Predict the results of bank telemarketing

Data Source: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

MACHINE LEARNING CASE STUDY:
A data-driven approach to predict the results of bank telemarketing

The goal is to predict whether a customer will subscribe to a term deposit or not, given customer relationship data. This is a binary classification problem type of machine learning.

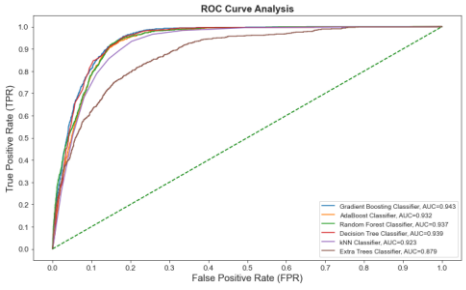
In this project, we are analyzing the marketing data of a Portuguese banking institution which is based on phone calls. Potential clients are contacted through phone calls to access if the bank term deposit would be subscribed or not. Often, more than one contact to the same client was required, in order to assess if the product (bank term deposit) would be (or not) subscribed.

The dataset has 20 predictors excluding the target variable which includes numerical and categorical data. Data was analyzed and various methods such as missing data imputation, scaling (one-hot encoding and standard scaling using a pipeline) and verification of data imbalance were implemented to prepare the data for modelling purposes.

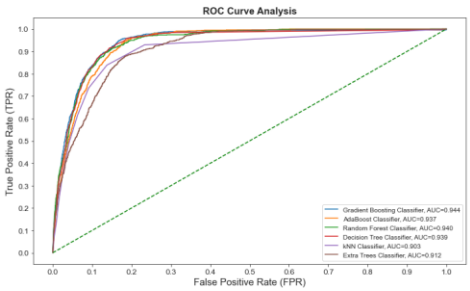
Balance checking on the original dataset was performed at the beginning of our research. The original dataset has 41,188 records. As the bank marketing dataset is imbalanced (target column: 11.27% "yes", 88.73% "no"), the training dataset had to be resampled (under-sampled and up-sampled) and the models were tested using both re-sampling methods. After under-sampling the majority class, the training dataset had 7424 samples in total: 3712 subscribing bank term deposit, 3712 not subscribing bank term deposit. After over-sampling the minority class, the training dataset had 58476 samples in total: 29238 subscribing bank term deposit, 29238 not subscribing bank term deposit.

One of the testing methods used was **hyper-parameter tuning of various models** (*Decision Tree, Random Forest, AdaBoost, Gradient Boosting Classifier, Extra Trees Classifier, kNN*) using **GridSearchCV**. An EstimatorSelectionHelper class was created so that two dictionaries (models and parameters) could be passed to the class. After each model was fitted, the **score_summary**() would provide us with a data frame that shows each model's instance with the parameters used.

Tests were performed using various models on two sets of data, ***under-sampled and original pre-processed*** data.



Under-sampled data



Original pre-processed data

When we compare results from the under-sampled and original pre-processed training set, we can see that roc_auc score is almost the same. Both training sets give a **score of 94% for the best tuned model which is Gradient Boosting Classifier**.

Further tuning for 4 ensemble models were performed using RandomizedSearchCV to analyze if performance scores could be further improved. Results of both under-sampling and over-sampling majority classes can be seen below:

Results of under-sampling majority class

ensemble model	5-fold cv mean roc_auc	5-fold cv std_test_score	5-fold cv mean_fit_time	one-time roc_auc	one-time accuracy	one-time precision	one-time fit_time	comments
Random Forest	0.9433	0.0068	1.2024	0.9487	0.8568	0.4365	0.3709	had 2nd best 5-fold cv mean roc_auc score; the best when predicting using original test dataset
Extra Trees	0.9431	0.0033	0.7934	0.9439	0.8479	0.4205	0.1996	had comparably good roc_auc scores; fit faster than the other 3 ensemble models
AdaBoost	0.9359	0.0065	1.1173	0.9392	0.8670	0.4525	0.6478	had the lowest roc_auc scores
Gradient Boosted Trees	0.9454	0.0073	2.0770	0.9473	0.8603	0.4423	1.2259	had the highest 5-fold cv mean roc_auc score; took the longest time to fit; the 2nd best when predicting using original test dataset

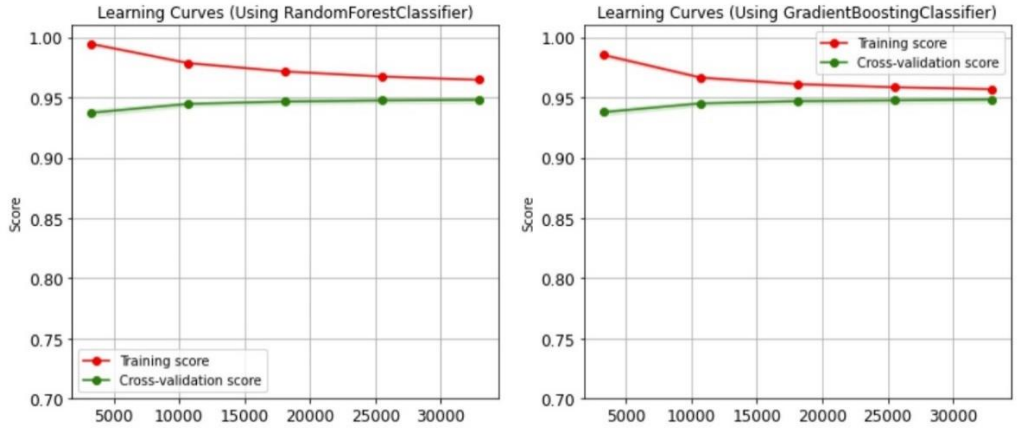
Results of over-sampling majority class

ensemble model	5-fold cv mean roc_auc	5-fold cv std_test_score	5-fold cv mean_fit_time	one-time roc_auc	one-time accuracy	one-time precision	one-time fit_time	comments
Random Forest	0.99986	0.00007	4.32185	0.94119	0.91175	0.61922	0.85748	had the 2nd best 5-fold cv mean roc_auc score; also generated the 2nd best roc_auc score, the best accuracy and precision score when predicting using original test dataset
Extra Trees	0.99996	0.00008	3.01230	0.90998	0.90240	0.61698	0.69096	had the best 5-fold cv mean roc_auc score; fit faster than the other 3 ensemble models; had lower scores when predicting using original test dataset
AdaBoost	0.94274	0.00113	5.78664	0.94198	0.87145	0.46220	4.10827	had the lowest 5-fold cv mean roc_auc scores, but the best roc_auc score when predicting using original test dataset
Gradient Boosted Trees	0.99188	0.00027	39.95699	0.88976	0.90325	0.56583	21.85838	had comparably good 5-fold cv mean roc_auc score; took the longest time to fit; had the lowest roc_auc score when predicting using original test dataset

When considering both roc_auc score and the precision score, using "over-sampling for the minority class" during resampling, then tuning a RandomForestClassifier with RandomizedSearchCV, we were able to establish an ensemble model with the best overall performance: prediction roc_auc score **0.9412**, and with the best precision score **0.6192**.

Whereas if we considered only the roc_auc score, using "under-sampling for the majority class" during resampling, then tuning a GradientBoostingClassifier or RandomForestClassifier using RandomizedSearchCV would yield the best ensemble model, with prediction roc_auc scores up to **0.9487** and hence, used as our sampling method for training purposes.

Learning curves for the under-sampled training dataset plus testing dataset



Another prediction model, "***TensorFlow***" of neural network was tested for two different sets of parameters. The best AUC for the tensor flow models of training is ~ 0.959 which is a bit higher and works much faster than the other previously used methods (Gradient Boosting Classifier AUC ~0.943). When used on the test set, prediction of this model gives only us an AUC score of 0.87 which is lower than what we saw for Gradient Boosting Classifier.

Using various resampling methods and tuning methods, prediction performance scores of all the ensemble models/classifiers were similar in terms of roc_auc scores. Over-sampling could lead to the highest training roc_auc scores close to 1(>0.9999), but tended to cause overfitting.