Comparative Study on Supervised Machine Learning Classifiers applied to a Bank DataSet

1 Objective of the Study

The performance of specific classifiers is going to be examined when applied to a given dataset. More precisely, information related to bank marketing, comprised of phone calls intending to subscribe clients to bank products, the so called "term deposits", is provided. Aiming at predicting the client decision, that is determining whether a client will accept or reject the subscription to a "term deposit", different classification models employing supervised Machine Learning (ML) classifiers are going to be designed and analyzed.

The remainder of this document is organized as follows.

The adopted strategy for handling the considered problem is introduced in 2.

In § 3 the dataset is explored and different data processes as well feature engineering are presented. Various ML model designs are introduced in § 4 where the related performance is evaluated. Finally, § 5 resumes this study and presents new objectives related to potential future work.

2 Adopted Strategy

For handling the considered problem a set of treating and modeling procedures are going to be conducted. More precisely, various types of ML classification models are going to be designed more or less complex intending to obtain the "best" possible one regarding the related outcomes and reliability. The provided dataset will be investigated and processed in order to provide functional data to the different ML models. An optimized model configuration will be determined after tuning the adequate values for each involved hyperparameter. All models will be experimented on the same dataset while their performance is going to be measured through various evaluation metrics. A relational model appraisal will be stated through comparative evaluation of the resulting outputs.

3 Data Process

3.1 Data Treatment

A dataset consisted of $41,188$ entries is available. Each entry provides information for 21 different features. No missing values are involved in the dataset. A brief description of the feature variables is presented hereafter.

1. age: the age of the related client
2. job: type of job
3. marital: marital status
4. education: the education level of the considered client
5. default: describes whether the client has credit in default
6. housing: describes whether the client has housing loan
7. loan: describes whether the client has personal loan
8. contact: contact communication type
9. month: last contact month of year
10. day_of_week: last contact day of the week
11. duration: last contact duration, (in second
12. campaign: number of contacts performed during this campaign and for this client (includes last contact)
13. pdays: number of days that passed by after the client was last contacted from a previous campaign
14. previous: number of contacts performed before this campaign and for this client
15. poutcome: outcome of the previous marketing campaign social and economic context attributes
16. emp.var.rate: employment variation rate - quarterly indicator
17. cons.price.idx: consumer price index - monthly indicator
18. cons.conf.idx: consumer confidence index - monthly indicator
19. euribor3m: euribor 3 month rate - daily indicator
20. nr.employed: number of employees - quarterly indicator
21. y: informs whether the client subscribed to a term deposit (binary: 'yes','no').

Features $1-11$ concern client data related to the bank where feature "y" is the one to be predicted. In the dataset non numerical features exist, that is features the value of which is not a real number. After vectorization, the values of all features are turned tino numeric ones. There are multiple alternatives allowing vectorization. In this work, the *sklearn* "LeaveOneOut" encoder, a target based coding, is adopted maintaining unchanged the sample dimension space (no additional feature variables are created).

Regarding the feature vectorization, two different approaches are employed:

1. vectorization type 1: all features are involved in the encoding
2. vectorization type 2 : only numeric features are transformed.

From the obtained results, despite the fact that data distribution seems to vary with features implicated in the encoding, the final outcomes result to the same “best” ML models with slight differences in some numerical results of the evaluation metrics.

Figures 1,2,3,4 illustrate the data distribution, the feature correlation heatmap, the feature correlation coefficient and feature pairwise relationship respectively, for the dataset treated according to vectorization type 1.

Similarly, Figures 5,6,7,8 depict the data distribution, the feature correlation heatmap, the feature correlation coefficient and feature pairwise relationship respectively, for the dataset treated according to vectorization type 2.

Along the diagonal of Figures 4,8 the variance of each feature is represented.

Figure observation shows that according to the type of vectorization different data relationships occur.

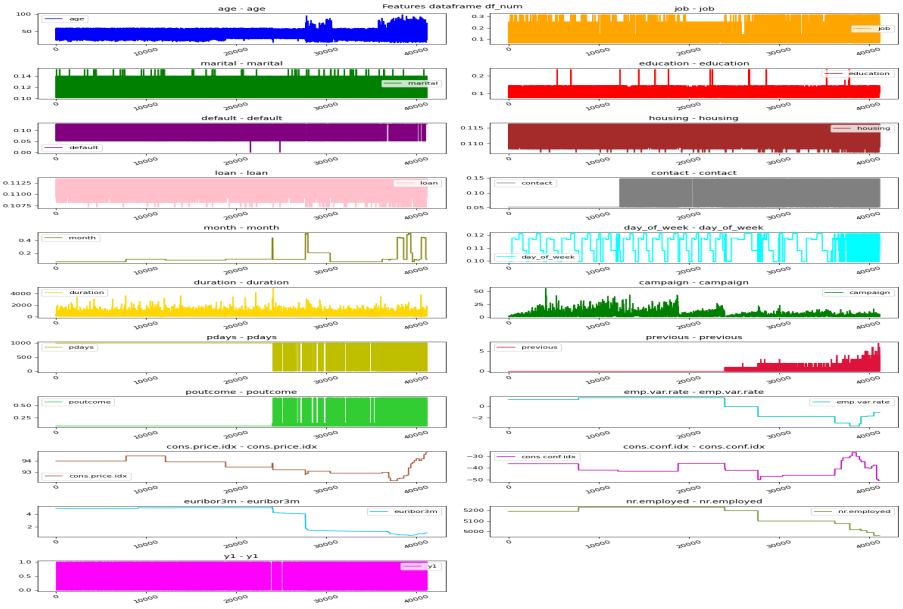

Figure 1: Feature Visualization Plot-Dataset Vectorization Type 1.

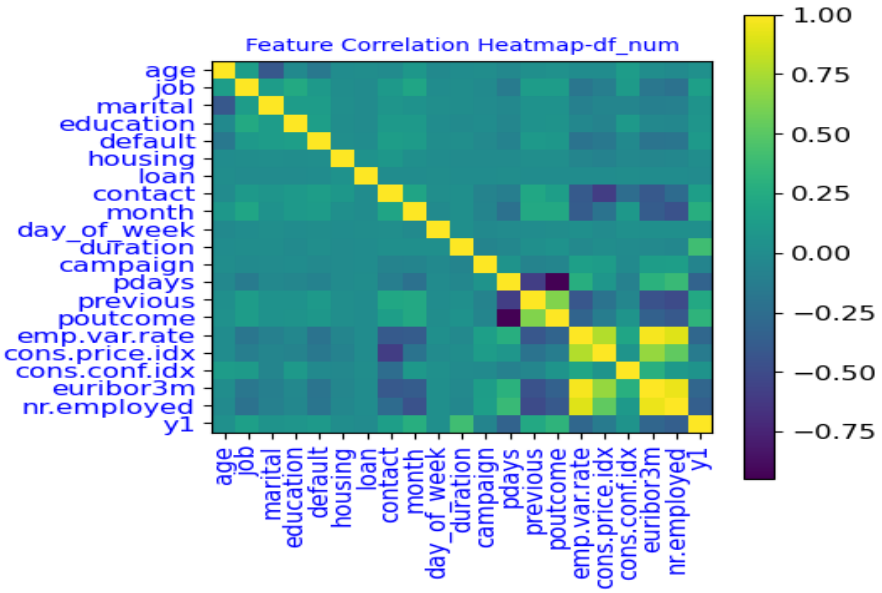

Figure 2: Feature Correlation Heatmap-Dataset Vectorization Type 1.

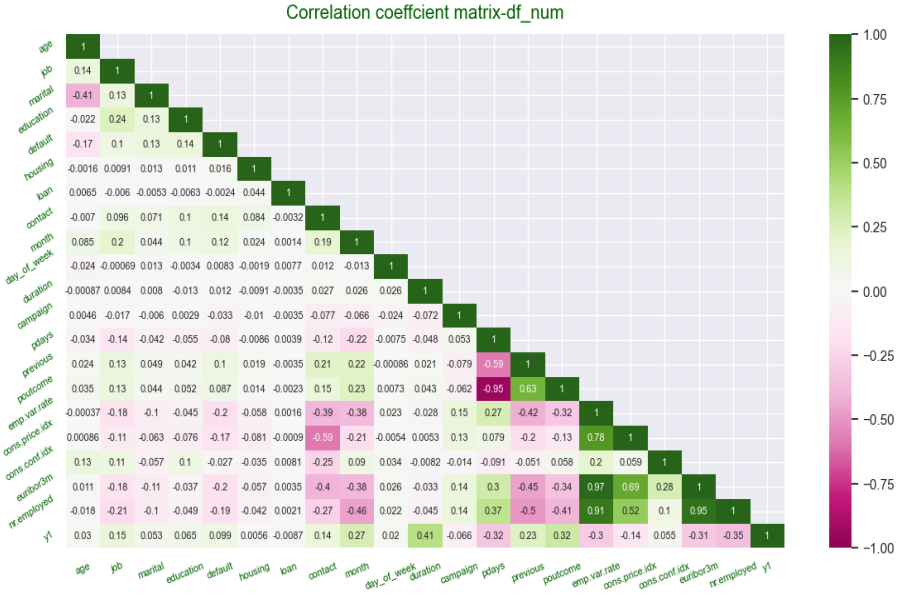

Figure 3: Feature Correlation Coefficient-Dataset Vectorization Type 1.

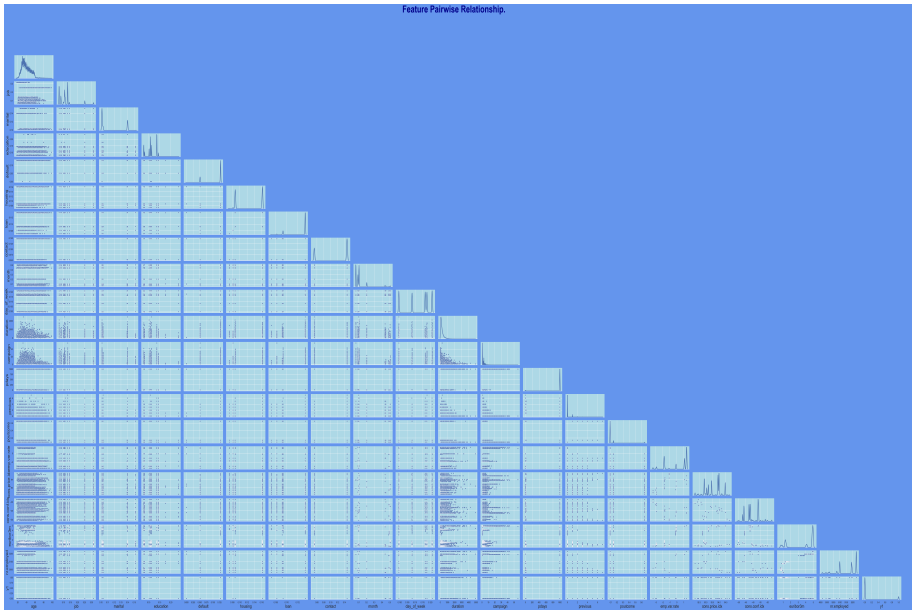

Figure 4: Feature Pairwise Relationship-Dataset Vectorization Type 1.

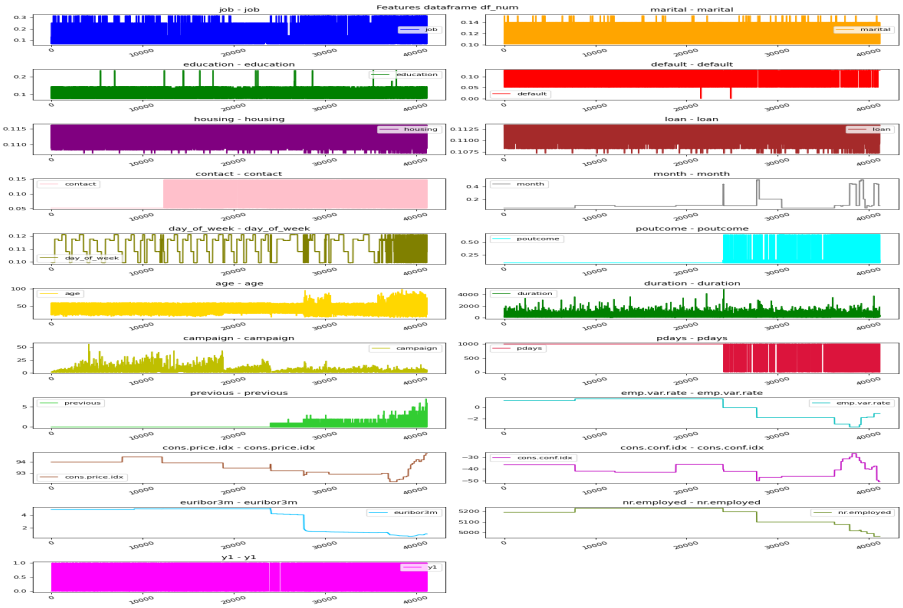

Figure 5: Feature Visualization Plot-Dataset Vectorization Type 2.

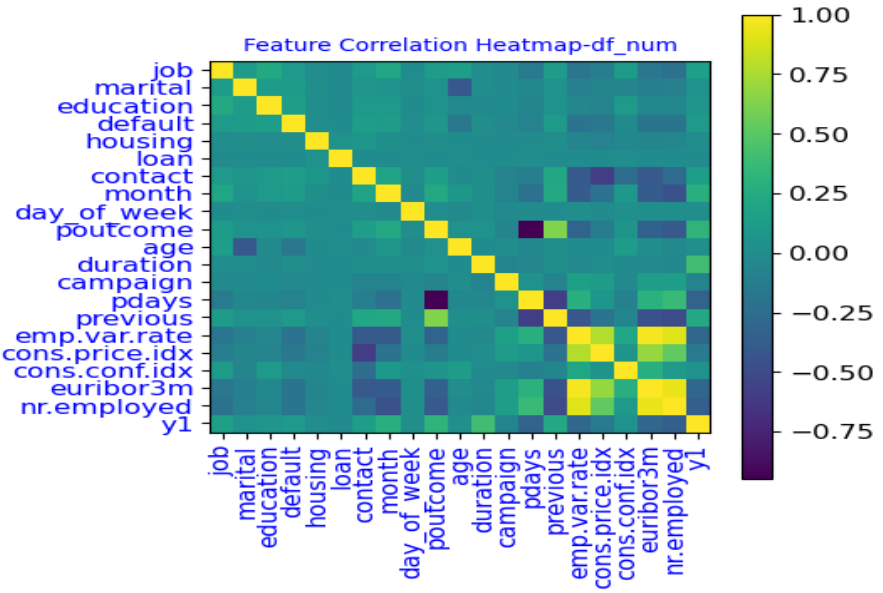

Figure 6: Feature Correlation Heatmap-Dataset Vectorization Type 2.

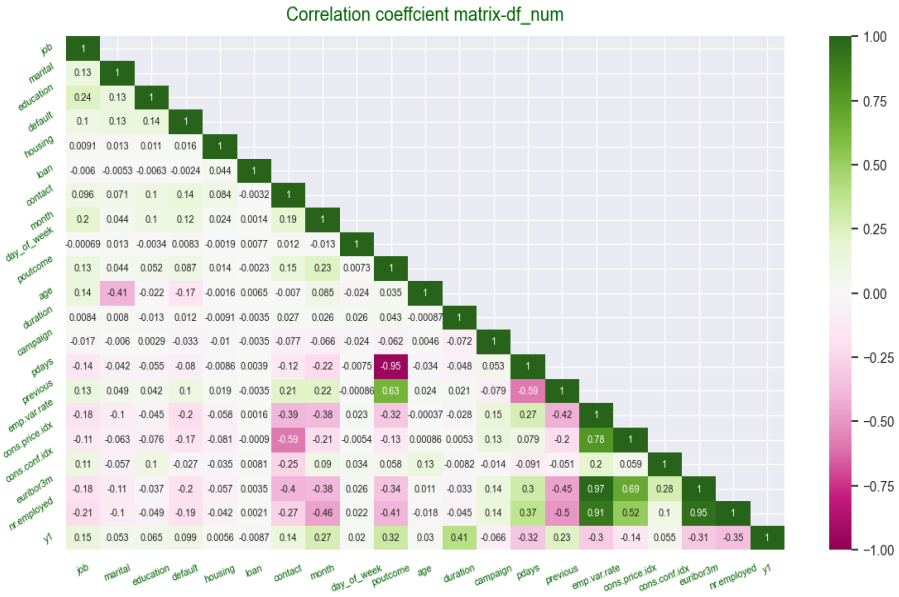

Figure 7: Feature Correlation Coefficient-Dataset Vectorization Type 2.

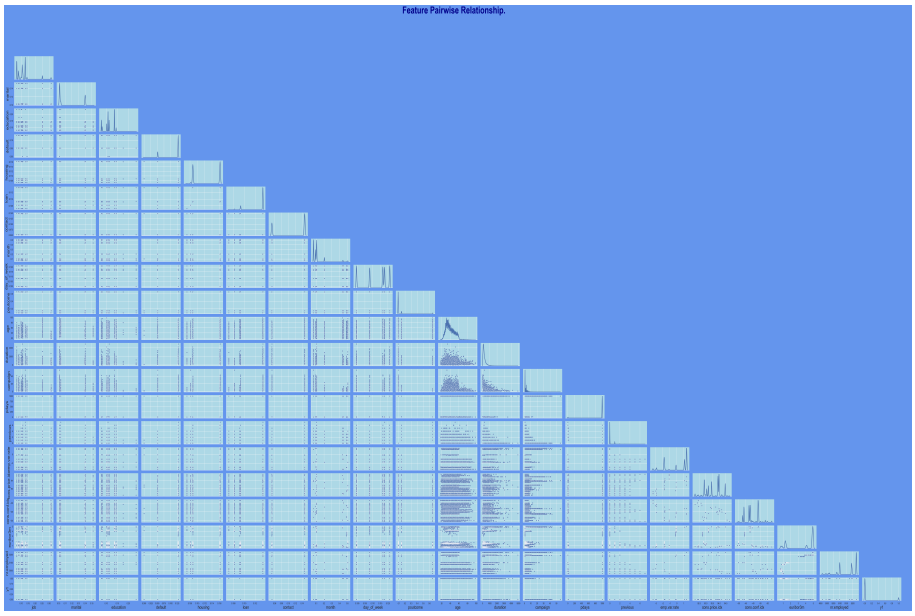

Figure 8: Feature Pairwise Relationship-Dataset Vectorization Type 2.

3.2 Feature Selection

Among multiple possibilities for selecting pertinent features, two different approaches are considered in this study.

3.3 Feature Selection using Random Forest Classifier

For both datasets (vectorization type 1 and 2) the classifier selected the following 6 more important features features: “marital”, “default”, “housing”, “loan”, “contact” and “poutcome”.

Figures 9,10 illustrate the feature classification according to the Random Forest Classifier for the datasets vectorized according to type 1 and 2, respectively. Observation of Figures 9,10 implies a difference in the classification order regarding features associated with importance larger than 6. It worths to be referred that this order may vary from one implementation to another. However, no crucial changes are observed.

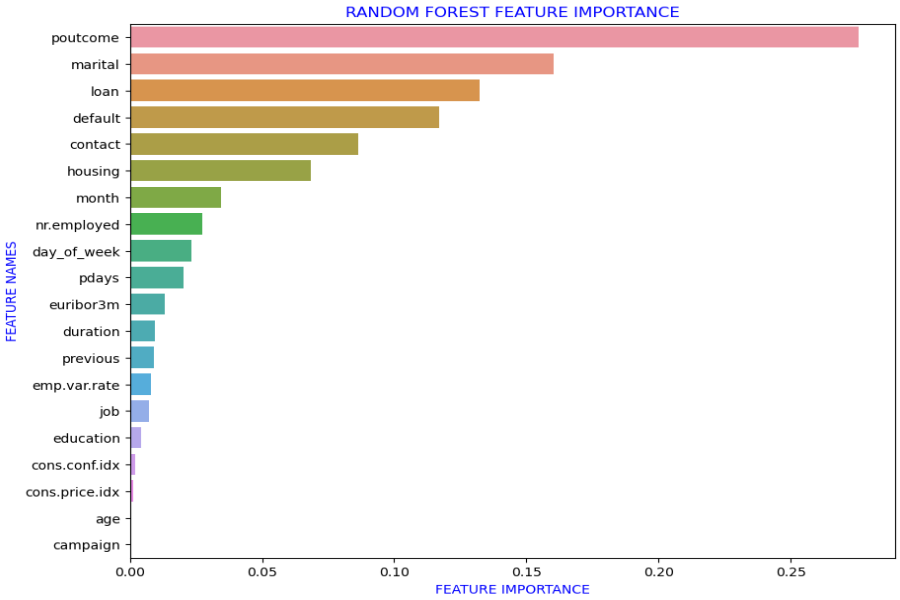

Figure 9: Feature Selection with Random Forest-Dataset Vectorization Type 1.

3.4 Feature Selection using Logistic Regression and L1 penalization

Tables 1,2 illustrate the features (first column of the table) and the related coefficients (second column of the table) obtained with Logistic Regression and L1 penalization for datasets with vectorization of type 1 and 2 respectively. The greater the absolute value of the coefficient is the higher the feature importance. Comparison of the two tables shows a (slight) difference in the feature coefficients for the two datasets.

The provided article recommended to ignore feature “duration”. However, both feature classifiers implied a small importance for this feature regarding all other ones.

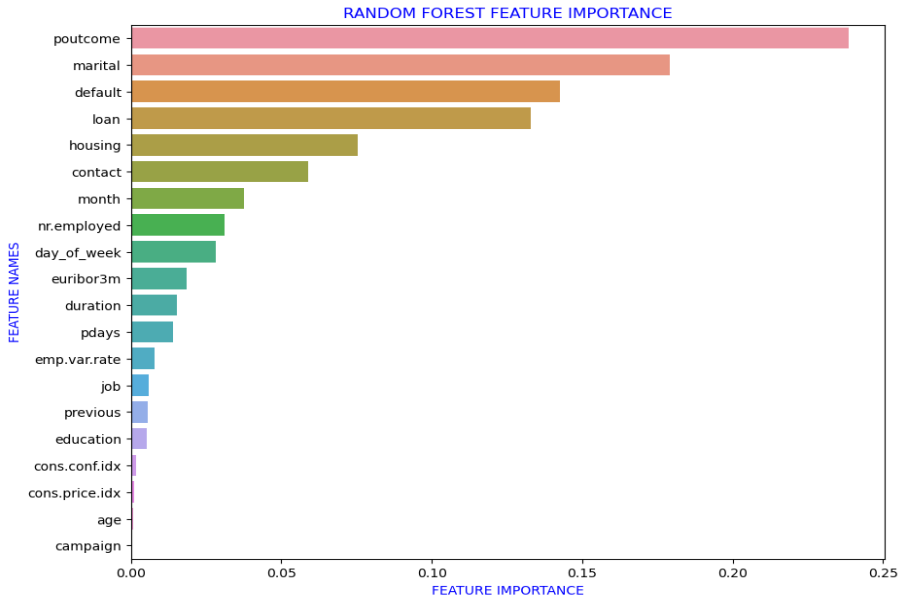

Figure 10: Feature Selection with Random Forest-Dataset Vectorization Type 2.

Table 1: Features Importance with Logistic Regression
Dataset with Vectorization Type 1.

Variable (Feature)	Coefficient
age	0.001246
job	2.487570
marital	0.801548
education	4.241945
default	4.013558
housing	0.000000
loan	0.000000
contact	6.369338
month	1.824071
day_of_week	0.000000
duration	0.004547
campaign	-0.027846
pdays	-0.001608
previous	-0.285009
poutcome	0.517679
emp.var.rate	-0.488541
cons.price.idx	0.413968
cons.conf.idx	0.016107
euribor3m	0.228179
nr.employed	-0.008461

Table 2: Features Importance with Logistic Regression
Dataset with Vectorization Type 2.

Variable (Feature)	Coefficient
job	2.507470
marital	0.827040
education	4.273747
default	4.050508
housing	0.000000
loan	0.000000
contact	6.212288
month	1.859166
day_of_week	0.000000
poutcome	0.506891
age	0.001315
duration	0.004545
campaign	-0.028272
pdays	-0.001616
previous	-0.280119
emp.var.rate	-0.456983
cons.price.idx	0.391234
cons.conf.idx	0.017253
euribor3m	0.186471
nr.employed	-0.008010

4 Model Creation and Performance Appraisal

As previously mentioned different models are going to be designed. Each model will employ one of the four supervised (ML) classifiers: KNeighborsClassifier (KNN), Logistic Regression, Decision Tree and Support Vector Machines (SVMs) Classifier. Before proceeding to the model building a baseline model is going to be considered serving as a reference to the predictive performance of the employed ML models.

4.1 Baseline Model

A very simple solution of the involved problem would be to consider the majority class of all samples.

Figure 11 illustrates the number of samples belonging to each one of the possible values $(0, 1)$ of the (training) target dataset. These values express whether the inferred client rejected or accepted the subscription to a "term deposit". As the considered vectorization technique for the "categorical features" does not modify the target variable the same quantification holds true for both datasets independently of their vectorization type. Hence, the $88,7\%$ of the clients rejected the subscription (value of target variable "y1" equals to zero). The same numbers hold true for the test set too.

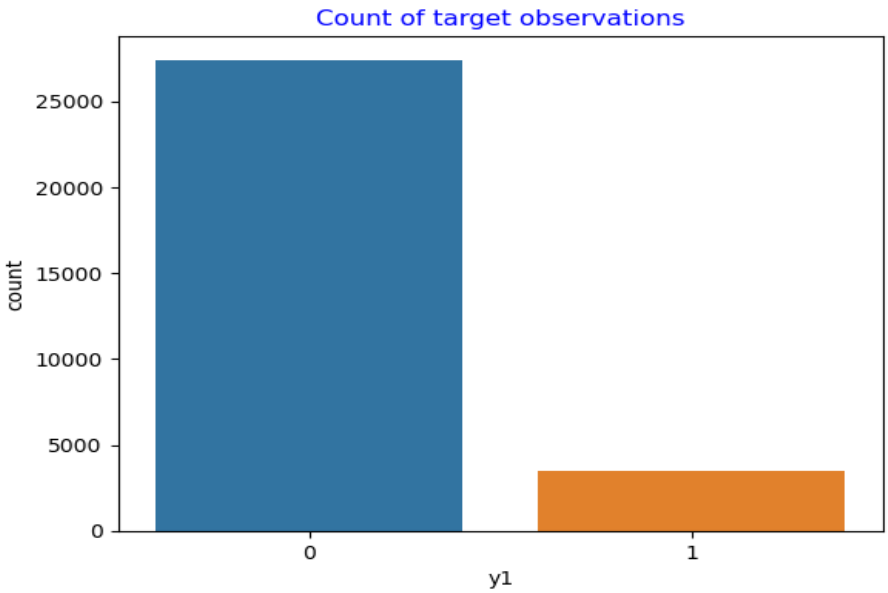

Figure 11: Quantification of Client Acceptance/Rejection Subscription to a Term Deposit.

4.2 Simple ML Models

Models employing simple versions of KNN, Logistic Regression, Decision Tree and Support Vector Machines are now considered. Each such model employs the default values of the related classification algorithm. For the three classifiers, KNN, Logistic Regression and Support Vector Machines the dataset is previously scaled. Models are fitted in all features (for both datasets of type 1, 2 vectorization). The training time, training and test accuracy are computed.

Tables 3,4 illustrate the performance of each classification model for the datasets with vectorization of type 1 and 2, respectively. Observation of the two tables implies that for both datasets the models present similar performance regarding the training time, train and test accuracy. However, mostly due to different delay values potentially occurring within the employed pipelines, the presented values (especially the train duration) may change within different implementations utilizing the same data. Repeating multiple implementations and considering the average value for each metric could increase the stability of the evaluation numbers. Tables 3,4 clearly show that each classifier operates better than the developed baseline model the accuracy of which is 88.7%.

4.3 Improving ML Models

For each model, different hyperparameter values are going to be considered.

Hence for the KNN classification model:

1. values $3, 7, 10, 30, 70$ are examined regarding the number of neighbors to be used by the related classifier

Table 3: Simple Model Evaluation-Dataset with Vectorization Type 1.

Model	Train Time	Train Accuracy	Test Accuracy
KNN	0.013	0.926386	0.904147
Logistic Regression	0.066	0.909197	0.911916
Decision Tree	0.089	1	1
SVM	8.006	0.921725	0.914538

Table 4: Simple Model Evaluation-Dataset with Vectorization Type 2.

Model	Train Time	Train Accuracy	Test Accuracy
KNN	0.013	0.926386	0.904147
Logistic Regression	0.065	0.909197	0.911916
Decision Tree	0.168	1	0.999903
SVM	8.232	0.921725	0.914538

2. "uniform", "distance" are the candidate values for hyperparameter "weights" indicating the weight function to be used in prediction.

Concerning the Logistic Regression classification model:

1. values $100, 1000$ are explored for the hyperparameter "max_iter" indicating the maximum number of iterations taken for the solvers to converge.

Regarding the Decision Tree model:

1. for hyperparameter "criterion", indicating the function measuring the quality of a split, values "gini" for the Gini Impurity and "entropy" are considered

2. hyperparameter "min_impurity_decrease", corresponding to the induced threshold regarding the impurity decrease, values $0.01, 0.02, 0.03, 0.05$ are examined

3. hyperparameter "max_depth", representing the maximum depth of the tree, is explored for values $2, 5, 10$

4. finally, hyperparameter "min_samples_split" ,representing the minimum number of samples required to split an internal node, is studied for values $0.1, 0.2, 0.05$.

Lastly, for the SVMs classification model:

1. hyperparameter "kernel", specifying the kernel type to be used, the considered candidate values are "rbf" and "sigmoid"

2. hyperparameter “gamma”, indicating the kernel coefficient, values $0.1, 1, 10, 100$ are studied

3. hyperparameter “C”, determining the regularization strength, values $1, 10, 100$ are experimented.

Through Grid Search the “best” classification model is provided, configured according to optimized hyperparameter values (among the considered ones). Five evaluation metrics, “accuracy”, “f1”, “precision", “recall” and “roc_auc” are employed along with a Grid search defining the “best” model for the related metric.

Hence, metric

1. “accuracy” corresponds to the “subset accuracy” where the set of labels predicted for a sample must exactly match the corresponding set of labels in the target set

2. “precision” is defined as the ratio of the true positives over the sum of the true and false positives. Intuitively, “precision” corresponds to the ability of the classifier not to label as positive a sample that is negative

3. “recall” is the ratio of true positives over the sum of true positives and false negative evaluations. Intuitively, “recall” presents the ability of the classifier to find all the positive samples. The best value is 1 and the worst value is 0

4. “f1” can be interpreted as the harmonic mean of the precision and recall, where an “f1” score reaches its best value at 1 and worst value at 0

5. “roc_auc” curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It expresses the model capability of distinguishing between classes. The higher the AUC (area under the curve), the better the model performance is at distinguishing between the positive and negative classes.

Each classification model is fitted and tested on the six most important features determined by the Random Forest Classifier.

Table 5 illustrates the “best” classification model associated with the optimized hyperparameters configuration for each one of the considered evaluation metric. The first column of Table 5 indicates the related metric where the second column presents the value of the metric (that is the model performance on the dataset). Finally, the third column refers the classifier and the optimized values for the required hyperparameters to be adjusted. The current implementations showed the same results for both datasets (vectorization of type 1, 2). Observation of Table 5 implies that each model presents a greater performance of the considered baseline model associated with an accuracy of $88, 7\%$.

Table 5: Best Classifier Per Metric-Dataset with Vectorization

Evaluation Metric	Value	Best Model
“accuracy”	0.99961	SVC Classifier: C=100, gamma=10
“f1”	0.99827	SVC Classifier: C=100, gamma=10
“precision”	1	Decision Tree Classifier: max_depth=2, min_impurity_decrease=0.01, min_samples_split=0.1, random_state=42
“recall”	1	Decision Tree Classifier: criterion=“entropy”,max_depth=5, min_impurity_decrease=0.01, min_samples_split=0.1, random_state=42
“roc_auc”	0.9999	SVC Classifier: C=100, gamma=10

5 Discussion and Work Extension

A preliminary study comparing the performance of supervised machine learning algorithms applied on a dataset involving bank information is introduced. Hence, aiming at predicting client acceptance or rejection of bank proposals regarding “term deposits”, various machine learning classification models are constructed.

Each of these models employs supervised classifiers such as KNN, Logistic Regression, Decision Tree and SVMs. More or less complex models are constructed and explored on the same dataset, involving multiple client personal, professional characteristics as well specific bank information for each considered client. Thus, simple models involving the previously referred classification algorithms with their default hyperparameter values are tried.

Additionally, configurable classifiers are designed and experimented for various hyperparemeter values. The “best model” and the related optimized hyperparameter values is obtained through Grid Search. Furthermore each “best model” is appraised regarding multiple evaluation metrics such as “accuracy", “f1”,“precision”,“recall” and “roc_auc”. As a result and regarding the considered implementations, all these models result a greater performance than the considered baseline model always predicting the majority class. However, this statement remains an observation and concerns only the considered implementations and dataset.

Many interesting objectives remain to be studied. Thus,

- concerning the proposed models, they should be explored for a greater number of hyperparameters associated with a larger range of values. Furthermore, in order to increase the stability of the numerical outputs, many implementations could be tried allowing the computation of the average value for each output

- additional techniques for feature engineering and feature selection should be examined aiming at improving the acquired data knowledge and/or decreasing the dimension of the sample space. Examination of “majority voting techniques” allowing feature importance classification should also be considered

- during the proposed work, a specific splitting technique of the original dataset was adopted aiming at minimizing the effects of a large imbalance in the distribution of the target classe. Thus, the resulting train and test sets contain approximately the same percentage of samples of each target class as the complete set. Nevertheless, in order to obtain a better understanding of the data alternative splitting approaches should be examined

- additional classification algorithms should be explored such as approaches combining unsupervised learning techniques, Naive Bayes classification, Neural Networks. Semi-Supervised Learning (SSL), see [1], [2],[3],[4], is a hybrid class of ML algorithms utilizing both labeled and unlabeled data, treating samples differently depending upon whether a label is available or not.

  With reference to the present problem, such a technique might be interesting to be explored, especially if supplementary data, not necessary labeled ones, could be collected providing extra information on additional client characteristics as well on the bank consultants interacting with clients during campaigns.

References

[1] X. Zhu, *Semi-Supervised Learning*. Boston, MA: Springer US, 2010, pp. 892–897. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_749

[2] Y. Xiangli and S. Zixing, *A Survey on Deep Semi-supervised Learning*. IEEE, 2021.

[3] L. Cances and E. Labbé, *Comparison of semi-supervised deep learning algorithms for audio classification*. SpringerOpen, 2022.

[4] N. Li and A. Martin, *Combination of supervised learning and unsupervised learning based on object association for land cover classification*. hal-01922096, 2018.