BCG TASK 3 – Feature Engineering Modelling

Q1. Is this problem best represented as classification or regression?

This Problem Best represent for Classification , Because We are trying to predict ,will the customer churn or not. We are not going to predict the possibilities of Churning. It Is a Binary Classification. So we go with problem is Best represent Classification .

Q2. What kind of model performance do you think is appropriate?

Accuracy: The proportion of correctly classified churn and non-churn customers. While accuracy is essential, it may not be the only metric to consider, especially if the data is imbalanced (e.g., a small percentage of churn customers compared to non-churn customers).

Precision and Recall: Precision measures the proportion of true churn cases among the predicted churn cases, while recall measures the proportion of true churn cases that were correctly predicted. High precision indicates low false positives (misclassifying non-churn customers as churn), and high recall indicates low false negatives (correctly identifying churn customers). Both metrics are essential, but their importance depends on the business's priorities. For example, in a churn classification problem, recall might be more critical because you want to identify as many churn customers as possible to take appropriate retention measures.

F1-Score: The F1-score is the harmonic mean of precision and recall and is a useful metric when there is an imbalance in the dataset. It considers both false positives and false negatives, providing a single value that represents the overall performance of the model.

F1-Score is Used Because Dataset is Imbalance with Churn (Retention or Churned)

Q3. How would you tie business metrics such as profits or savings to the model performance?

Define Business Goals: Start by understanding the business objectives related to churn. For example, the goal might be to reduce customer churn by a certain percentage to increase overall revenue or to minimize the costs associated with churn, such as marketing efforts to acquire new customers.

Assign Costs and Benefits: Quantify the costs associated with false positives (predicting a customer will churn, but they don't) and false negatives (predicting a customer won't churn, but they do) as well as the benefits of true positives (correctly predicting churn) and true negatives (correctly predicting non-churn). These costs and benefits could include factors like lost revenue from churn, cost of retaining customers, and potential revenue from new customers.

Create a Cost-Benefit Matrix: Construct a cost-benefit matrix that lays out the financial impact of different model predictions (true positive, true negative, false positive, false negative). This matrix will help you quantify the net financial gain or loss associated with the model's performance.

Feature Engineering :

Feature engineering is a crucial step in the machine learning process where you create new and meaningful features from the existing data to improve the performance of your models. It involves selecting, transforming, and creating features that provide valuable information to the model, making it easier to learn patterns and make accurate predictions.

1.Handling Missing Values :

The Dataset didn’t have any null valuses .

2.Feature Selection :

Unwanted Features like Id and after feature engineering some feature are not neaded because it creates redendency. Some feature we don’t really need , It will depreciate model performance or it may create overfitting.

3.Categorical Encoding :

Features like Channel_sales, Origin_up, Has_gas have categorical data so those features are encoded. We use two types of Encoding Technique : 1. Target Label encoding, One-hot encoding. Both Encoding Techniques are used for better model performance.

4.Time-Series Feature :

Features Like dates are considered as ordinal data. So we Use Ordinal Encoding Technique . So the data is Organised without loss of feature.

5. Feature Selection:

Based on high Correlation Some features are dropped. It will also increase the model performance.

6.Feature Importance Analysis:

After training your model, analyze feature importances to identify which features have the most impact on the predictions. This analysis can help you validate the effectiveness of the engineered features and guide further improvements.

Random Forest Classifier :

Using a Random Forest Classifier for churn prediction is a popular and effective approach due to its ability to handle complex relationships in the data, deal with non-linearities, and handle both numerical and categorical features well.

STEPS:

1.Data Preprocessing and Feature Engineering: Prepare your data by cleaning it, handling missing values, encoding categorical features, and performing feature engineering as discussed earlier. Ensure that your target variable (churn) is properly defined, and the dataset is split into training and testing sets.

2.Import Libraries: Import the necessary libraries, including scikit-learn for machine learning, numpy, and pandas for data manipulation, and matplotlib/seaborn for visualization.

3.Create the Random Forest Classifier: Initialize the Random Forest Classifier by setting hyperparameters like the number of trees in the forest, the depth of each tree (if desired), and other relevant parameters. You can fine-tune these hyperparameters later through cross-validation.

4.Train the Model: Fit the Random Forest Classifier on your training data using the fit method. The model will learn from the features and their corresponding target labels.

5.Feature Importance Analysis: After training the model, analyze the feature importances using the feature_importances_ attribute of the trained classifier. This analysis will help you understand which features are most relevant for churn prediction.

6.Make Predictions: Use the trained Random Forest Classifier to make predictions on the test dataset using the predict method. This will give you the churn predictions for the test instances.

7.Evaluate the Model: Assess the performance of your Random Forest Classifier using various evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. You can use scikit-learn's classification_report and confusion_matrix functions for this purpose.

8.Hyperparameter Tuning: Perform hyperparameter tuning using techniques like Grid Search or Random Search to find the optimal hyperparameters for your Random Forest Classifier. This step can further improve the model's performance.