**Project Bank Marketing**

**BLOG/ Article**

**Evaluation Project**

**Phase – IV**

**Batch DS2403**

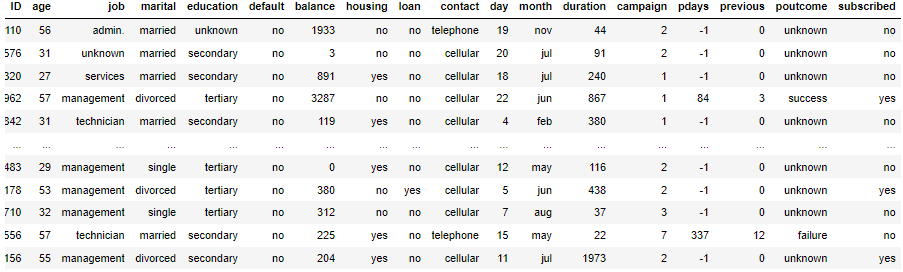
**-Harsh Kumar**

**Project Bank Marketing**

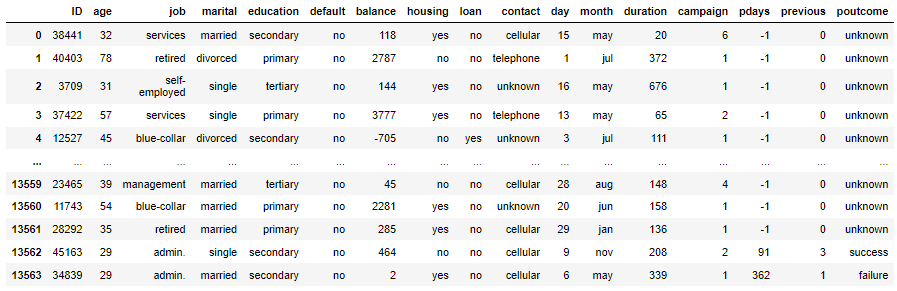
**Predicting Whether the Customer Will Subscribe to Term Deposit (Fixed Deposit)**

This project is focused on predicting whether a customer will subscribe to a term deposit (fixed deposit) using historical data. I worked with two datasets: train.csv for building and training the model, and test.csv for testing and making final predictions. Here’s a breakdown of what I have done:

**train.csv**

****

**test.csv**

****

**Problem Definition**

The Portuguese bank is facing a revenue decline due to a decrease in the frequency of term deposits made by its clients. Term deposits are essential for banks, as they provide a stable source of income and enable banks to invest in higher-yield financial products. Additionally, having clients with term deposits increases the likelihood of cross-selling other financial products, such as funds and insurance, thereby enhancing overall revenue.

To address this issue, the bank aims to identify existing clients who are most likely to subscribe to term deposits. By focusing their marketing efforts on these clients, the bank can optimize its resources and increase the effectiveness of its outreach campaigns. Given the high costs associated with telephonic marketing campaigns, it is crucial to accurately predict which customers are likely to convert. This targeted approach not only saves resources but also increases the chances of successful subscriptions.

The dataset provided includes various attributes of the clients, such as their age, job type, marital status, and other demographic factors. It also contains information related to the marketing calls, including call duration, the day and month of the call, and the outcome of these calls. The primary objective of this project is to predict whether a client will subscribe to a term deposit based on this available data.

By developing a predictive model, the bank can better strategize its marketing efforts, ensuring that they reach out to clients with a higher likelihood of conversion. This initiative aims to reverse the revenue decline and enhance customer engagement, ultimately benefiting the bank's financial health.

The task is to predict whether a customer will subscribe to a term deposit based on their demographic, financial, and interaction history with the bank. To achieve this, we are working with two datasets:

* **train.csv:** Used for training the machine learning model. It contains customer-related information and whether they subscribed to a term deposit.
* **test.csv:** Used to test the trained model and make predictions on whether new customers will subscribe.

The problem is a **binary classification** task, where the goal is to predict if a customer will subscribe (yes) or not (no), helping the bank target the right customers for their marketing campaigns.

**Data Analysis**

I began by loading both the **df (train.csv)** and **df\_test (test.csv)** datasets. For a better understanding of the data structure, I printed the number of rows and columns for both datasets.

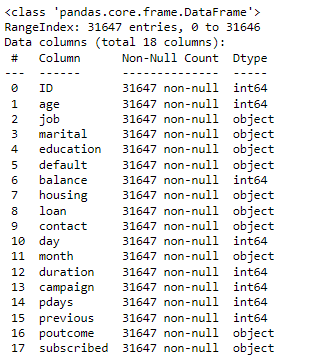
Upon inspecting the df dataset, I found that it contains 9 numerical columns and 8 categorical columns. The numerical columns include features such as ID, age, balance, day, duration, campaign, pdays, and previous. These columns are represented as integers, meaning they contain numeric data that will require scaling during preprocessing.

The categorical columns, on the other hand, include job, marital status, education, default, housing, loan, contact, month, and poutcome. These are represented as objects, meaning they contain text or categorical data that will need to be encoded into numeric values for machine learning algorithms to process effectively.

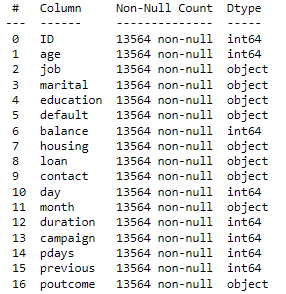
I then used **df.describe()** to generate a statistical summary of the numerical columns. This gave insights into the mean, minimum, and maximum values of features like age and balance, which will help in further analysis and preprocessing.

Finally, I checked for missing values using **df.isnull().sum()**. The output showed that there were no missing values in the dataset, so no further action was needed to handle missing data.

**train.csv**



**test.csv**

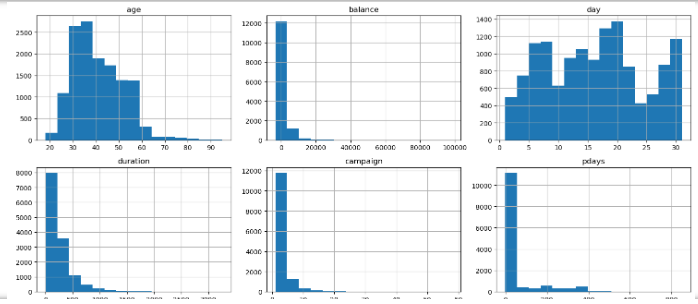
****

The train.csv dataset contains the target variable subscribed, which indicates whether a customer has subscribed to a term deposit, while the test.csv dataset does not include this column, as it is used solely for making predictions.

**EDA**

In the Exploratory Data Analysis (EDA), I generated histograms for key numerical features. Here are the highlights:

* Age: Most customers are between 20 and 40 years old.
* Balance: Highly skewed to the right; most customers have low balances.
* Day: Fairly uniform distribution of contact days across the month.
* Duration: Skewed right; most contact durations are short.
* Campaign: Most customers were contacted fewer than 10 times.
* Pdays: Skewed right; most customers have a pdays value of 0.
* Previous: Most customers were contacted only once in prior campaigns.

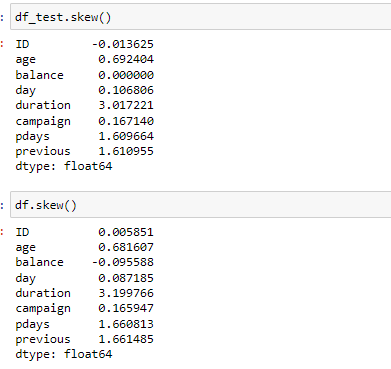


To analyze the categorical features in the dataset, I employed count plots for each categorical variable. This method allowed me to visually count and compare the occurrences of different categories within each feature, providing clear insights into the distribution of customer demographics and behaviors. By examining features such as **job**, **marital status**, **education**, **default**, **housing**, **loan**, **contact**, **month**, and **poutcome**, I could identify prevalent categories and potential relationships with the target variable, **subscribed**. This understanding helps in tailoring marketing strategies, as it reveals which customer segments may be more likely to respond positively to term deposit offers.

I applied the same count plot technique to both the training dataset (df) and the test dataset (df\_test). This comparative analysis allowed me to observe similarities and differences in categorical features across both datasets, enhancing my understanding of the customer base and ensuring that the model training is grounded in representative data.

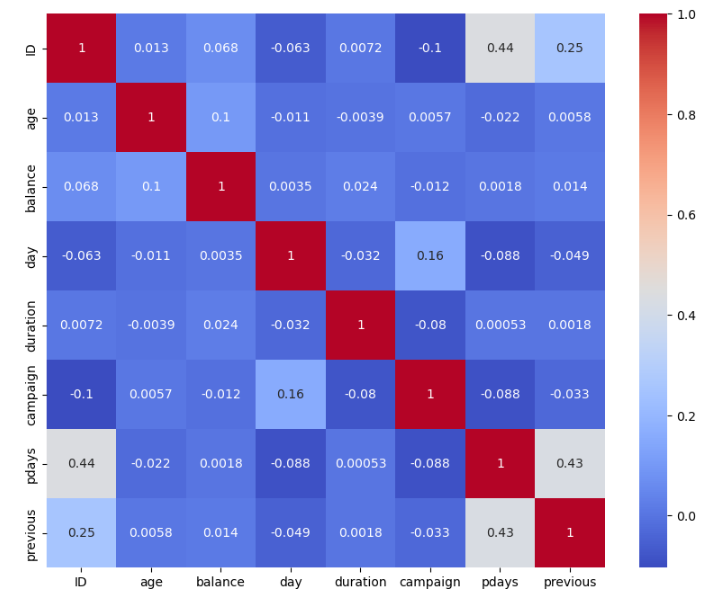
Additionally, I used box plots to detect outliers in the numerical features, effectively visualizing the spread and identifying extreme values that could skew the model's performance. Recognizing and handling outliers is crucial for improving the accuracy of predictions. I also employed distribution plots (dist plots) to assess the skewness of the data distributions. This analysis helped me identify which numerical features might need transformation to achieve a more normal distribution, which is important for many machine learning algorithms that perform better with normally distributed data.

Overall, these visualizations provided a comprehensive understanding of both categorical and numerical features, laying a strong foundation for informed data preprocessing and subsequent modeling decisions.

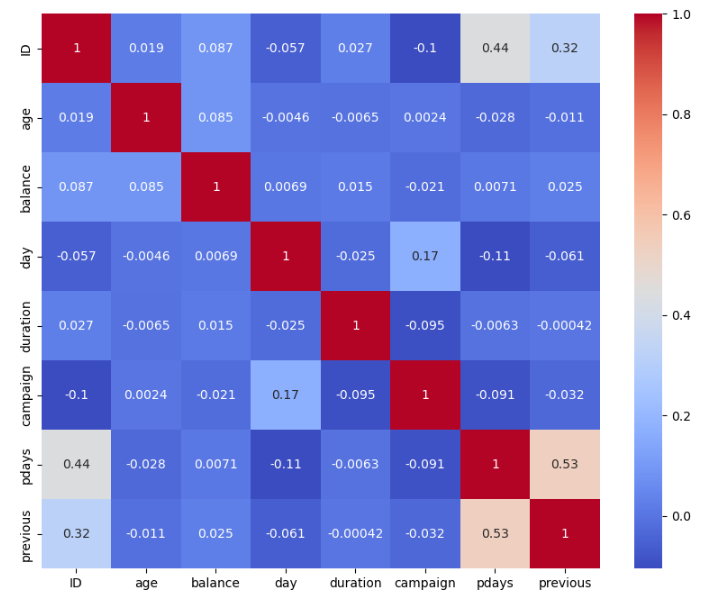


To further understand the relationships between numerical features in the dataset, I generated correlation matrices for both the training dataset (df) and the test dataset (df\_test). By calculating the correlation coefficients, I could assess how closely related each pair of numerical variables is. The output was visualized using heatmaps, which clearly depict the strength and direction of these relationships.

The heatmaps, with values ranging from -1 to 1, indicate positive correlations (values close to 1) and negative correlations (values close to -1). This analysis helped identify which features are strongly correlated, guiding feature selection and engineering efforts. For example, if two features are highly correlated, it may be beneficial to retain only one of them to reduce redundancy in the model. This step is crucial for improving model interpretability and performance.

**train\_corr\_matrix**

**test\_corr\_matrix**

****

**Pre-Processing**

In the pre-processing phase, my first step was to ensure that both the training dataset (df) and the test dataset (df\_test) were free of missing values. I performed a thorough examination using functions to check for null entries in each dataset. Confirming that there were no missing values was crucial, as handling these appropriately is vital for maintaining data integrity and model accuracy. With this assurance, I proceeded to encode the categorical variables into numerical values using one-hot encoding. This method transforms categorical data into a binary format, allowing the model to interpret these values effectively. Each category is represented as a separate column, with a 1 or 0 indicating the presence or absence of that category for each observation.

Next, I focused on standardizing the numerical features to ensure uniformity across the dataset. To achieve this, I applied StandardScaler, which standardizes the data by centering it around a mean of 0 and scaling it to have a standard deviation of 1. This process is essential because many machine learning algorithms are sensitive to the scale of the input data. I made sure to apply this scaling consistently to both the training and test datasets, which helps maintain the relationship between the data points.

In addition to encoding and scaling, I also addressed outliers in the dataset. By using insights gained from box plots, I identified extreme values that could negatively impact the model's performance. Removing these outliers was a critical step to ensure that the model could learn from a cleaner dataset, ultimately leading to better predictions.

After these adjustments, I split the training dataset into features (X) and the target variable, **subscribed** (y), to prepare it for modeling. For the target variable, I used LabelEncoder to convert it into a numerical format, enabling the models to process it effectively.

In the data preparation process, I established a structured pipeline for handling the categorical features. I utilized OneHotEncoder, which is designed to convert categorical values into a one-hot numeric array, ensuring that any unknown categories encountered during prediction are ignored. To efficiently apply this transformation only to the specified categorical features, I employed ColumnTransformer, which allows for selective preprocessing of the data while leaving the remaining features unchanged.

After fitting the preprocessor to the training dataset, I transformed it into a preprocessed dataset named **X\_preprocessed**. I ensured that the same preprocessing steps were applied to the test dataset (df\_test), resulting in **X\_test\_preprocessed**. This meticulous approach ensured that both datasets were consistently prepared for modeling, laying a solid foundation for building and evaluating the machine learning models.

**Building Machine Learning Models**

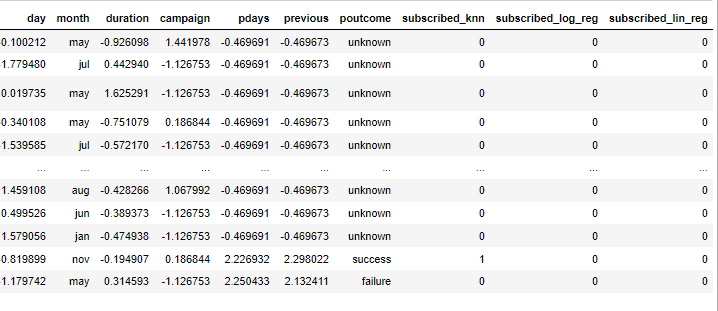
n this part of the project, I focused on building and evaluating three distinct machine learning models to predict whether customers would subscribe to a term deposit. First, I implemented the K-Nearest Neighbors (KNN) model, which operates on the principle of identifying the closest data points in the training set to make predictions. This model analyzes the subscription status of nearby customers, allowing it to classify a new customer based on the majority subscription status of their nearest neighbors.

Next, I utilized a Linear Regression model to estimate the likelihood of a customer subscribing. This model calculates a continuous probability value, which I then converted into binary predictions (0 or 1) using a threshold of 0.5. This approach allows me to interpret the predicted probabilities in a straightforward manner, determining whether a customer is likely to subscribe based on their features.

Finally, I implemented a Logistic Regression model, a popular choice for binary classification tasks. Logistic Regression calculates the probability of subscription and applies a logistic function to ensure that the output values are between 0 and 1, making it easy to classify customers into subscribed or not subscribed categories.

After training all three models on the preprocessed dataset, I generated predictions on the test dataset. The predictions for each model were stored in new columns within the test dataset, specifically labeled as **subscribed\_knn**, **subscribed\_log\_reg**, and **subscribed\_lin\_reg**. To facilitate further analysis and sharing of results, I saved these predictions to a CSV file

named **predictions.csv**. Additionally, I printed the results for each model to review the predicted subscription statuses, enabling a comparison of their performance and insights into how different algorithms approached the same problem.



**Concluding Remarks**

In this project, I successfully developed a predictive model to determine whether customers would subscribe to a term deposit. I began by defining the problem and performing thorough data analysis on the provided datasets. Through Exploratory Data Analysis (EDA), I gained insights into customer demographics and behaviors, which informed my modeling decisions.

I implemented a structured pre-processing pipeline that handled missing values, encoded categorical variables, and scaled numerical features, ensuring the data was ready for modeling. After building three different machine learning models—K-Nearest Neighbors (KNN), Linear Regression, and Logistic Regression—I evaluated their performance on the test dataset. The predictions were saved for further analysis.

This work not only highlighted the importance of data preparation in machine learning but also demonstrated how different algorithms can provide valuable insights into customer behavior. Future work could focus on further refining the models and exploring additional features to enhance prediction accuracy.