Bank Marketing Analysis ---- whether the client will subscribe a term deposit

Data Science Initiative DATA1030 Final Report----Qingyan Guo

Github: https://github.com/gfreya/DATA1030-Bank-Marketing-Analysis

1. Introduction

Over the course of the latest technology revolution and increasing globalization, one of the industries that is being influenced the most by the escalation of volume and variety of data is bank marketing. Bank industry begins to implement the direct marketing campaigns (phone calls) to attract clients' attention to specific products or services. Bank industry comes up with an effective strategy to offer term deposit with appreciable interest rate via direct marketing campaigns (phone calls), however, it is necessary to improve the efficiency: lesser direct campaign should be tolerated, but a considerable number of successes (subscribing a term deposit) should be kept.

This project aims to do a binary classification task: predict whether the client will subscribe a bank term deposit based on the data of bank marketing. The dataset consists of 4,119 data points with 20 independent variables and 1 target variables. A data mining (DM) approach has been proposed to predict the success of bank telemarketing [1] where the researchers come up with the data-driven approach using the similar dataset as the one in our project. The researchers compare four DM models and summarize that the best result is presented by neural network (NN) with an AUC score of 0.8 and an ALIFT of 0.7.

Several works have used machine learning methods to improve the bank marketing campaigns. The authors introduce the concepts of 'confidence measurement' and use 'lift' as evaluation criterion to solve the problem of inadequacy of binary classification algorithms [2]. Machine learning algorithms are suitable to work on a classification problem where the task is to predict the label of a data point into target classes (i.e., "yes" or "no" in our project).

2. Exploratory Data Analysis

The dataset used in this project has 20 variables out of which 10 are categorical features and 10 are continuous features. We first investigate the target variable 'deposit', and we find that the data is imbalanced (figure 1). Then we explore 'duration' variable which is highlighted in description file (figure 2). As suggested in the file, we will drop this variable in our project. We have some interesting findings in both categorical features and continuous features.

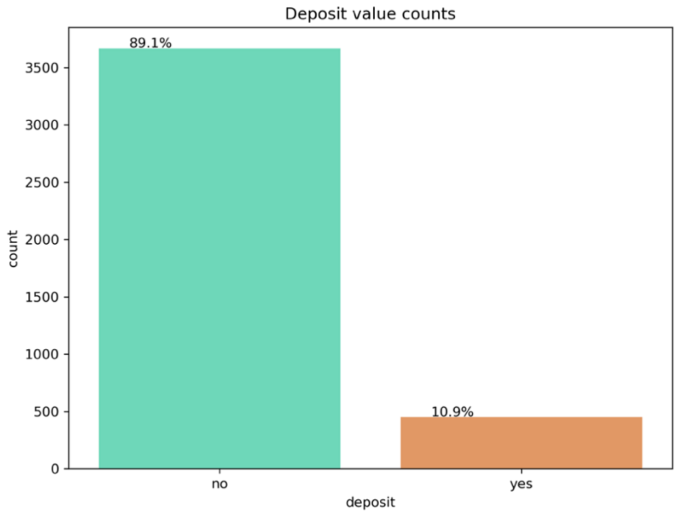

Figure 1. Counts for Deposit Value

Figure 1. This bar plot shows the distribution of target variable y (deposit). The dataset is imbalanced, where the number of people who will not subscribe a term deposit is almost 8 times the number of ones who will subscribe a term deposit.

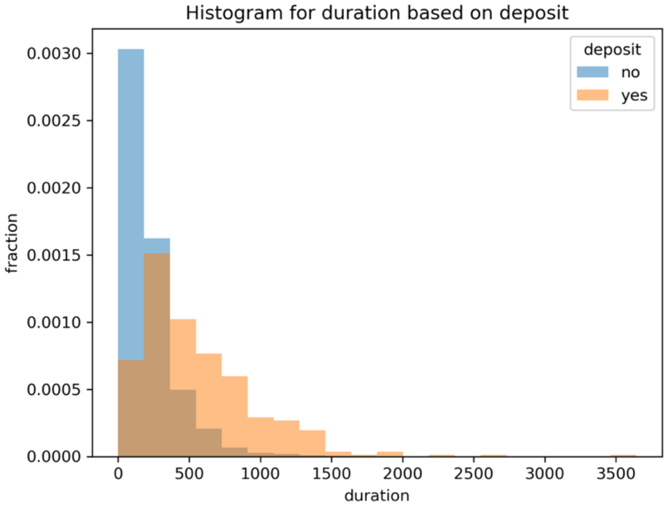

Figure 2. Histogram for Duration Based on Deposit

Figure2. This histogram denotes the duration of the last contact. As mentioned in the description of dataset, this attribute highly affects the target variable. As we can see that if the duration=0, then deposit=’no’. Duration is not known before a call is performed. Also, after the end of the call, the result of subscribing a term deposit is determined. Thus, this input should be discarded if the intention is to have a realistic predictive model as the two are highly correlated and have direct causal relationship.

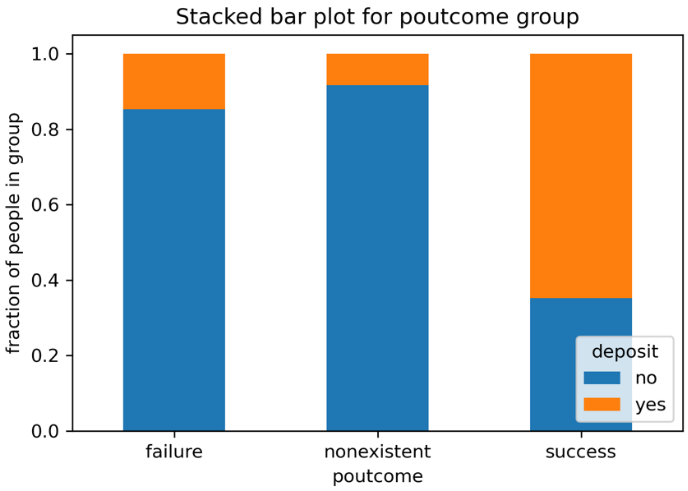

Figure 3. Stacked Bar Plot for Previous Marketing Campaign Outcome

Figure 3. For most of the customers, the previous marketing campaign outcome does not exist, which means most of the clients are new clients who haven't been contacted earlier. There is one thing to note here that, for the clients who had a successful outcome from previous campaigns, majority of those clients did subscribe for a term deposit.

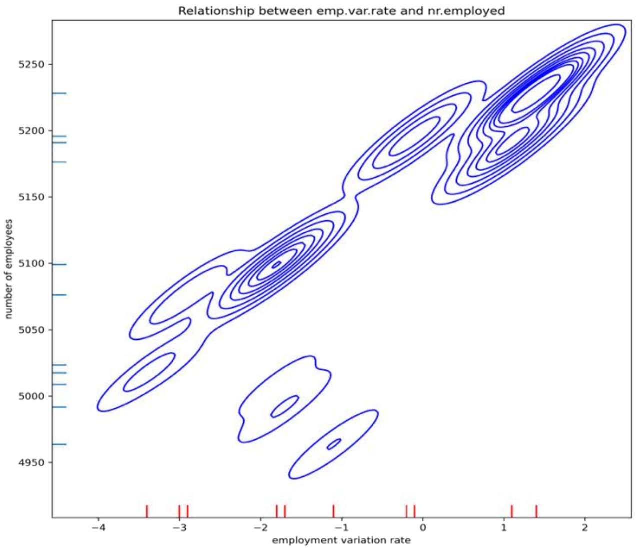

Figure 4. Relationship between Employment Variation Rate and Number of Employees

Figure 4. This density plot determines the interdependency between 'emp.var.rate' and 'nr.employed'. The features employment variation rate and number of employees are numerical variables, and they are related to employment. It is not surprised to find they have high correlations.

3. Methods

3.1 Data Splitting and Preprocessing

As each data point represents the basic information, campaign results, social and economic influences of one client, the bank market analysis dataset is independent and identically distributed without group structure and time-series properties. The whole dataset is very imbalanced: the class ‘yes’ is only one nine of the dataset. Random oversample method can make the dataset very balanced. So, the dataset has 7,336 data points with 3,778 for each class. We can use basic data split method adding the kfold validation to the other set. In each cross validation, the preprocessor will fit and transform on the training set first. As a result of exploratory data analysis, we drop the variable ‘duration’. Currently, 10 categorical features and 9 continuous features left. There are no missing values in the dataset, and we can just consider the ‘unknown’ category as a new category for corresponding categorical features. The feature ‘education’ is ordinal and categorical, so we will use OrdinalEncoder. For the left 9 categorical features, we use OneHotEncoder. We use StandardScaler to all 9 continuous features. After we finish preprocessing works, there are 55 features in total. Also, we replace ‘no’ with 0 and ‘yes’ with 1 for the target variable ‘y’ (deposit) as it only has two labels. We can see the distributions of original data points and the oversampled data points in 2D (figure 5).

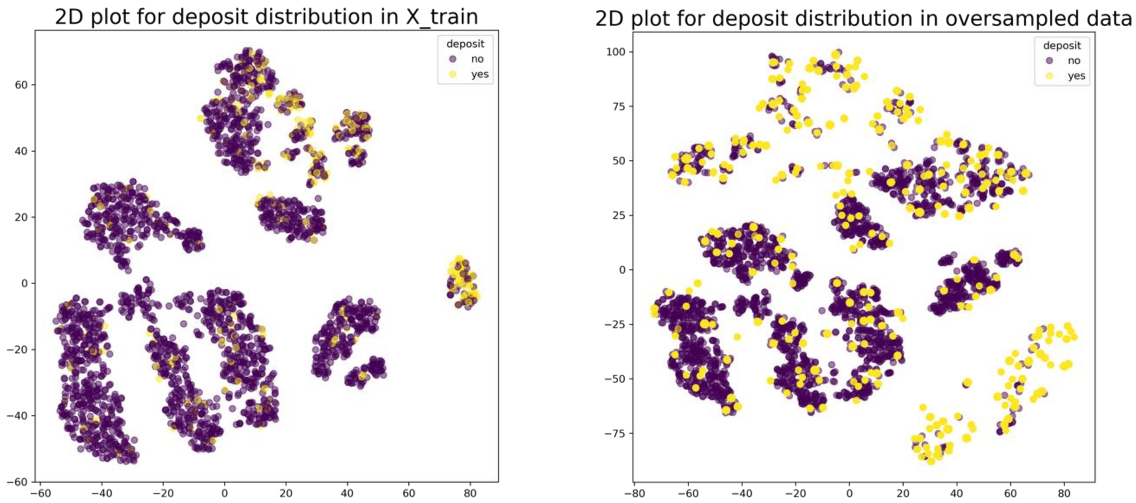

Figure 5a. 2D distribution for original data

Figure 5b. 2d distribution for oversampled data

Figure 5. Comparisons between distributions of original data and oversampled data

Figure 5. 2D visualizations for original data and oversampled data.

Generate new samples by randomly sampling with replacement the current available samples. Even though there seems to be some overlap in the data, there is also some distinction between the two classes in deposit.

3.2 Model Selection

After basic splitting with kfold validation and preprocessed part, ten different machine learning models were trained and compared. All models use the GridSearchCV function to tune the hyperparameters to find the optimal parameter combination for each model. The parameters are summarized in table 1. It is repeated on 10 different random states for 10 different splits. Below are the parameters tuned and values tried for each model:

MODELS	PARAMETERS
LR	**penalty**: 'none'; **solver**: 'sag','saga'; **max_iter**: 10000
LR (L1)	**C**: 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2; **solver**: 'sag','saga'
LR (L2)	**C**: 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2; **solver**: 'sag','saga'
LR (EN)	**C**: 1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2; **solver**: 'saga' **l1_ratio**: 0.01, 0.1, 0.25, 0.5, 0.75, 0.9, 0.99
RFC	**max_depth**: 1, 3, 10, 30, 100; **max_features**: 0.5,0.75,1.0
DT	**Criterion**: "gini", "entropy"; **splitter**: "best", "random"; **max_depth**: 1, 3, 10, 30, 100, None; **min_samples_split**: 1, 3, 6; **min_samples_leaf**: 2, 4
SVC	**gamma**: 1e-2, 1e-1, 1e0, 1e1, 1e2, "auto"; **C**: 0.1,0.3,1.0,3.0,10.0
KNN	**n_neighbors**: 1, 2, 3, 5, 10, 30, 100; **weights**: 'uniform', 'distance'
XGBOOST	**min_child_weight**: 1, 3, 5, 7; **gamma**: 0, 0.1, 0.2, 0.3, 0.4; **subsample**: 0.45,0.6, 0.75; **colsample_by_subtree**: 0.3, 0.4, 0.5, 0.7, 1
ADABOOST	**base_estimator:** DecisionTreeClassifier(class_weight = "balanced"), LogisticRegression(n_jobs = -1, class_weight = 'balanced'); **learning_rate**: 0.001,0.01, 0.1, 1.0

Table 1. Tuned parameters for each model

Table 1. Summarize tuned parameters.

After tuning, each grid search's best model parameters were extracted and used for comparison on accuracy score on the test set.

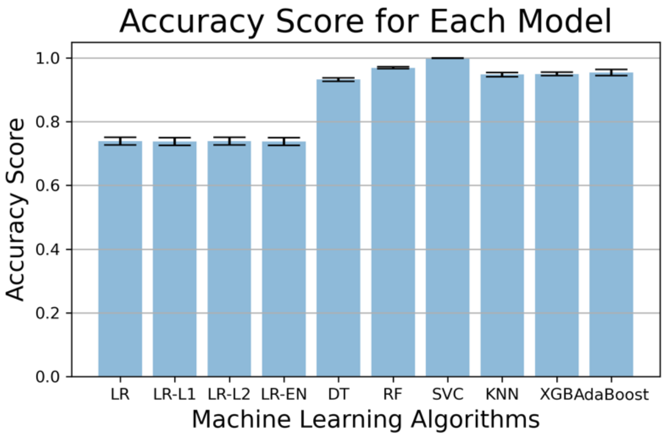

Figure 6. Accuracy score for each model

Figure 6. Accuracy Scores.

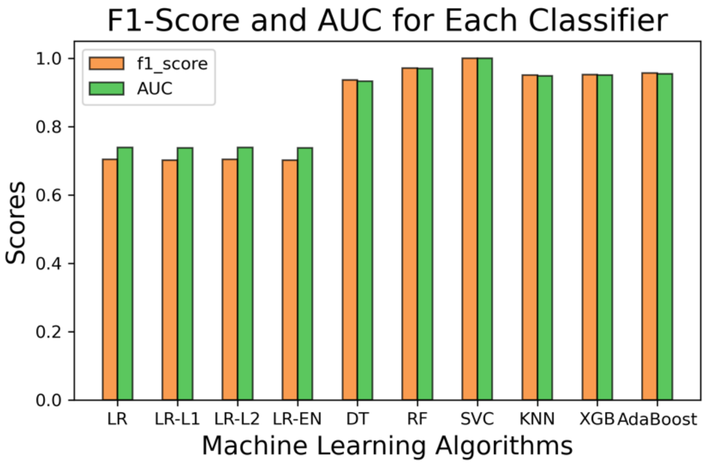

Figure 7. F1 score and AUC for each model

Figure 7. F1 scores and AUC.

As our dataset is very balanced, the two classes have similar distributions, we can choose to use accuracy score. Furthermore, we pay more attention to the class 'yes' (the clients who subscribe a term deposit) which means true positive and true negative are more important. So, accuracy is a better matrix in this situation.

3.3 Final Model Selection

From the above plots and corresponding evaluation matrices, we can see that SVC has the highest accuracy. So, in the comparisons and interpretations, we will use this SVC model with best parameters. We also use Random Forest and Logistic Regression as RFC has a comparatively good results and latter one is easy to interpret.

4. Results

4.1 Evaluation of models

We use accuracy score, F1 score and AUC to evaluate the performances of all values, the mean scores and standard deviations are represented in the following table 2.

Model	ACC_mean	ACC_std	F1-Score	AUC
LR	0.738896	0.011932	0.703401	0.738388
LR (L1)	0.737943	0.011716	0.701829	0.737418
LR (L2)	0.738965	0.011829	0.703408	0.738464
LR (EN)	0.737602	0.011833	0.701369	0.737076
DT	0.932357	0.005399	0.935646	0.932294
RF	0.969278	0.002800	0.970638	0.970009
SVC	0.999387	0.000566	0.999378	0.999395
KNN	0.947888	0.006648	0.950256	0.948115
XGBoost	0.950000	0.005553	0.951830	0.950180
AdaBoost	0.954360	0.954360	0.009643	0.956312

Table 2. Evaluation scores for each model

Table 2. Summarize different scores and uncertainties for all machine learning models.

As Support Vector Classification has the highest accuracy score with lowest standard deviation, we use SVC again in 10 different random states using basic split and kfold validation to compare the results with the baseline model (predict 1 all the time). The results can be seen in table 3.

Model	ACC_mean	ACC_std	F1_mean	F1_std
Baseline	0.498025	0.011610	0.664828	0.010362
SVC	0.999387	0.000566	0.999378	0.000566

Table 3. Comparisons between baseline model and SCV model

Table 3. Accuracy and F1 scores for SVC and baseline models.

For baseline model, F1 score is 0.665 with 0.01 standard deviation. For SVC model, F1 score can achieve 0.999 with standard deviation 0.001. The trained SVC models achieves an F1 score that is 33 standard deviations above baseline. Similarly, the baseline model's F1 score was 334 standard deviations below the average of the trained SVC model.

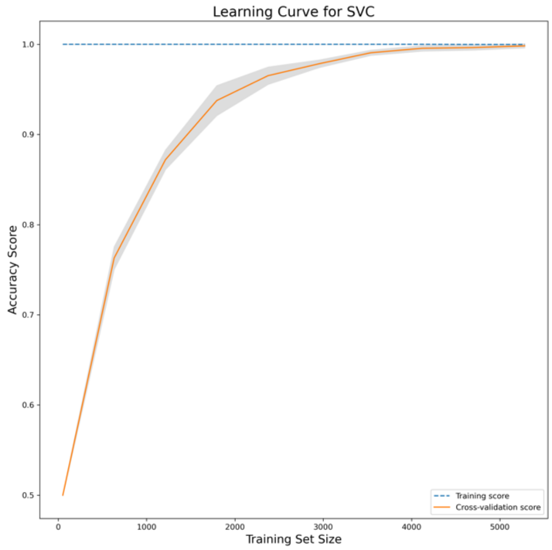

Figure 8a. learning curve for SVC

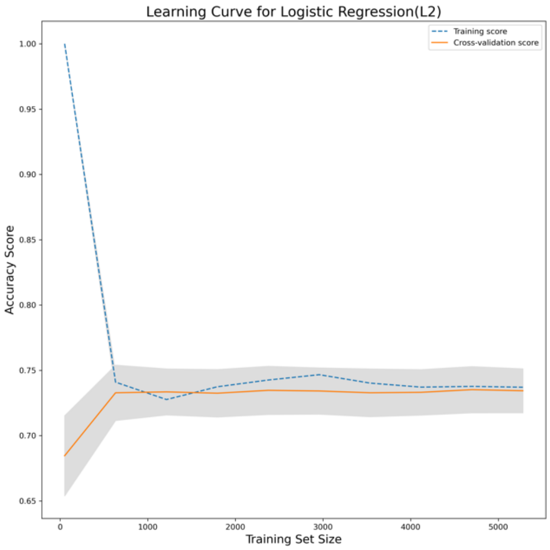

Figure 8b. learning curve for LR(L2)

Figure 8. From figure 7a and figure 7b, we can see the learning curves to simulate the training process of SVC and LR(L2) model.

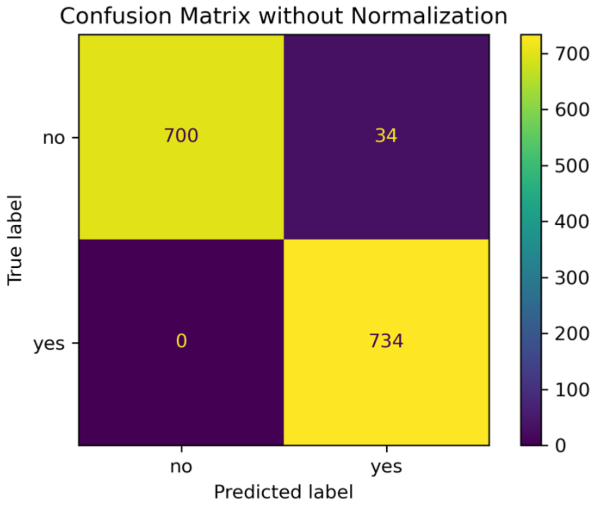

Figure 9. Confusion matrix of SCV model

Figure 9. We also can see the confusion matrix for SVC model in figure 9 From the confusion matrix, we can find that the model can predict better on class “yes”.

4.2 Interpretation of Results

Global Feature importance for the model is calculated using the coefficients of the model, the permutation importance and the SHAP values. Below are the results of these evaluations:

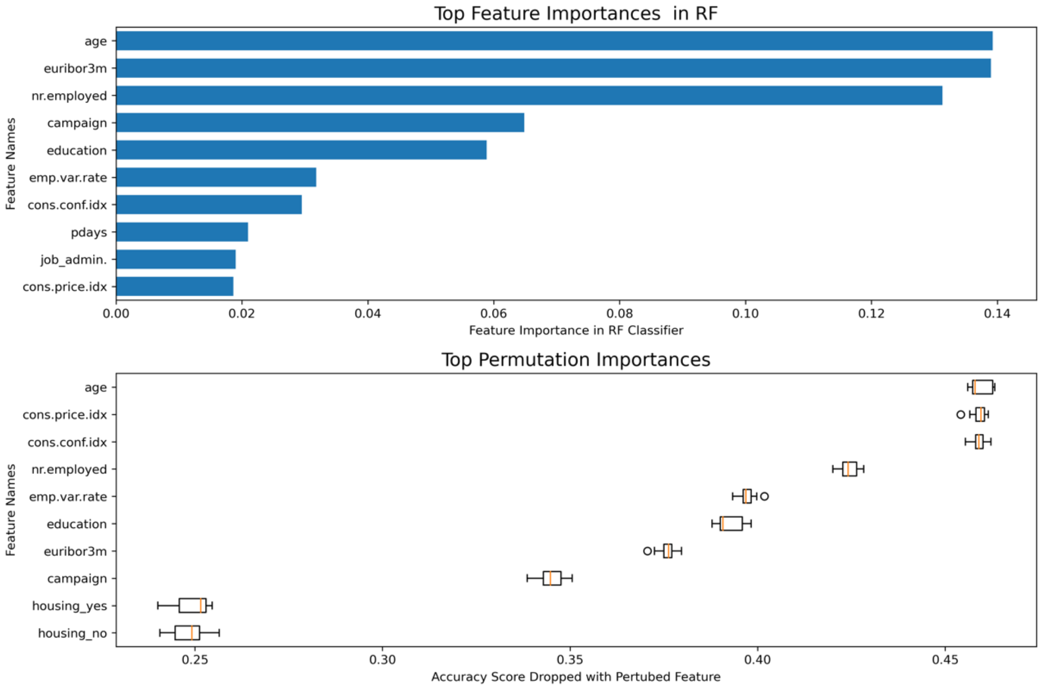

Figure 10. Global feature importance using RF model and permutation method.

Figure 10. Global feature importance.

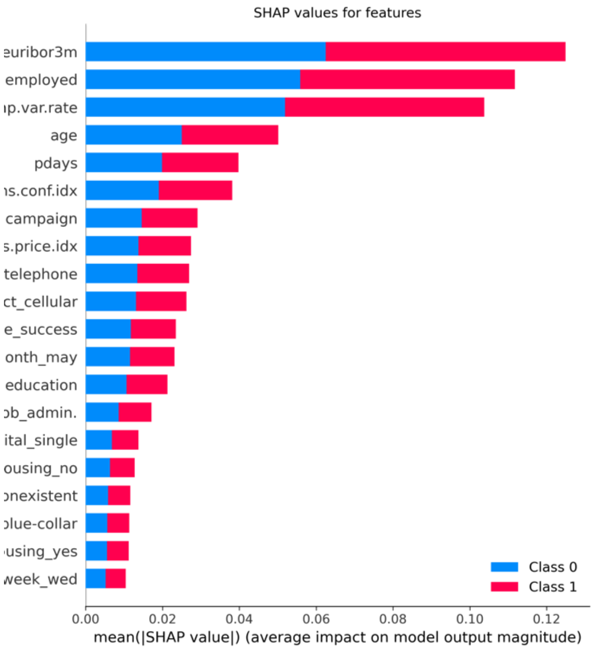

Figure 11. Global feature importance using SHAP values.

Figure 11. Global feature importance (SHAP)

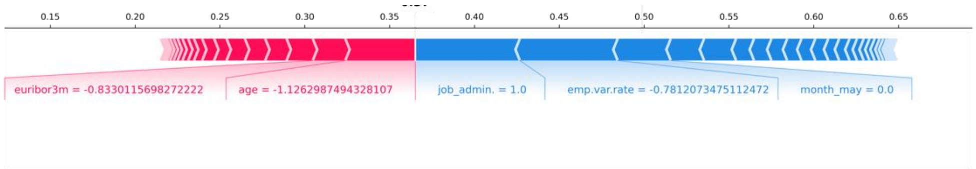

Figure 12. Local importance using SHAP.

Figure 12. Local feature importance (SHAP)

We use RF model to get the feature importance as SVC model (use not linear kernel) cannot have the importance. We also use permutation importance. Plots for global feature importance can demonstrate that age, nr.employed, emp.var.rate, cons.conf.idx and education are somewhat more important than other features. And we can know that default_yes feature is the least important feature. We can summarize that the target clients must be in a specific age range with high employment rate, education level and consumption confidence. Also, it is not important that the client has a default credit.

5. Outlook

Through this project, we compare several machine learning models to predict how likely clients will subscribe to a bank term deposit. The best model is support vector machine with optimized hyperparameters. However, the SVC model is somewhat difficult to interpret. We use resampled methods to deal with imbalance, but we can also try using “weight” parameter in each model and compare the results. For the feature correlations, we know there are some features with high correlation, we can improve the model with manual feature selections then the model can be interpreted in a clear way. So generally, we should focus on targeting customers with high cons.price.idx (consumer price index) and euribor3m (3-month indicator for paying off loans) as they are high importance features for the model and business. The customer's age affects campaign outcome as well. Future campaigns should concentrate on customers from age categories below 30 years old and above 50 years old. Therefore, we save time and money knowing the characteristics of clients we should market to and that will lead to increased growth and revenue.

Reference:

[1] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems (2014), doi:10.1016/j.dss.2014.03.001.

[2] Ling, X. and Li, C., 1998. “Data Mining for Direct Marketing: Problems and Solutions”. In Proceedings of the 4th KDD conference, AAAI Press, 73-79.

Github Repository: https://github.com/gfreya/DATA1030-Bank-Marketing-Analysis