Technical Report : Direct Marketing Optimization
Kannan Ravikumar Girija

1. Introduction

This task is maximizing revenue from direct marketing campaigns using the provided dummy data, simulating a real world marketing scenario. The objective is to optimize resource allocation for marketing efforts by analyzing client data. The datasets include social-demographic information, product holdings, financial transactions, and sales/revenue data for 60% of clients. With a contact limit of 15% of clients and one marketing offer per client, the goal is to strategically target the most promising clients to maximize revenue through tailored offers based on their profiles and behavior.

The process is comprised of three key steps: 1) Data Processing, 2) Model Training, and 3) Production Inference. The following sections will provide a detailed explanation of each step, outlining the approach and methodology used throughout the process.

2. Data Processing

The data is initially read from multiple Excel sheets and merged into a single Pandas DataFrame using the client ID as the key. Missing values in the "count" and "revenue" columns are filled with zeros, assuming that a missing value indicates no balance or transaction. For the "Sex" column, missing values are imputed with the mode ("M") and then converted to binary values (1 for male, 0 for female). Rows corresponding to clients with no transaction data are excluded from further analysis.

Next, the correlation between features is calculated, revealing that the "Current Account (XX_CA)" features exhibit a correlation greater than 0.9 with the corresponding accounts. These features are dropped from the analysis to avoid redundancy.

To create the training and test datasets, the "Sales_Revenues" sheet is merged with the other data using the client ID. Ten percent of the data is reserved for testing purposes. The data is then split into six distinct datasets for classification and regression tasks, focusing on CL (Consumer Loan), CC (Credit Card), and MF (Mutual Funds). It was observed that there is a data imbalance, which will be addressed during the model training phase. If time permitted, I would have explored additional techniques such as oversampling, undersampling, and SMOTE (Synthetic Minority Over-sampling Technique) to further mitigate the imbalance and improve model performance.

Feature standardization is applied using StandardScaler(), and the scaler metadata is saved for future use. To identify the most relevant features, Recursive Feature Elimination (RFE) is employed separately for each of the three models (CL, CC, MF). The selected features are stored for future data processing, while the irrelevant ones are discarded.

To reduce dimensionality, Principal Component Analysis (PCA) is performed, retaining components that capture approximately 90% of the total variance. The processed data is then saved for subsequent training and testing phases.

3. Model Training

For this task, I will train six models—three classification models and three regression models. The classification models will determine whether a client should be selected for the campaign for a specific product, while the regression models will predict the potential revenue generated from each client.

Due to time constraints, I will focus on using a single model: XGBoost. Specifically, I will utilize the XGBClassifier for classification tasks and the XGBRegressor for regression tasks. Had more time been available, I would have employed cross-validation techniques to identify the optimal model. However, in this case, I randomly selected parameter values to prevent overfitting and proceeded with training the models.

The trained models were then evaluated using various performance metrics. Below, I present the evaluation metrics for the classification models.

Classification Report:

	precision	recall	f1-score	support
0.0	0.77	0.70	0.74	71
1.0	0.32	0.40	0.36	25

Classification Report:

	precision	recall	f1-score	support
0.0	0.84	0.76	0.80	78
1.0	0.27	0.39	0.32	18

Classification Report:

	precision	recall	f1-score	support
0.0	0.76	0.75	0.76	64
1.0	0.52	0.53	0.52	32

It can be observed that the recall for the positive label is relatively low, which may result in missed detection of potential clients. To address this, techniques such as lowering the decision threshold could be applied to improve recall and capture more potential clients.

Similarly, the regression models were also evaluated. Below are the evaluation results for the test data.

```
Mean Squared Error: 33.1067
R²: 0.0008
Mean Squared Error: 176.4631
R²: -0.0246
```

```
Mean Squared Error: 1671.6245
R²: -0.0484
```

The evaluation results indicate that the model's performance is sub-optimal, as it struggles to generalize the data effectively. Due to time constraints, I was unable to dedicate additional time to improve the model further. Therefore, I will proceed with using the current model for inference purposes.

4. Production Inference

For production inference, all necessary metadata, including PCA components, standard scalars, selected features, and trained models, will be utilized. The entire dataset is loaded from an Excel file, and a data processing pipeline is created to handle tasks such as standardization, feature selection, and PCA.

Once the data is processed, it will be passed through the prediction pipeline for both classification and regression models. The classification model will identify clients eligible for targeting in the campaign, while the regression model will predict the

revenue each client is likely to generate. Only the predicted revenue for clients selected for the campaign will be considered. An additional column indicating the product category (CL, CC, or MF) will be added for later processing.
The resulting data frames for the three product categories will be concatenated and sorted by predicted revenue. From this, only the top entry for each product category—those with the highest revenue—will be retained, while the rest will be discarded. The top 100 entries for each product category will then be selected as the campaign targets. Using this data, the final revenue prediction will be made.

5. Script and Folder Details

This section outlines the scripts and associated folders for each step of the process:

- Data Processing: DataProcessing.ipynb
- Model Training: ModelsTraining.ipynb
- Production Inference: ProductionInference.ipynb

Additionally, the following folders contain the relevant data:

- data: Contains the input data for analysis.
- inference_metadata: Stores saved metadata generated during the data processing stage, which is later used during inference.
- model: Contains the trained models.
- processed_dataset_to_train: Holds the processed data used for training and testing.
- ***prediction_result***: Includes the prediction results, including client details and predicted revenue.