	train	0.792486	0.736557	0.646168	0.788759	0.826947	0.710379
5	SGD	valid	0.799965	0.748927	0.648069	0.811828	0.849785	0.720764
6	NB	train	0.771106	0.692522	0.491656	0.821798	0.893387	0.615236
7	NB	valid	0.779663	0.702432	0.496423	0.844282	0.908441	0.625225
8	DT	train	0.864211	0.784456	0.667800	0.871020	0.898640	0.755991
9	DT	valid	0.748045	0.719599	0.610873	0.780622	0.822604	0.685393
10	RF	train	0.812559	0.750309	0.631644	0.828201	0.868974	0.716690
11	RF	valid	0.794637	0.754649	0.642346	0.828413	0.866953	0.723610
12	GB	train	0.899603	0.820457	0.765760	0.859820	0.875155	0.810069
13	GB	valid	0.777319	0.721030	0.683834	0.738794	0.758226	0.710253

Figure: Table: Comparing Models Scores

Picking AUC performance indicator over other indicators since it is widely used and an easier metric to compare many models with.

All the algorithms have similar training AUC, but the ones that stood out are decision tree (DT) and gradient boosting (GB). Gradient boosting is considered the best metric to use because it has a higher AUC (0.89) than the other algorithms. At a threshold of 0.5, an AUC of 0.89 is good as it signifies that it is more than just a random guess towards a positive class.
After choosing the high score model let's specify the features' importance in GB. The following table shows the top 5 most important features in the model.



	importance
nr.employed	0.426433
cons.conf.idx	0.132930
euribor3m	0.101667
age	0.062249
campaign	0.029272

Table: Feature Importance Score - Gradient Boosting Classifier

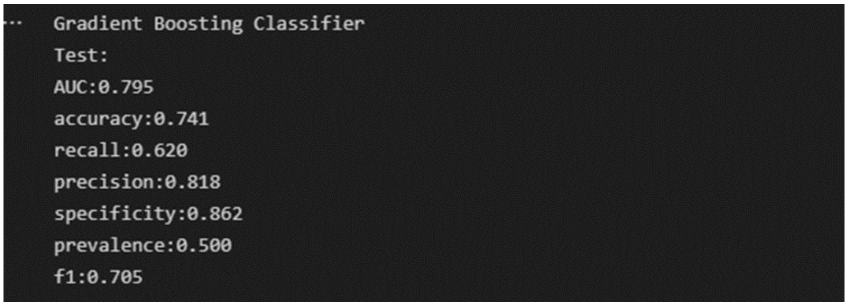
As we choose Gradient Boosting Classifier as the best scoring model, we use cross validation to optimize the results. Using RandomizedSearchCV we define the parameters and create an object. Then we have fitted the train data in it and finally with the print statements we can print the optimized values of hyperparameters. And we got our results:

Optimized GradientBoostingClassifier

Training AUC:0.817

Validation AUC:0.798

After optimizing the classifier, we save it and use it in the testing set, this will prevent data leakage.



```
Gradient Boosting Classifier
Test:
AUC:0.795
accuracy:0.741
recall:0.620
precision:0.818
specificity:0.862
prevalence:0.500
f1:0.705
```

6. Discussion

Overall, this is an excellent project to help understand the whole cycle of a data science project, from collecting the data, preprocessing, modelling, and evaluating results.

Despite the project's progress in implementing interpretation methods, it faced several challenges. As mentioned in the data exploration step, domain knowledge of banking is required because the data contain several features that need to be fully understood to make a feature engineering. If a more experienced analyst analysed the outcome, some interesting insights could be obtained to aid the business decision. For instance, deploying marketing campaign on primary client segment (subscribed term deposit customers), which are married/single, non-existent poutcome, and do not have loans.