Digital Transformation in Banking Sector

Business Objective

Bank XYZ has a growing customer base where the majority of them are liability customers (depositors) vs. borrowers (asset customers). The bank is interested in expanding the borrowers base rapidly to bring in more business via loan interests.

A campaign that the bank ran in the last quarter showed an average single-digit conversion rate. In the last town hall, the marketing head mentioned that digital transformation being the core strength of the business strategy, how to devise effective campaigns with better target marketing to increase the conversion ratio to double-digit with same budget as per the last campaign.

As a data scientist, you are asked to develop a machine learning model to identify potential borrowers to support focused marketing.

Data Description

The dataset has 2 CSV files,

- Data1 - 5000 rows and 8 columns
- Data 2 - 5000 rows and 7 columns

The data consists of the following attributes:

1. ID: Customer ID
2. Age Customer's approximate age.
3. CustomerSince: Customer of the bank since. [unit is masked]
4. HighestSpend: Customer's highest spend so far in one transaction. [unit is masked]
5. ZipCode: Customer's zip code.
6. HiddenScore: A score associated to the customer which is masked by the bank as an IP.
7. MonthlyAverageSpend: Customer's monthly average spend so far. [unit is masked]
8. Level: A level associated to the customer which is masked by the bank as an IP.
9. Mortgage: Customer's mortgage. [unit is masked]
10. Security: Customer's security asset with the bank. [unit is masked]
11. FixedDepositAccount: Customer's fixed deposit account with the bank. [unit is masked]
12. InternetBanking: if the customer uses internet banking.

13. CreditCard: if the customer uses bank's credit card.
14. LoanOnCard: if the customer has a loan on credit card

Aim

Build a machine learning model to perform focused digital marketing by predicting the potential customers who will convert from liability customers to asset customers.

Tech stack

- Language - Python
- Libraries – numpy, pandas, matplotlib, seaborn, sklearn, pickle, imblearn

Approach

1. Importing the required libraries and reading the dataset.
    - Merging of the two datasets
    - Understanding the dataset
2. Exploratory Data Analysis (EDA) –
    - Data Visualization
3. Feature Engineering
    - Dropping of unwanted columns
    - Removal of null values
    - Checking for multi-collinearity and removal of highly correlated features
4. Model Building
    - Performing train test split
    - Logistic Regression Model
    - Weighted Logistic Regression Model
    - Naive Bayes Model
    - Support Vector Machine Model
    - Decision Tree Classifier
    - Random Forest Classifier
5. Model Validation
    - Accuracy score
    - Confusion matrix
    - Area Under Curve (AUC)
    - Recall score
    - Precision score
    - F1-score
6. Handling the unbalanced data using imblearn.
7. Hyperparameter Tuning (GridSearchCV)
    - For Support Vector Machine Model

8. Creating the final model and making predictions
9. Save the model with the highest accuracy in the form of a pickle file.

Modular code overview

```
input
  |_Data1.csv
  |_Data2.csv

src
  |_Engine.py
  |_ML_Pipeline
            |_model_evaluation.py
            |_grid_model.py
            |_train_model.py
            |_utils.py

lib
  |_Digital transformation in Banking sector.ipynb


output
  |_finalized_model.sav
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. input
2. src
3. output
4. lib

1. Input folder **-** It contains all the data that we have for analysis. There are two csv files in our case,
    - Data1
    - Data2

2. Src folder - This is the most important folder of the project. This folder contains all the modularized code for all the above steps in a modularized manner. This folder consists of:
    - Engine.py
    - ML_Pipeline

The ML_pipeline is a folder that contains all the functions put into different python files, which are appropriately named. These python functions are then called inside the engine.py file.

3. Output folder **–** The output folder contains the best-fitted model that we trained for this data. This model can be easily loaded and used for future use and the user need not have to train all the models from the beginning.
   Note: This model is built over a chunk of data. One can obtain the model for the entire data by running engine.py by taking the entire data to train the models.

4. Lib folder **-** This is a reference folder. It contains the original ipython notebook that we saw in the videos.

Project Takeaways

1. Understanding the business problem.
2. Importing the dataset and required libraries.
3. Performing basic Exploratory Data Analysis (EDA).
4. Removal of unwanted features and missing data handling if required, using appropriate methods.
5. Checking data distribution using statistical techniques.
6. Using python libraries such as matplotlib and seaborn for data interpretation and advanced visualizations.
7. Splitting Dataset into Train and Test using sklearn.
8. Training a model using Classification techniques like Logistics Regression, Naïve Bayes, Decision Tree Classifier, Random Forest Classifier and Support Vector Machine.
9. Tuning hyper-parameters of models to achieve optimal performance.
10. Making predictions using the trained model.
11. Gaining confidence in the model using metrics such as accuracy score, confusion matrix, recall, precision and f1 score
12. Handling the unbalanced data using various methods.
13. How Target variable is dependent on the values of Input features.
14. Selection of the best model based on performance metrics.
15. Saving the best model in pickle format for future use.