Data Glacier

Your Deep Learning Partner

Model Building and Selection

Team Muadh

Muadh Faizan
muadhfaizan@gmail.com
Data Science
13 August 2021

Agenda

Executive Summary

Problem Statement

Approach

Evaluation Criteria

Model Building & Comparison

Best Model

Conclusion & Recommendations

Executive Summary

This presentation will illustrate the different methods used to build classification models in order to predict if a customer of a Portuguese bank will buy a product or not. The programming language used to carry out this project was Python, using Jupyter notebook.

The business problem will first be described, followed by the different models created and their metrics. Then a comparison table will be shown with all models and their metrics, and the best model will be identified. Finally, some recommendations to the bank will be presented.

Problem Description

The project chosen to be done is the Bank marketing project, which is regarding a term deposit product of a Portuguese bank. The bank wants to sell a term deposit product to its customers but want to know which of their customers will be more likely to buy it, based on the past interactions with the bank. A classification problem is at hand, where the dataset contains information of several customers such as their age, gender, and whether the customer had bought the product or not.

The aim of the project is to analyse the dataset and come up with a classification model which would be able to predict if a customer would buy the product or not.

Approach

The different modelling techniques attempted are as follows:

- Logistic Regression
- Decision Tree
- Random Forest
- Bagging Classifier
- Gradient Boosting Classifier
- XG Boost
- Stacking Classifier

The hyperparameters of all ensemble models were also tuned in order to find better models.

Evaluation Criteria

The incorrect predictions are the following:

- Customer is predicted to buy but doesn't buy (Type I error).
- Customer is predicted not to buy but does buys (Type II error).

The more costly error is the type II error, as the bank would not market the product to this customer and fail to make a sale because the customer would have in reality bought the product.

Type I errors will make the bank focus on customers that would not buy the product, so it is not too much of a cost except for the time and effort put into these customers.

Model Building

- The first model was a Logistic Regression model. The dataset was checked for multicollinearity, and then the model was built. The insignificant features were dropped, and then the metrics were generated.
- The default threshold is at 0.5, which is usually not the optimal one. Using the recall-precision curve, a better threshold was identified at 0.12 and an optimized model was created.

Model Building

- The next models created were Decision Tree, Random Forest, Bagging Classifier, Gradient Boosting Classifier and XG Boost Classifier.
- The recalls of these models were generally low and most of the models had overfitting, so the next step was hyperparameter tuning.
- For Bagging, Gradient Boosting and XG Boost Classifiers, I tuned the hyperparameters and sought the best models using RandomizedSearchCV.
- Finally, I created a Stacking Classifier model using the previous models as estimators.

Model Comparison

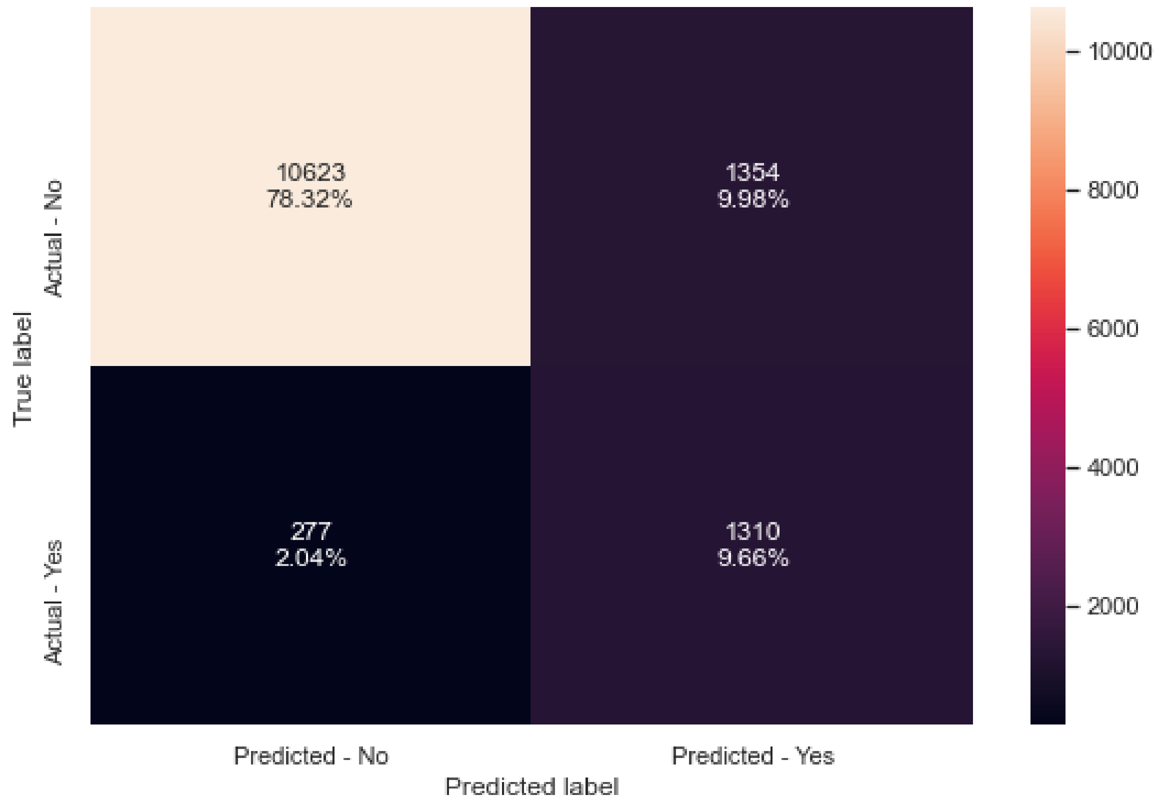
Model	Train Accuracy	Test Accuracy	Train Recall	Test Recall	Train Precision	Test Precision
Stacking Classifier	0.906	0.864	0.982	0.839	0.555	0.455
Tuned XGBoost	0.891	0.880	0.886	0.825	0.521	0.492
Optimised Logistic Regression	0.845	0.844	0.819	0.817	0.417	0.414
Decision Tree	1.000	0.873	1.000	0.502	1.000	0.460
XGBoost Classifier	0.958	0.907	0.731	0.491	0.893	0.633
Tuned Gradient Boost	0.917	0.907	0.486	0.440	0.710	0.655
Gradient Boost	0.911	0.907	0.436	0.416	0.692	0.667
Bagging Classifier	0.992	0.902	0.932	0.398	0.995	0.629
Random Forest	1.000	0.904	1.000	0.381	1.000	0.656
Tuned Bagging Classifier	0.998	0.901	0.981	0.303	1.000	0.672

Best Model

Based on the test recalls, the highest and best three are the Stacking Classifier and the Tuned XG Boost. However, the Stacking Classifier has some clear overfitting, and the recall score of other two is only slightly lower and precision is similar. The Tuned XG Boost has a better precision, so I will choose the Tuned XG Boost as the best model for the bank to use, as it has a high recall and no overfitting, so it is likely to perform better in production.

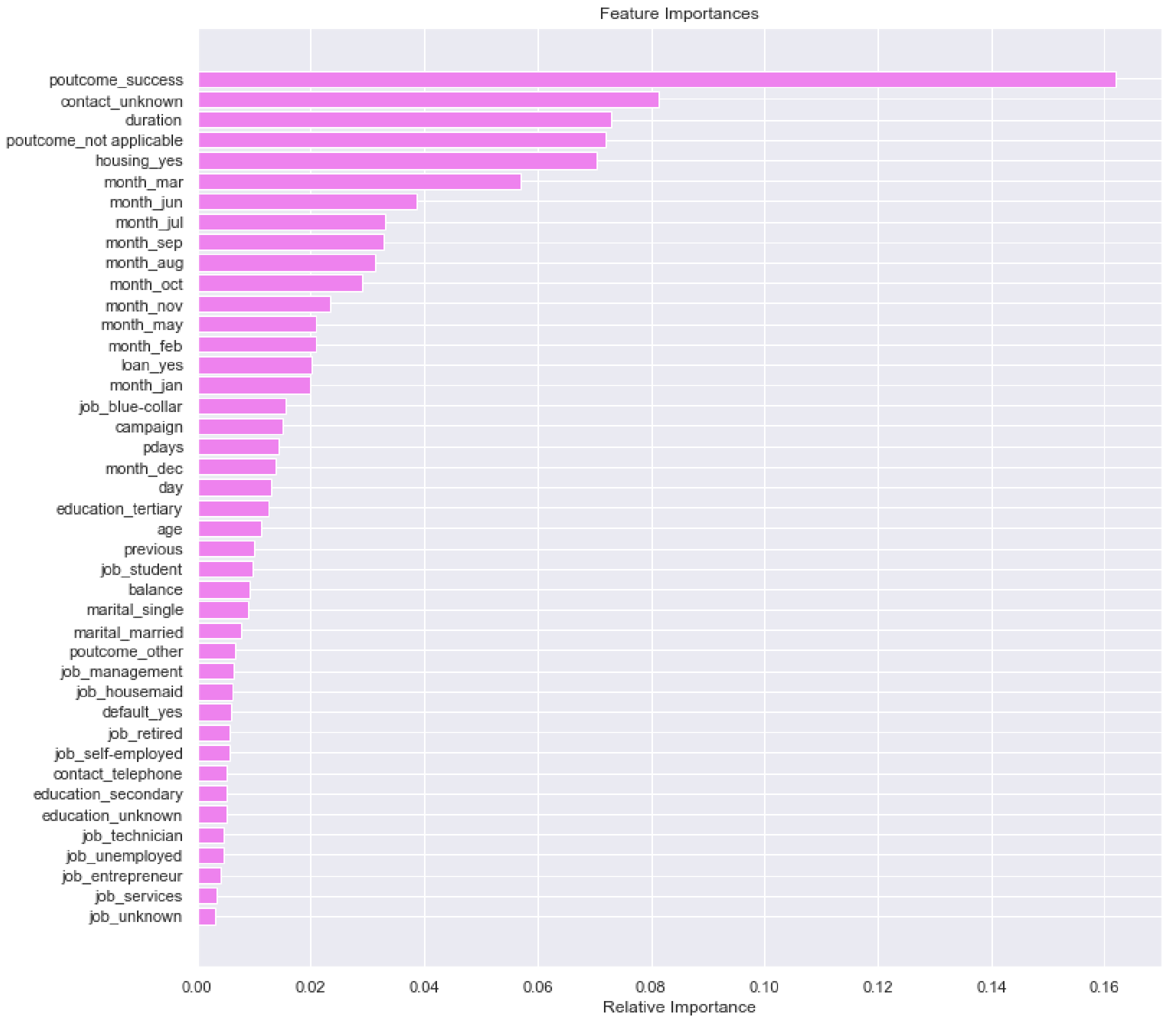
Best Model – Confusion Matrix

Out of 1587 customers who bought the product, the model correctly identified 1310 customers, which is excellent. It also incorrectly identified 1354 other customers as potential buyers. This is not ideal but considering the recall score and the number of correct buyers identified, this is totally fine.



Best Model – Feature Importance

- As expected, the model considers the outcome of previous campaign as the most important. This was clear in the EDA, where more than 50% of previous buyers bought the term deposit as well. The only issue would be the lack of data since very few customers were contacted for a previous campaign.
- The second most important feature is contact_unknown, followed by duration which I am not too sure what the reason is.



Recommendations

Some recommendations to the bank are:

- Focus on customers that previously bought a product from a campaign.
- Keep in mind that there will be a lot of customers recommended by the model who will not buy, but the model will get as much potential buyers as possible.
- The model can be improved by getting better data, such as average call duration instead of the last one only, how long a customer has been with the bank, etc.
- It would also be useful to segment the customers into groups, and then products can be better tailored and marketed to the customers.

Thank You