Patricio Massaro - BairesDev Coding Challenge

I. Introduction and Objective

This coding challenge consists in predicting which leads from an email marketing application have the highest chances of increasing a metric called Business value, based on leads from the past.

This challenge is implemented in python, using scripts (.py) for processing the files and a Jupyter Notebook (.ipynb) with all the data processing of the dataset and the implementation of the prediction model. The predictions are attached in csv format, the last column shows whether the lead has high chances of increasing Business Value or not. The steps of this challenge are explained in the following sections.

II. Data-engineering

A. Files processing

The Data consisted in two separate datasets. The first dataset consisted in leads that the company is planning to contact, with information about the client size, location, etc. It consisted in 10 files.

The second dataset has information about contacted leads in the past, there are 10 files with negative responses (no replies or not interested) and 1 file with positive responses. All the positive and a couple of the negative responses have a score column, which represented the business value assigned by the sales team, based on the profile of the contact and the potential of the opportunity that the potential customer requested.

After solving some issues in a couple of lines in the csvs, the files were processed into two single datasets, one for training and the other one for prediction. As the response of the leads in the training datasets is not present in the columns, it was added using the title of the files. The details of the implementations is in *File_processing.py*

B. Target variable construction

The objective of the challenge is to predict which leads have high chances of **increasing** the business value if contacted. Taking this information into account, the target variable is whether the lead increased the business value (score column).

The calculation of the score is not clear in the challenge, but using the information available, a model of it could be:

$$S = D \times O \tag{1}$$

Where S is the score, D is a given value for the decision making capabilities of the lead and O is the opportunity that the lead requested. The multiplication in the model reflexes that both terms must be present in order to have a score value greater than 0. Some aspects and assumptions to be taken into account are:

- If the lead never responded, O will be 0, and the score will be 0.
- If the lead answers and requests services for the first time, the score will rise from 0 to a value.
- If the same lead answers again, but the opportunity remains the same, the answer will be positive but the score should not increase
- If the same lead answers again, and a bigger opportunity arises, the score will rise.

Using this logic, the 'increased_value_flag' is created, it will be '1' if the contact induced an increment in the business value and '0' otherwise.

C. Independent variables cleaning

The datasets contain characteristics of the lead and the date of the contact, the details of the processing and cleaning of each attribute is stated below:

The *campaign_bulk_date* is a string column with the date of the marketing campaign, using multiple formats. To normalize them, they were casted to datetime using *pd.to_datetime* and then casted to string using *dt.strftime*. Unfortunately, all the positive responses had null in this fields, limiting it's further uses.

The *country* column gives information about the country of the lead, this column was not modified, all values were non-nulls and fixed with no spelling mistakes.

The *kw_industry* column has the type of Industry of the lead, this column has roughly 140 categories, a map function with a dictionary was implemented reduce this number. The type of industry was mapped to the Global Industry Classification Standard from S&P.

The *kw_size* column refers to the size of the company of the lead. This column has a few categories and no-nulls, no modifications were made.

The *score* column shows the score given by BairesDev Analysts, for non-responsive leads it is nan. All nan values were transformed into 0.

The *state* column shows the state of the lead, the problem with this column is nan when the country is not USA, Canada or UK. These nan values were converted to 'Unknown'.

The *title* columns shows the position of the contacted lead, this column has a huge amount of different categories. It is mandatory to reduce them to make the column adequate to be used by a ML model. Three categories were created based on the decision making capabilities. Upper management includes C levels, owners and founders. Middle management includes managers, team leads, coordinators, etc, and lower management includes engineers, specialists, analysts, etc. All non-matching titles were converted to 'Others'. Nulls were converted to 'Not defined'

The *Increased_value_flag* column is the target variable, it is very imbalanced, having roughly 97% zeros (no Business value increased) and 3% ones (Business value increased)

III. Prediction Model

The prediction model will determine which leads Baires-Dev should contact first. In this section the steps involved in will be explained.

A. Performance Metric

Taking into account the problem to solve and the imbalance in the target variable, *Recall* was selected as the performance metric. This score shows the proportion of the real positives that the algorithm predicted. The imbalance of the target variable shows that using accuracy we could create a predictor that predicts always zero and we would have circa 97% accuracy (and 0% recall), leading to wrong conclusions.

B. Train, validation and test

At first, some columns of the dataset were dropped in order to generate the model :

- The Date column was dropped because it is not available in the leads to predict. In addition, no variable could be generated using the date because the positive responses have no date values, only nan.
- The City column was dropped because of very high dimensionality, spelling errors and null values.
- The idlead column must be dropped.
- The Response column was dropped because it is not available in the leads to predict.

The test dataset is the one with the leads to be contacted. The dataset with leads contacted in the past will be subdivided in Training and validation.

The model will be trained using training data (70% of the original dataset) and will be tested using the validation dataset (30% of the original dataset). The train and validation division is made taking into account the balance of the target classes.

This way, the prediction model can be tested several times with different combinations of algorithms and hyperparameters to maximize the performance metric. After this optimization process, the model can make a better prediction of the test data.

C. Model

The algorithm selected for the task is a random forest classifier, due to it's versatility in classification problems with high dimensionality and robustness to outliers.

A random search for hyperparameters using k-folds cross validation was intended to be implemented, but due to the lack of computational resources and time restrictions, i decided to use the default parameters.

Using the training data, the model was created. Then, the predictions for the validation data was made. the results are shown in the table below:

Set	Recall Score
Train Set	0.95
Validation Set	0.73

Table I
Recall score for training and validation sets

This results could be improved by adjusting the hyperparameter of the model, which seems like it's overfitting to the training data a little bit based on the difference between scores. However, the score is acceptable for a first approach.

IV. Future improvements

- The obtained score in the validation set is acceptable. However, a hyperparameter tuning should be made to maximize the metric. In addition, a k-folds cross validation should be implemented.
- The date of the campaign is a key aspect in the understanding of the model, unfortunately, it is not present in the positive responses. For example, a time series of the responses for a given client could be made. Another possible column is the month of the positive response (maybe the companies tend to accept business offers in may, when they are planning the budget year). Another possible column could be how much time passed from the first contact to the first positive response.
- Financial information about the companies could be attached, a firm with budget deficit is probable more reluctant to hire new services.
- The state of the leads should be complete, even in countries outside of the US, Canada and UK.
- Fixed categories should be implemented in the City, industry and title category, to avoid high dimensionality.