Task for Junior Quantitative Analyst in Credit Risk Model Validation

Objective:
Analyze the Portuguese Bank Marketing dataset (Moro et al., 2011) and build a predictive model for client subscription to a term deposit (“y”) based on demographic and campaign features.

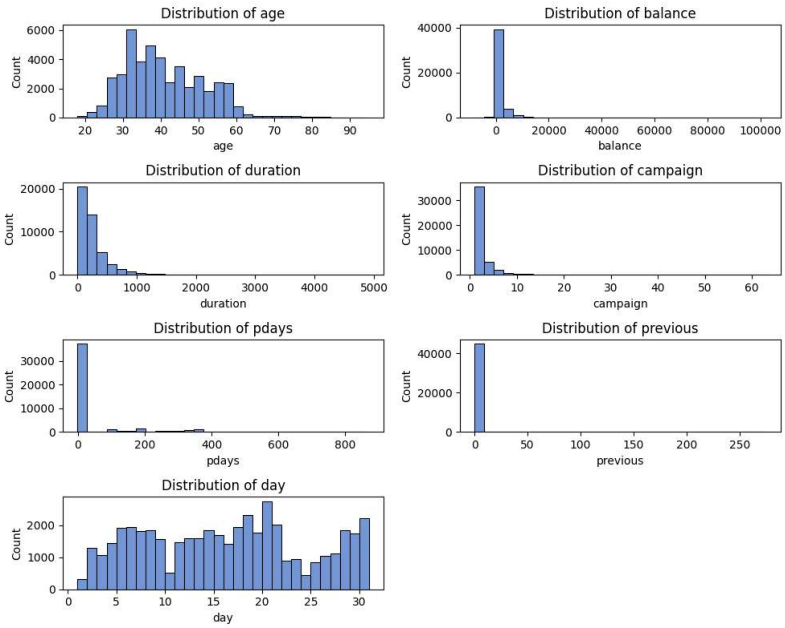
## 1. Data Overview

The dataset contains 45,211 records, 16 input features and target “y”.
No missing or duplicate values were detected.
Class imbalance was detected, where only 11.7 % of the data corresponds to target “y”.
Class imbalance can be problematic because models may become biased toward the majority class, leading to poor detection and prediction accuracy for the minority class

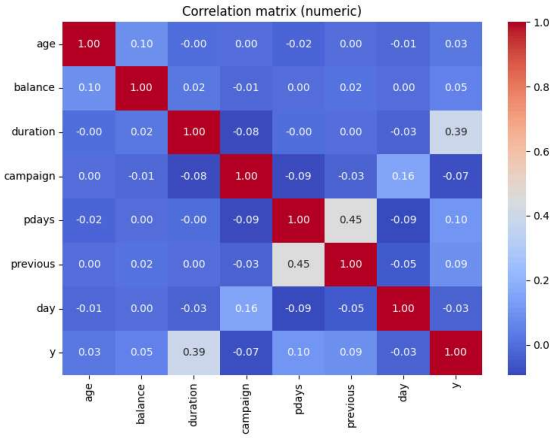
## 2. Exploratory Data Analysis (EDA)

### 2.1. Numeric features

Histograms were plotted to better understand the distribution of the numeric variables



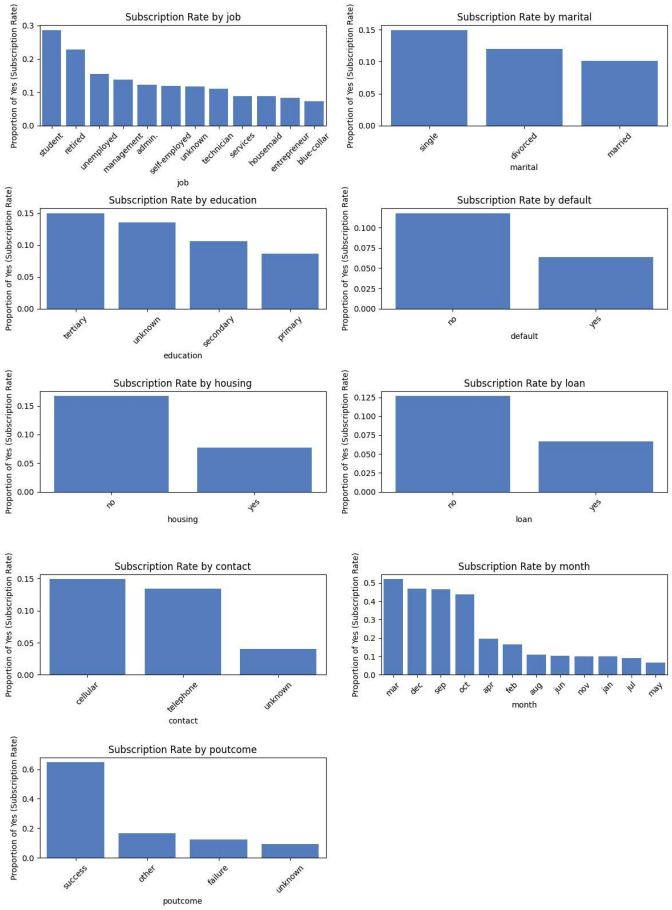

Outliers were present in almost all variables, however, they did not seem like anomalies or mistakes, rather just the nature of the dataset. No corrections to the dataset were made at this point.
Correlation matrix was built to see the correlation between variables and outcome “y”



The correlation matrix indicated that there was a mild correlation between outcome “y” and duration of the last contact. Variables previous and pdays are also correlated, however, the correlation coefficient is not high enough, so multicollinearity should not become a real issue

## 2.2. Categorical Variables

Categories of variables are not distributed equally, there are some categories that appear more often than others. A bar chart was plotted to display the proportion of outcome ‘y’ per category




An educated guess could be made that job, contact type, last contact month and last promotion outcome have an effect on outcome ‘y’. However, other categories also display differences and at this point it is difficult to say if they are useful or not. Low p values for these variables $p < 0.001$ indicate that the categorical variable and the target “y” are not independent, suggesting that these features have a statistically significant relationship with the outcome. However, it doesn’t measure how strong or useful that relationship is for prediction.

It was decided to leave all numeric and categorical variables in for now and use an RFECV (Recursive feature elimination with cross-validation) algorithm later to reduce redundant or not effective variables

# 3. Feature Processing

3.1. Data was split into training and testing parts with a ratio of 2:1

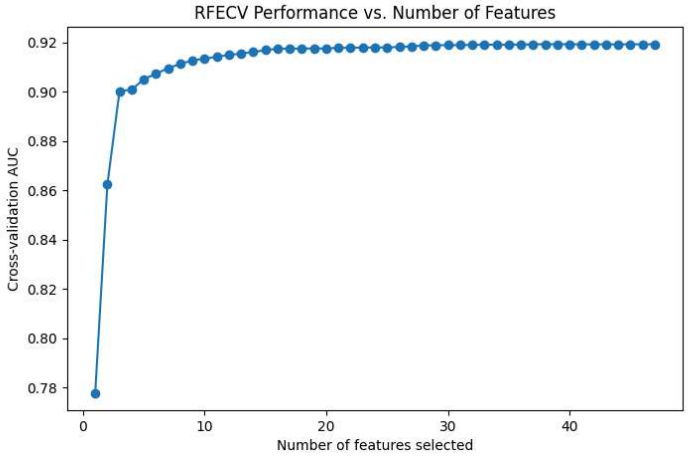
3.2. For Numeric feature processing standardscaler() function was used to standardize the values of numeric features. It transforms data so that the mean becomes 0 and the standard deviation becomes 1. A feature measured in large units (like “balance”) can dominate a small-scale feature (like “pdays”), simply because of scale differences.
Edit: a duplicate flow without standardscaler() was tested: results were very similar, with precision for “y” changing from 0.42 to 0.43 and recall remaining the same. However, the flow selected more features (50) and needed more iterations for logistic regression

3.3. For categorical features, oneHotEncoder() was chosen to Encodes categorical features as a one-hot numeric array as logistic regression models cannot use string values. Another alternative could have been get_dummies(), however, oneHotencoder() handles unseen categories better.

3.4. To prevent the model from being biased towards the majority due to imbalance in the dataset (11.7 % “y”), SMOTE (Synthetic Minority Over-sampling Technique) was used on the training set to produce synthetic data for category y = 1, which was underrepresented.

4. RFECV

RFECV (Recursive feature elimination with cross-validation) was used to pick relevant features. Unlike RFE(), RFECV() does not require to specify how many features in total I would like, and determines the optimal number itself by estimating cross-validation AUC. 42 out of 51 features were kept by the RFECV




5. Modeling and validation

5.1. Logistic Regression

max_iter=2000, solver='liblinear' was used to model logistic regression.

5.2. Classification report and confusion matrix

While the accuracy for the default threshold of 0.5 seemed fine - 0.85, but the model had many false positives. Precision was very low - 0.42.

A threshold of 0.7 was adjusted for the classification report, to hopefully reduce the number of false positives. But this also lowered recall significantly. The confusion matrix illustrates these findings.

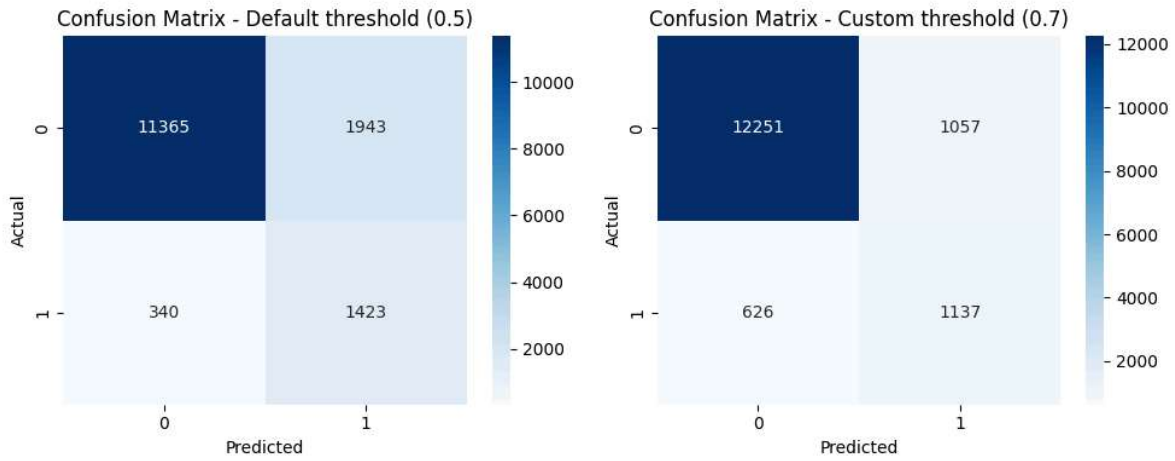
The model does a decent job at finding customers who might subscribe, but it still makes too many incorrect positive predictions. This happens mostly because the data is unbalanced, with far fewer positive cases than negative ones.

Classification report (default threshold = 0.5):

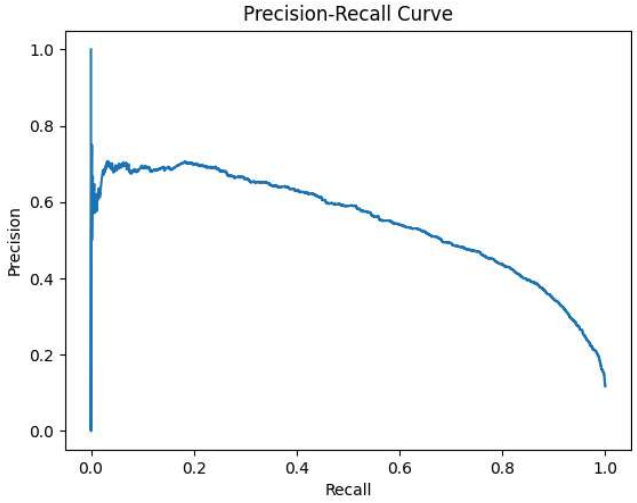
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.85 | 0.91 | 13308 |
| 1 | 0.42 | 0.81 | 0.55 | 1763 |
| accuracy | | | 0.85 | 15071 |

Classification report (custom threshold = 0.7):

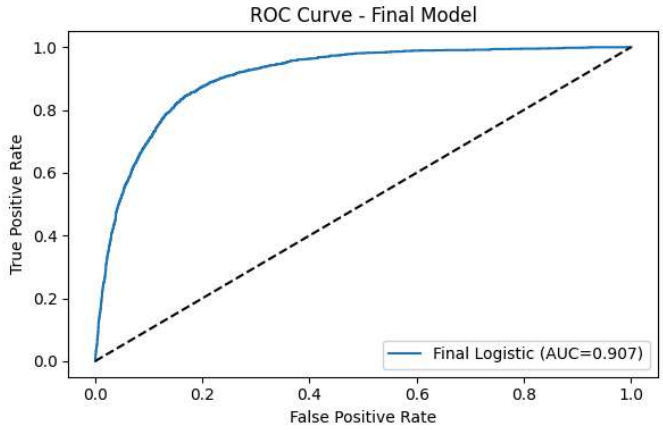
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.95 | 0.92 | 0.94 | 13308 |
| 1 | 0.52 | 0.64 | 0.57 | 1763 |
| accuracy | | | 0.89 | 15071 |




Precision-recall curve shows that there is no sweet spot for obtaining precision above ~0.7, and even then, recall drops significantly




Precision at top 10% (top 1507 samples) would be 0.595
Lift at top 10%: 5.08x better than random




The model can somewhat tell the difference between customers who will subscribe and those who won't, but it's not very strong. It's better than random guessing, but there's still plenty of room for improvement.

## Conclusion:

The model reasonably distinguishes deposit subscribers but struggles with precision due to class imbalance.

## Other considerations:

To improve results, it would be useful to try other models such as Random Forest since they can handle non-linear relationships and are often better at dealing with imbalanced data. Another option is to adjust class weights in the logistic regression model so that errors on the minority class (“y” = 1) are penalized more, which could help the model focus more on correctly identifying potential subscribers.