T.C. YEDITEPE UNIVERSITY

ACM 476 PROJECT REPORT

[BANK MARKETING PREDICTION]

Group Members:
Arda Altınörs - 20211308013
Çağrı Mercan - 20201314008

1. INTRODUCTION

1.1. Business Environment

The banking or financial industry is probably the main focus of the business environment for the Bank Term Deposit initiative. It's crucial to take market trends, legal frameworks, and economic situations into account in this situation. Understanding the opportunities and difficulties banks encounter in promoting term deposits can be gained through an analysis of the business environment.

Since the Bank Term Deposit initiative's dedication to the banking and financial sectors raises a number of important issues. Let's explore the significance of economic conditions, regulatory frameworks, and market developments in more detail:

1. Market Trends:

Customer Behavior: Analyzing market trends involves understanding how customers in the financial market behave, what factors influence their decisions, and how they respond to various financial products, including term deposits.

Competitive Landscape: Examining market trends helps identify the positioning of term deposits relative to other financial products offered by competitors. It provides insights into market demand and helps in developing effective marketing strategies.

2. Legal Frameworks:

Regulatory Compliance: A number of regulations apply to the banking sector. To maintain compliance and keep away from conflict with the law, the Bank Term Deposit project has to understand and operate by these legal frameworks.

Consumer Protection: Legal considerations may include aspects of consumer protection, ensuring transparency in financial transactions, and maintaining the integrity of the banking system.

3. Economic Situations:

Interest Rate Environment: The performance of term deposit programs is significantly influenced by economic conditions, particularly changes in interest rates. Term deposits may become less appealing than alternative investment options in response to changes in interest rates.

Consumer Confidence: Economic conditions influence consumer confidence, affecting their willingness to commit to long-term financial products like term deposits. Understanding economic indicators is key to predicting customer behavior.

4. Opportunities and Difficulties:

Customer Segmentation: It's critical to identify target consumer categories and modify marketing tactics in relation to behavioral and demographic variables. By using the dataset to identify trends in client feedback, banks may improve their strategy.

Marketing Effectiveness: Evaluating the effectiveness of past marketing campaigns provides insights into what works and what doesn't. This data informs future term deposit marketing campaigns.

5. Business Environment Analysis:

SWOT Analysis: Conducting a comprehensive SWOT analysis (Strengths, Weaknesses, Opportunities, Threats) based on the dataset helps in understanding the internal and external factors influencing the Bank Term Deposit initiative.

Risk Assessment: For the term deposit program to be successful in the long run, risks including economic downturns and changes in regulatory environments must be identified and addressed.

6. Data-Driven Decision-Making:

Predictive Analytics: Leveraging data analytics to make predictions about future trends and customer behavior enables proactive decision-making.

Continuous Monitoring: The world of business is constantly evolving. Strategies are kept in line with changing market conditions by consistently updating and observing the dataset.

In conclusion, the Bank Term Deposit effort will not succeed unless it adopts a comprehensive strategy that takes economic conditions, regulatory frameworks, and

market trends into account. The dataset is a useful resource for learning about the specifics of the business environment, which improves strategic planning and well-informed decision-making.

1.2. Dataset Description

Dataset Title: Direct Marketing Campaigns for Bank Term Deposits

Source and Purpose: This dataset, which comes from a Portuguese bank, includes information from phone calls and direct marketing campaigns. Determining the probability of clients subscribing to the bank's term deposits was the main goal of these campaigns. The primary purpose of the dataset is to train prediction models that estimate the behavior of customers with regard to subscribing to these term deposits.

Dataset Features:

Age: Represents the age of the customer. This variable, which is numerical in nature, is essential for understanding various demographic groups that might have various preferences towards becoming term deposit subscribers.

Job: The customer's occupation is described in this categorical variable. It can involve a variety of jobs, such as management, blue-collar work, teaching, etc. A customer's profession may have an impact on their investing habits and financial decisions.

Marital Status: A feature that classifies a customer's relationship status as single, married, divorced, etc. One important consideration while making financial decisions is one's marital status.

Education: This variable indicates the customer's highest degree of education attained, such as elementary, secondary, or higher education. Investment behavior and financial literacy can be correlated with education level.

Default: A binary variable indicating whether the customer has credit in default (yes or no). This is a crucial sign of risk and financial stability.

Balance: This numerical feature represents the balance in the customer's bank account. It offers information about the client's financial situation and potential investments.

Housing Loan: A binary variable that states whether the customer has a housing loan (yes or no). If a customer has large debts, such as mortgages, it may influence their choice to invest in term deposits.

Contact Communication Type: This categorical variable indicates the method used to contact the customer, such as via telephone or cellular phone. The success rate of various communication techniques in campaigns may vary.

Day: A numerical feature representing the day of the month when the last contact with the customer occurred. This can be useful in understanding out response patterns and the best times to get in touch.

Duration: This numerical variable measures the length of the last contact with the customer, in seconds. Longer periods could suggest more client interaction or interest.

Campaign Contacts Count: A numerical count of how many contacts were made with the customer during the current campaign. This can show the level of effort put into converting a potential customer.

pdays: Represents the number of days that have passed since the customer was last contacted from a previous campaign. This is a numerical variable that is essential to evaluating the frequency of customer follow-up.

Subscription Status to a Term Deposit (Dependent Variable): This is a binary categorical variable, typically represented by values such as "yes" or "no". It shows whether the customer signed up for a term deposit after the marketing effort.

"Yes": Indicates that the customer agreed to subscribe to the term deposit. This outcome is of particular interest as it represents a successful transformation in the marketing campaign.

"No": Indicates that the consumer decided not to sign up for a term deposit after the campaign.

1.3.Objective (s)

This project's main goal is to create a predictive model that can categorize the results of a bank's direct term deposit marketing campaign. The model's specific goal is to estimate, using a variety of customer factors and marketing data, whether a consumer will sign up for a term deposit.

For this project, we are employing Classification Analytics, a branch of Supervised Learning in Machine Learning (ML). This approach is chosen because the target variable -subscription status to a term deposit - is categorical ('yes' or 'no'). Our task is to classify each customer into one of these two categories.

We expect to find specific patterns and relationships within the data that significantly influence a customer's decision to subscribe to a term deposit. Identifying these key predictors will not only provide insights into customer behavior but also guide the bank in optimizing future marketing strategies. For instance, we might discover that certain demographics or customer segments are more likely to subscribe, which can lead to more targeted and cost-effective marketing efforts.

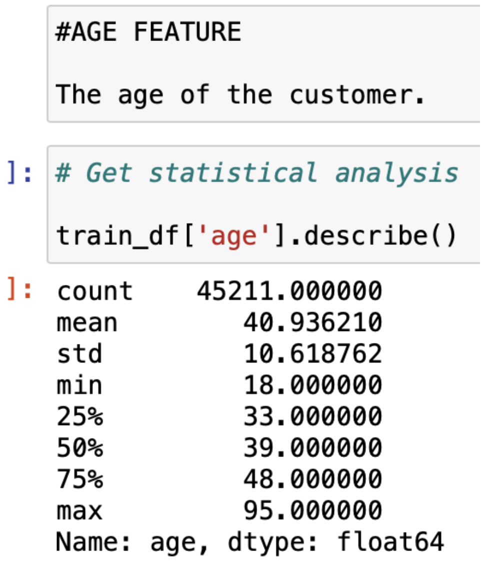
Several preprocessing processes have been performed before the model is trained. These include categorical variable encoding, missing value management, and data cleaning. Preprocessing improves data quality and ensures that it is in a format suitable for effective model training.

We will train various classification algorithms on the preprocessed dataset. These will include logistic regression, decision tree and K-nearest neighbor. Each model will be evaluated to determine its performance in accurately predicting the campaign's outcome.

2. DESCRIPTIVES

Statistical Summaries:

-For each feature in our dataset, we generated statistical summaries using Python's .describe() function. This approach provided us with essential metrics, such as mean, median, standard deviation, minimum, maximum, and quartile values for numerical features. These metrics are fundamental in understanding the general behavior and distribution characteristics of each feature. You can see an example codeblock below:



```
#AGE FEATURE

The age of the customer.
```

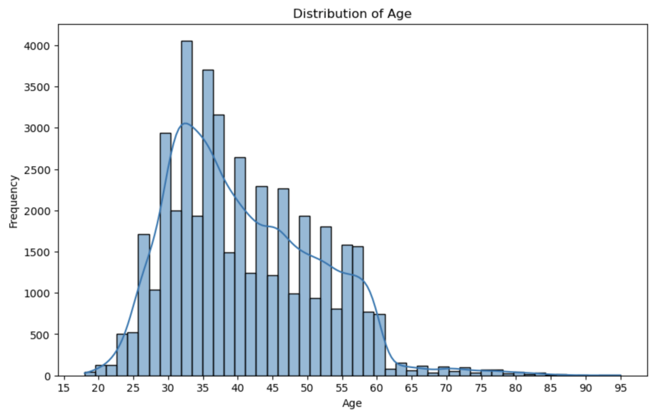
```
]: # Get statistical analysis

   train_df['age'].describe()
```

```
]: count    45211.000000
   mean        40.936210
   std         10.618762
   min         18.000000
   25%         33.000000
   50%         39.000000
   75%         48.000000
   max         95.000000
   Name: age, dtype: float64
```

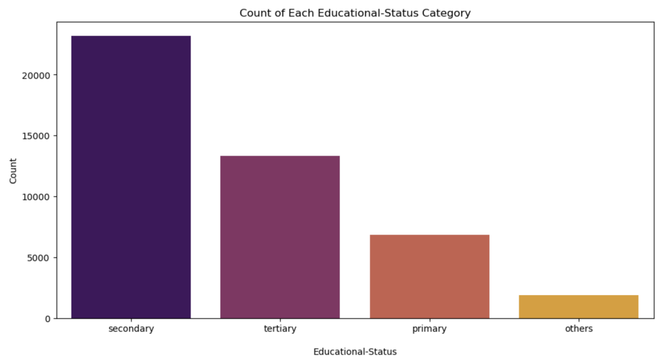
Visualization Approach :

-In our descriptive analysis, we employed various visualization techniques to complement our statistical summaries:
Numerical Features: For these, histograms with Kernel Density Estimates (KDE) were utilized. This approach allowed us to observe the distribution and identify skewness in the data. Box plots were also used to highlight outliers and the overall data spread.

Categorical Features: Bar charts were the primary tool for these features, providing a clear view of category frequencies and distributions. These visualizations were key in revealing underlying patterns and aiding in our understanding of each feature's characteristics. You can see some examples below :



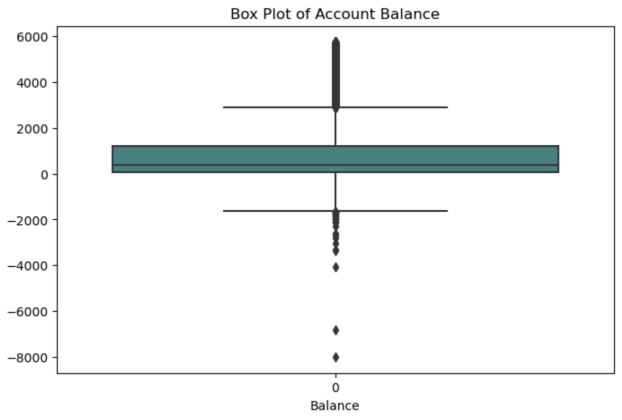
• The most prevalent age range is [30 - 47]

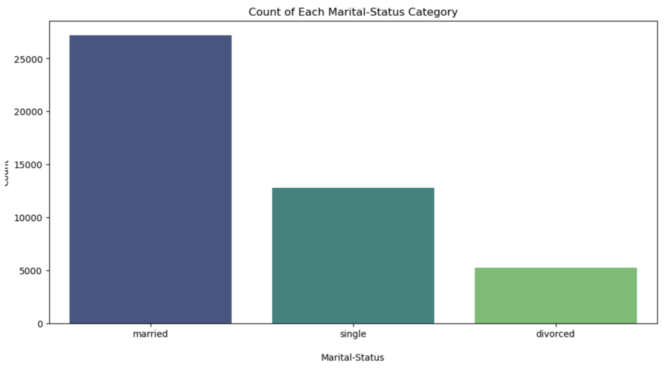


Feature-Specific Analysis:

-Each feature was examined individually to understand its unique characteristics. This level of detail helped us identify features with significant variance, unusual distributions, or imbalances that could influence the modeling process.

-We paid special attention to ensure that the visualizations and statistical analyses were aligned with the nature of each feature, whether numerical or categorical.



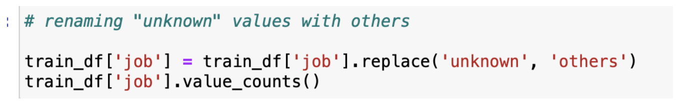


3. DATA PREPROCESSING

In the data preprocessing stage, our focus was on enhancing the dataset's quality for more accurate predictive modeling. Here are some key steps:

3.1 Feature Transformation

- For features containing 'unknown' values, we transformed these entries into 'other'. This approach avoids data loss and ensures a more comprehensive representation of the dataset, enhancing the integrity of our analysis. For example:



```
# renaming "unknown" values with others

train_df['job'] = train_df['job'].replace('unknown', 'others')
train_df['job'].value_counts()
```

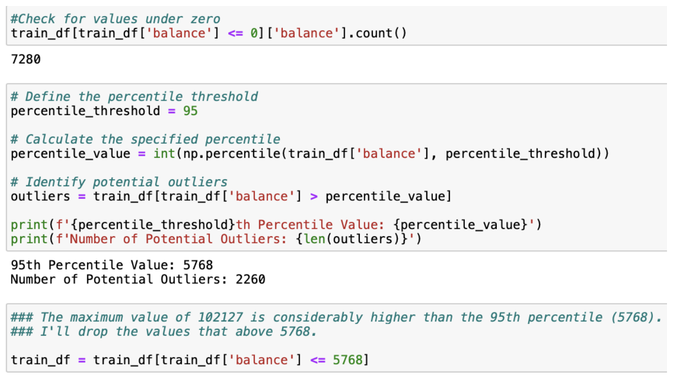
3.2 Outlier Detection and Treatment

We focused on addressing outliers in the **'balance'** feature, essential for the accuracy of our predictive model.

Identifying Outliers: We used statistical methods to identify upper outliers above the 95th percentile and lower outliers below the 5th percentile in the 'balance' data.

Removing Outliers: Outliers exceeding these thresholds were removed from the dataset, aiming to normalize the distribution of account balances.

Example Codeblock below:



```
#Check for values under zero
train_df[train_df['balance'] <= 0]['balance'].count()

7280

# Define the percentile threshold
percentile_threshold = 95

# Calculate the specified percentile
percentile_value = int(np.percentile(train_df['balance'], percentile_threshold))

# Identify potential outliers
outliers = train_df[train_df['balance'] > percentile_value]

print(f'{percentile_threshold}th Percentile Value: {percentile_value}')
print(f'Number of Potential Outliers: {len(outliers)}')

95th Percentile Value: 5768
Number of Potential Outliers: 2260

### The maximum value of 102127 is considerably higher than the 95th percentile (5768).
### I'll drop the values that above 5768.

train_df = train_df[train_df['balance'] <= 5768]
```

Results: This approach was chosen to mitigate the skewing effect of extreme values on our model.

Post-removal, visualizations like KDE plots and box plots confirmed a more regularized distribution, improving the dataset's reliability for predictive analysis.

Handling outliers in the 'balance' feature was a crucial step in refining our dataset, setting a solid foundation for accurate modeling.

3.3 Feature Reduction

In our dataset, we identified certain features that, due to their characteristics, offered limited value for our predictive analysis. The decision to remove these features was driven by their distribution and the nature of their data, which could potentially skew our model's performance.

a) 'Default' Feature
- Observation: A detailed examination of the 'default' feature revealed a highly skewed distribution, where the vast majority of values were 'no'.

- Action: Considering its overwhelming imbalance and the likelihood of minimal impact on the predictive outcome, we decided to drop this feature.

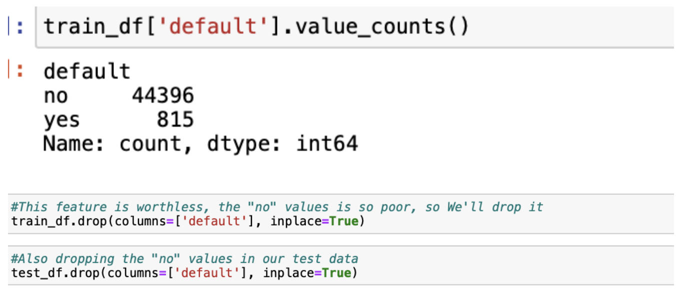
b) Previous' Feature:
- Observation: The 'previous' feature predominantly consisted of 0 values. This lack of variance indicated that the feature wouldn't contribute significantly to differentiating outcomes in our predictive model.

- Action: Based on this observation, the 'previous' feature was removed to simplify the model and focus on more informative attributes.

c) Poutcome' Feature:
-Observation: We noted that a substantial portion of the 'poutcome' (outcome of the previous marketing campaign) data was categorized as 'unknown', limiting its usefulness and reliability for analysis.

- Action : To avoid the potential bias and inaccuracies this could introduce, we opted to exclude 'poutcome' from our dataset.

Here is an example codeblock:



```
|: train_df['default'].value_counts()

|: default
   no     44396
   yes      815
   Name: count, dtype: int64
```

```
#This feature is worthless, the "no" values is so poor, so We'll drop it
train_df.drop(columns=['default'], inplace=True)
```

```
#Also dropping the "no" values in our test data
test_df.drop(columns=['default'], inplace=True)
```

'Default', 'previous', and 'poutcome' feature reduction was an important part of our preprocessing stage. It demonstrates our dedication to adjusting the dataset for the best possible predictive performance, guaranteeing the accuracy and applicability of our analysis.

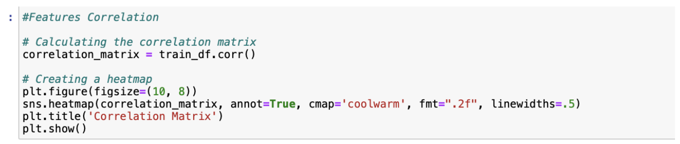
3.4 Correlation Analysis

Understanding the relationships between different features is vital for building effective models.

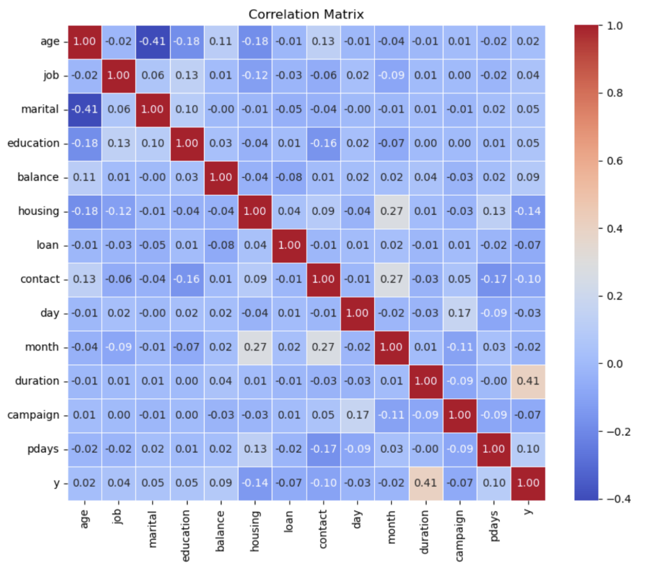
a) Correlation Matrix Computation

- Procedure: We calculated a correlation matrix for the dataset. This matrix quantifies the degree to which different variables are linearly related.

- Purpose: The correlation matrix helps in identifying features that are highly correlated with each other. Features with high multicollinearity can negatively impact some models, and identifying them early is key for potential feature selection or removal.



```
: #Features Correlation

  # Calculating the correlation matrix
  correlation_matrix = train_df.corr()

  # Creating a heatmap
  plt.figure(figsize=(10, 8))
  sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
  plt.title('Correlation Matrix')
  plt.show()
```



3.5 K-Means Clustering

To enhance our understanding of customer segments, we applied the K-Means clustering algorithm to the dataset, focusing on key attributes like duration and balance. These features were selected for their relevance to customer financial behavior and demographic details, which are crucial for segment-specific marketing strategies.

Steps Involved:

- Feature Standardization: Standardized features to ensure equal contribution to the algorithm, eliminating bias due to differing scales.

- Optimal Cluster Determination: Utilized the elbow method to determine the optimal number of clusters, supplemented by silhouette scores to assess cluster validity.

- Algorithm Application: Implemented K-Means using scikit-learn, assigning data points to clusters based on feature similarities.

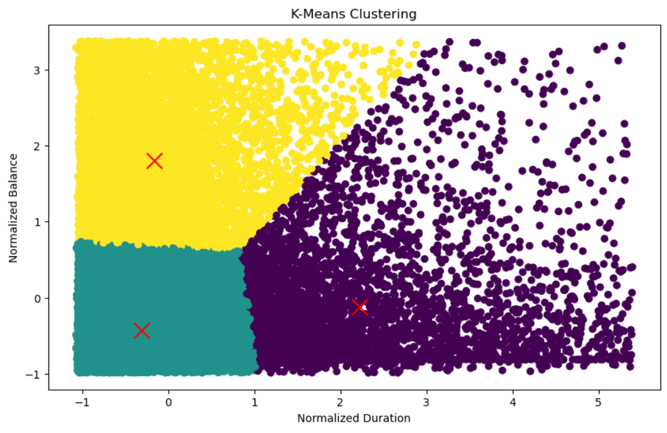
Results:

- Cluster Profiles: Analysis of clusters revealed distinct customer segments which were characterized by their duration and balance metrics.

Visualization: Cluster distributions were visualized using scatter plots, aiding in the

visual assessment of the separation and coherence of the clusters.

K-Means clustering provided crucial insights into customer segmentation, informing targeted marketing strategies and helping to optimize resource allocation in promotional activities.



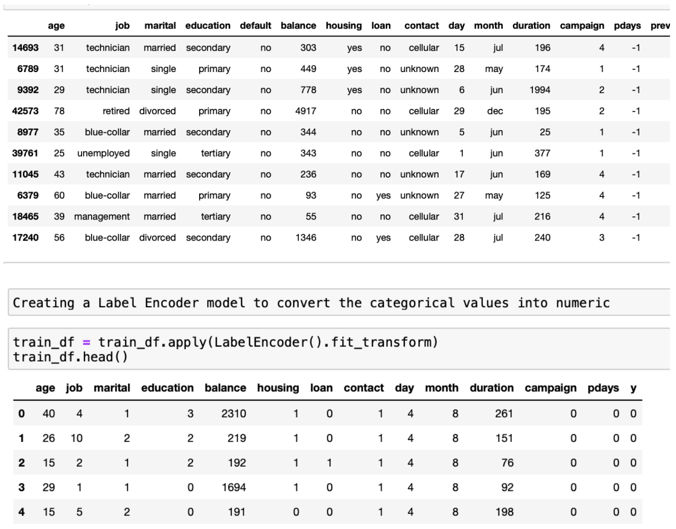
3.6 Encoding Categorical Features

One crucial step in preprocessing was the transformation of categorical variables into a format that can be easily used by machine learning algorithms.

a) Label Encoding:
- Process: We utilized Label Encoding, converting each categorical value into a unique integer. This step was implemented using the ‘LabelEncoder‘ from the scikit-learn library.

- Rationale: Many machine learning models, especially those based on mathematical functions, require numerical input. Label Encoding transforms categorical data into a machine-readable form without expanding the feature space.

Here are example images for before and after the *encoding* process:



	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	prev
14693	31	technician	married	secondary	no	303	yes	no	cellular	15	jul	196	4	-1	
6789	31	technician	single	primary	no	449	yes	no	unknown	28	may	174	1	-1	
9392	29	technician	single	secondary	no	778	yes	no	unknown	6	jun	1994	2	-1	
42573	78	retired	divorced	primary	no	4917	no	no	cellular	29	dec	195	2	-1	
8977	35	blue-collar	married	secondary	no	344	no	no	unknown	5	jun	25	1	-1	
39761	25	unemployed	single	tertiary	no	343	no	no	cellular	1	jun	377	1	-1	
11045	43	technician	married	secondary	no	236	no	no	unknown	17	jun	169	4	-1	
6379	60	blue-collar	married	primary	no	93	no	yes	unknown	27	may	125	4	-1	
18465	39	management	married	tertiary	no	55	no	no	cellular	31	jul	216	4	-1	
17240	56	blue-collar	divorced	secondary	no	1346	no	yes	cellular	28	jul	240	3	-1	

```
Creating a Label Encoder model to convert the categorical values into numeric
```

```
train_df = train_df.apply(LabelEncoder().fit_transform)
train_df.head()
```

	age	job	marital	education	balance	housing	loan	contact	day	month	duration	campaign	pdays	y
0	40	4	1	3	2310	1	0	1	4	8	261	0	0	0
1	26	10	2	2	219	1	0	1	4	8	151	0	0	0
2	15	2	1	2	192	1	1	1	4	8	76	0	0	0
3	29	1	1	0	1694	1	0	1	4	8	92	0	0	0
4	15	5	2	0	191	0	0	1	4	8	198	0	0	0

3.7 Train-Test Split

a)Implementation:

- We divided our dataset into ‘X_train‘, ‘y_train‘ for training, and ‘X_test‘, ‘y_test‘ for testing purposes. The ‘X‘ variables contain the input features, while ‘y‘ refers to the target variable.We partitioned our dataset into a training set, comprising 80% of the data, and a test set, containing the remaining 20%.

b)Purpose:

- This division is crucial for evaluating the performance of our machine learning models. It allows models to learn and adapt to patterns in the training data (‘X_train‘, ‘y_train‘) and subsequently be evaluated on unseen data (‘X_test‘, ‘y_test‘).

The Train-Test Split is an essential step in machine learning preprocessing, ensuring that our models are tested on unbiased, unseen data, thereby providing a realistic assessment of their predictive performance.

4. MACHINE LEARNING IN ACTION

4.1.Selected algorithms

In our project, we employed three distinct supervised learning algorithms: Logistic Regression, K-Nearest Neighbors (KNN), and

Decision Trees. Each of these methods offers unique advantages and approaches to predictive modeling, making them suitable for our objective of predicting outcomes in a bank's direct marketing campaign.

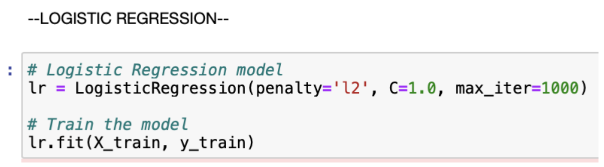
4.1.2 Logistic Regression

a). Overview: Logistic Regression is a fundamental classification algorithm in machine learning. It is particularly effective for binary classification tasks.

b)Working Principle: This method models the probability of a binary outcome using a logistic function. It estimates the relationship between the dependent binary variable and one or more independent variables.

c) Application in Our Project: We utilized Logistic Regression for its efficiency and simplicity in predicting binary outcomes, such as whether a customer will subscribe to a term deposit.

d) Implementation: In our project, we applied Logistic Regression with an L2 penalty and a regularization strength (C) of 1.0. We set the maximum number of iterations to 1000 to ensure convergence.



--LOGISTIC REGRESSION--

```
# Logistic Regression model
lr = LogisticRegression(penalty='l2', C=1.0, max_iter=1000)

# Train the model
lr.fit(X_train, y_train)
```

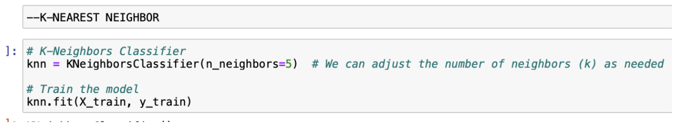
4.1.3 K-Nearest Neighbors (KNN)

a) Overview: KNN is a non-parametric, instance-based learning algorithm widely used for classification and regression.

b) Working Principle: It classifies a data point based on how its neighbors are classified. The algorithm looks at the K-nearest neighbors (where K is a specified number) and assigns the class based on the majority class of these neighbors.

c) Rationale for Use: We chose KNN for its intuitive nature and ability to make predictions based on the similarity of data points, offering a more nuanced understanding of customer behavior.

d) We utilized the K-Neighbors Classifier with n_neighbors set to 5, allowing the algorithm to consider the five nearest data points for classifying a new instance.



--K-NEAREST NEIGHBOR

```
# K-Neighbors Classifier
knn = KNeighborsClassifier(n_neighbors=5)  # We can adjust the number of neighbors (k) as needed

# Train the model
knn.fit(X_train, y_train)
```

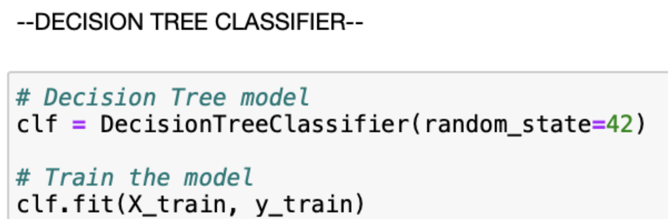
4.1.4 Decision Trees

a). Introduction: Decision Trees are versatile algorithms used for both classification and regression tasks.

b) Functionality: They model decisions and their possible consequences as a tree-like structure, where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label or decision.

c) Advantages in Our Context: In our project, Decision Trees were valuable due to their interpretability. They provide clear visualizations of the decision-making process, helping us understand the factors influencing customer decisions.

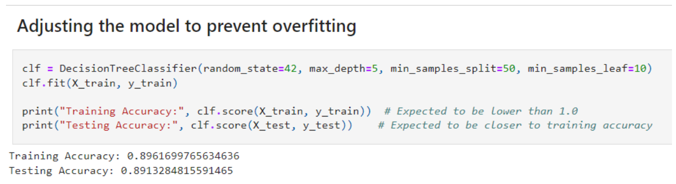
d) Setup: The Decision Tree model was implemented with a specified random_state for reproducibility.



--DECISION TREE CLASSIFIER--

```
# Decision Tree model
clf = DecisionTreeClassifier(random_state=42)

# Train the model
clf.fit(X_train, y_train)
```

The model's score was 1. It implies overfitting. To overcome this it is needed to use pruning method.



Adjusting the model to prevent overfitting

```
clf = DecisionTreeClassifier(random_state=42, max_depth=5, min_samples_split=50, min_samples_leaf=10)
clf.fit(X_train, y_train)

print("Training Accuracy:", clf.score(X_train, y_train))  # Expected to be lower than 1.0
print("Testing Accuracy:", clf.score(X_test, y_test))    # Expected to be closer to training accuracy
```

```
Training Accuracy: 0.8961699765634636
Testing Accuracy: 0.8913284815591465
```

These three algorithms work together to provide the framework for our predictive analysis. Logistic Regression delivers simplicity and effectiveness for binary outcomes, and Decision Trees provide depth with its accessible and visual decision-making structure.

By combining these methods, we may approach our prediction problem from multiple perspectives, which improves the flexibility and accuracy of our findings.

4.2. Performance measurement

In our project, we utilized a combination of metrics to evaluate the performance of each machine learning algorithm: Logistic Regression, K-Nearest Neighbors (KNN), and Decision Trees. These metrics provided a comprehensive view of each model's effectiveness in predicting the outcomes of the bank's marketing campaigns.

4.2.1 Accuracy Score

a). Method: We calculated the accuracy score for both the training and testing datasets.

b) Application: This was done using the '.score()' method on each model, which provided insights into how well the model performs in terms of correctly predicting outcomes.

c) Significance: The accuracy score, representing the ratio of correctly predicted observations to the total observations, was crucial in assessing the overall effectiveness of each algorithm.

4.2.2 Confusion Matrix

a) Construction: For each model, we created a confusion matrix using the 'confusion_matrix' function from scikit-learn.

b) Application: This was achieved by comparing the actual test outcomes ('y_test') against the predictions made by the model ('y_pred').

c) Utility: The confusion matrix helped us understand not just the errors of the models, but also the type of errors – whether they were false positives or false negatives.

4.2.3 Classification Report

a) Generation: We used the 'classification_report' function from scikit-learn to generate a detailed classification report for each model.

b) Content: These reports provided key metrics such as precision, recall, and F1-score for each class.

- *Precision:* Precision indicates the proportion of positive identifications that were actually correct. It is calculated as the number of true positives divided by the sum of true positives and false positives.

- *Recall (Sensitivity):* Recall measures the proportion of actual positives that were identified correctly. It is calculated as the number of true positives divided by the sum of true positives and false negatives.

- *F1-Score:* The F1-Score is the harmonic mean of precision and recall, providing a balance between these two metrics. It is particularly useful when the class distribution is imbalanced. The F1-Score reaches its best value at 1 (perfect precision and recall) and worst at 0.

- *Support:* This represents the number of actual occurrences of each class in the specified dataset. It is useful for assessing the evaluation metrics in the context of class imbalance.

These metrics collectively offer a comprehensive view of a model's performance, beyond mere accuracy, by considering both the precision of predictions and the model's ability to identify all relevant cases.

c) Relevance: The classification report was particularly useful for evaluating the performance of our models in a more nuanced manner. Precision and recall are critical in scenarios where the cost of false positives or false negatives varies, and the F1-score provides a balance between precision and recall.

By applying these performance measurement techniques uniformly across all three algorithms, we were able to conduct a fair and comprehensive evaluation. The combination of accuracy scores, confusion matrices, and classification reports allowed us to not only assess the overall prediction accuracy of each model but also to delve deeper into the specifics of their predictive capabilities. This multi-metric approach was instrumental in identifying the strengths and weaknesses of each algorithm, guiding our decision-making process for model selection and refinement.

4.2 Understanding Duration

4.2.1 Polynomial Regression

To improve our prediction of the 'duration' variable, which represents the last contact duration in seconds, we implemented Polynomial Regression. This approach was chosen to capture the non-linear relationships between the selected features and the target variable.

Implementation Steps:

- Feature Selection: Selected features included 'contact', 'month', 'day', 'campaign', 'age', 'job', 'balance', and 'housing', capturing a mix of categorical and numerical data.

- Encoding and Scaling: Categorical features were encoded using OneHotEncoder, and numerical features were passed through without transformation to maintain their scale for polynomial expansion.

Polynomial Features: We configured our model to use second-degree polynomial-features to explore quadratic relationships in the data.

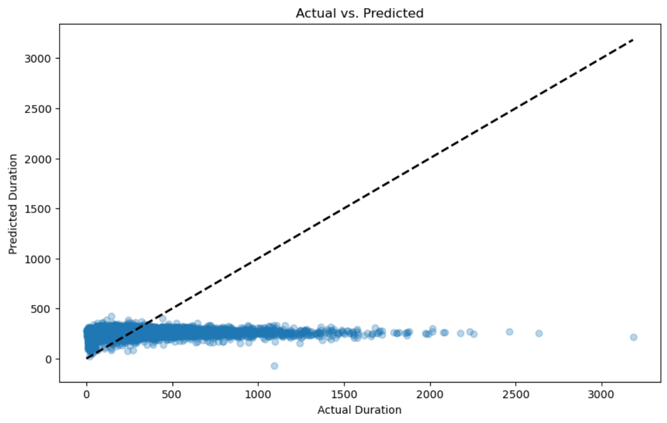
- Pipeline Setup: A pipeline was created combining a preprocessor for encoding and scaling, a polynomial feature transformer, and a linear regression model to predict the duration.

Results / Performance Metrics:

Mean Squared Error (MSE): Calculated to quantify the average squared difference

between the estimated values and the actual value, providing a clear measure of model accuracy.

- R-squared: Used to determine the proportion of variance in the dependent variable that is predictable from the independent variables.

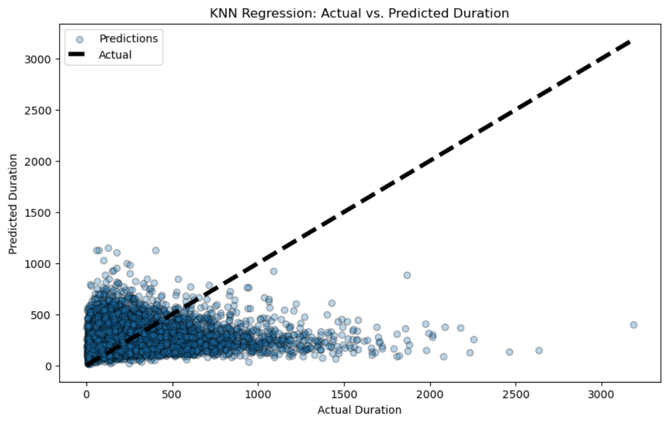


4.2.3 Polynomial Regression

To enhance our model's capability to predict 'duration', we implemented the K-Nearest Neighbors (KNN) regression. This model was chosen for its effectiveness in handling non-linear data and its ability to make predictions based on the 'neighborhood' of similar data points.

Identified and categorized features into numerical and categorical. Numerical columns included 'age', 'balance', 'day', 'campaign', and 'pdays'; categorical columns included 'job', 'marital', 'education', 'contact', 'month', 'housing', and 'loan'.

KNN Setup: Configured the KNN regressor with n_neighbors=5, determining the number of nearest neighbors to consult in making predictions.



4.3.Experiments

4.3.1 Experiments we set up for Logistic Regression:

Original Setup and Variations

a) Original Configuration:

- We began with a standard Logistic Regression model, employing an L2 penalty, a regularization strength (‘C‘) of 1.0, and the 'liblinear' solver. The model was trained on our training dataset and achieved an accuracy of approximately 88.74% on the test set.

- This initial setup served as our baseline for comparison with subsequent experiments.

b) Parameter Tuning Experiments:

- To explore the impact of different configurations on model performance, we experimented with varying the ‘C‘ value and the penalty type.

- *Varying ‘C‘ Values:* The ‘C‘ parameter, which controls the strength of regularization, was varied among 0.01 (strong regularization), 1.0 (moderate regularization), and 10 (weak regularization).

- *Penalty Type:* We tested both L1 and L2 penalties. The L1 penalty can lead to sparser solutions, potentially beneficial for feature selection, while the L2 penalty is known for its stability, especially with correlated features.

c) Analysis

- *Baseline Accuracy:*

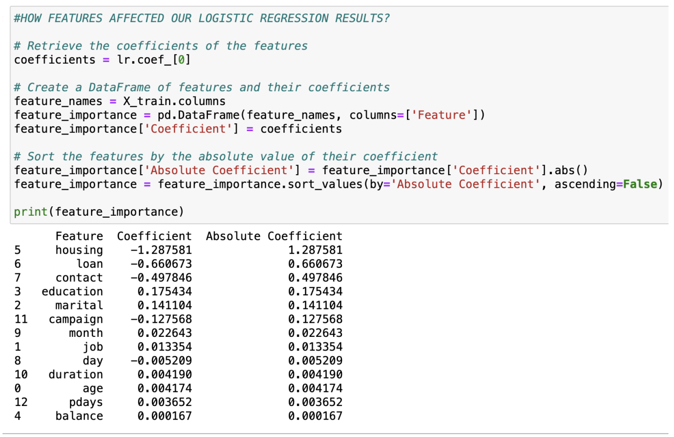
The original Logistic Regression model, set with an L2 penalty and a 'C' value of 1.0, achieved an accuracy of approximately 88.74%. This served as our benchmark for comparison.

- *Variations in Regularization and Penalty:*

-We tested different configurations by varying 'C' values (0.01, 1.0, 10) and penalties (L1, L2).

- Accuracy scores ranged from about 88.54% to 88.78%, with the highest being 88.78% under specific configurations.

- *Finding Out The Coefficients Of The Features:*



```
#HOW FEATURES AFFECTED OUR LOGISTIC REGRESSION RESULTS?

# Retrieve the coefficients of the features
coefficients = lr.coef_[0]

# Create a DataFrame of features and their coefficients
feature_names = X_train.columns
feature_importance = pd.DataFrame(feature_names, columns=['Feature'])
feature_importance['Coefficient'] = coefficients

# Sort the features by the absolute value of their coefficient
feature_importance['Absolute Coefficient'] = feature_importance['Coefficient'].abs()
feature_importance = feature_importance.sort_values(by='Absolute Coefficient', ascending=False)

print(feature_importance)
```

```
      Feature  Coefficient  Absolute Coefficient
5     housing    -1.287581              1.287581
6        loan    -0.660673              0.660673
7     contact    -0.497846              0.497846
3   education     0.175434              0.175434
2     marital     0.141104              0.141104
11   campaign    -0.127568              0.127568
9       month     0.022643              0.022643
1         job     0.013354              0.013354
8         day    -0.005209              0.005209
10   duration     0.004190              0.004190
0         age     0.004174              0.004174
12      pdays     0.003652              0.003652
4     balance     0.000167              0.000167
```

- The coefficients indicate the direction of the relationship with the target variable: positive coefficients imply a positive correlation, while negative coefficients suggest a negative correlation. The "Absolute Coefficient" highlights feature importance, with higher absolute values indicating a stronger impact on predictions. For instance, a negative coefficient of approximately -1.29 for "housing" implies that clients with housing are less likely to meet the target condition than those without.

Comparative Insights

- *Influence of 'C' Value:*

- The change in the 'C' value, which adjusts the strength of regularization, had a noticeable but not drastic effect on the model's accuracy.

- Lower 'C' (stronger regularization) did not significantly improve the model, suggesting that our dataset might not be highly prone to overfitting.

- *Effect of Penalty Type:*

- The L1 penalty slightly outperformed the L2 penalty in some configurations, particularly at 'C' values of 1.0 and 10.

- This suggests that the L1 penalty, which can simplify the model by reducing certain feature coefficients to zero, might be more suitable for this dataset.

- *Stability Across Configurations:*

- The relatively small variation in accuracy across different setups indicates that the Logistic Regression model is quite stable for this particular dataset.

- It suggests that the dataset characteristics and feature relevance are well-captured by the Logistic Regression algorithm, regardless of minor tuning.

Concluding Thoughts

The detailed analysis of the Logistic Regression experiments reveals insights into the impact of regularization strength and penalty type on model performance. The highest accuracy of 88.78% indicates a well-performing model, but the marginal differences between configurations highlight that the choice of 'C' value and penalty type, while important, should be considered alongside other aspects of model development and feature engineering. This experiment underscores the nuanced balance in parameter tuning for achieving optimal model performance.

4.3.3 Experiments We Set Up For Decision Tree

Decision Trees are an extremely understandable and flexible method that can be used for both regression and classification applications, and we used them in our supervised learning project.

a) Standard Decision Tree Model:

- Initially, we employed a basic Decision Tree model with a set random state to ensure reproducibility.

- While the model achieved perfect accuracy on the training set, it scored around 89.60% accuracy on the test set, indicating its effective learning capability but also suggesting some overfitting.

b) Decision Tree with Limited Features and Depth:

- In a subsequent experiment, we refined our approach by limiting the model to use only two features: 'age' and 'balance'. Simultaneously, we restricted the tree's depth to 3.

- This dual modification aimed to reduce the model's complexity, potentially mitigating overfitting, and to focus on the predictive strength of these specific features.

c) Results and Insights

- Performance of Full-Feature Model: The high accuracy on the training set and a strong but lesser accuracy on the test set indicated the model's effectiveness in learning but also raised concerns regarding its generalizability.

- Impact of Feature and Depth Restriction: The modified Decision Tree, utilizing only 'age' and 'balance' with limited depth, achieved an accuracy of about 82.70%.

This outcome highlights the role of other features in the dataset and demonstrates how controlling the depth can influence the model's balance between learning complex patterns and generalizing well to new data.

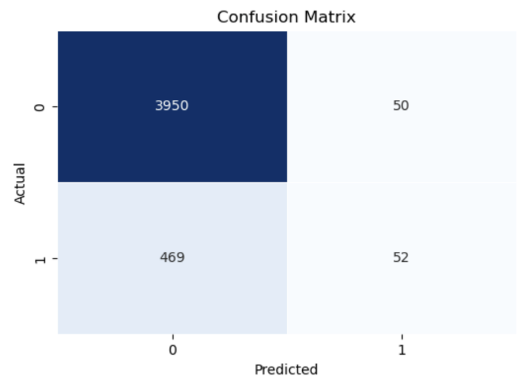
Conclusion

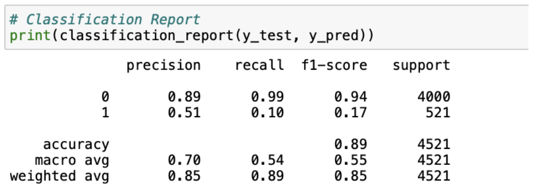
Through our Decision Tree experiments, we gained insights into the effects of feature selection and model complexity on performance. These findings emphasize the delicate balance in machine learning between capturing sufficient detail from the training data and maintaining the model's ability to generalize. The adaptability of Decision Trees to such modifications underscores their value in our array of machine learning techniques.

4.4.Results

Results of Machine Learning Algorithms We Applied:

Logistic Regression Initial Results:

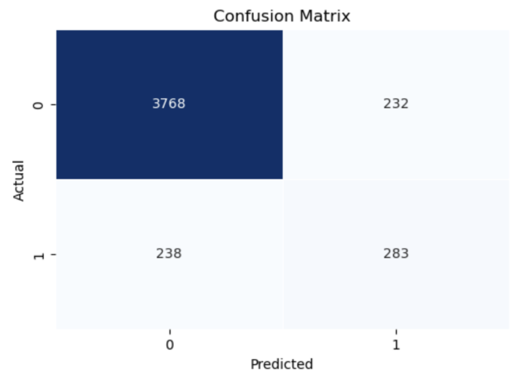


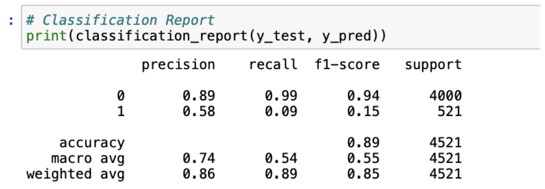


```
# Classification Report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.99      0.94      4000
           1       0.51      0.10      0.17       521

    accuracy                           0.89      4521
   macro avg       0.70      0.54      0.55      4521
weighted avg       0.85      0.89      0.85      4521
```

- Decision Tree Initial Results:





```
# Classification Report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.89      0.99      0.94      4000
           1       0.58      0.09      0.15       521

    accuracy                           0.89      4521
   macro avg       0.74      0.54      0.55      4521
weighted avg       0.86      0.89      0.85      4521
```

5. INSIGHTS

When creating new marketing initiatives, banks can benefit greatly from the application of machine learning (ML) techniques. ML can forecast customer reactions to upcoming efforts by examining past data and customer profiles, enabling more focused and effective marketing campaigns. Increased client happiness, lower marketing expenses, and higher conversion rates are the results of this focused strategy. Furthermore, machine learning (ML) assists in recognizing new trends and shifting consumer preferences, allowing banks to proactively modify their products. In the fast-paced financial market of today, where data-driven and customized solutions are essential to preserving a competitive edge, this method is priceless.

The power and importance of characteristics like marital status can be dramatically changed by economic situations, as we must acknowledge while analyzing the "Direct Marketing Campaigns for Bank Term Deposits" dataset. For example, in a stable economic environment, being single may be connected with lower family expenses and a stronger propensity to invest. But in an economic downturn, stress and uncertainties on finances can change this assumption. Regardless of their marital status, people may start making more conservative financial decisions. This realization emphasizes how crucial it is to place our forecasting models in the context of the contemporary economy. The elements affecting consumer behavior will change along with the state of the economy.

In order to effectively modify their plans and stay relevant and responsive to the constantly shifting financial needs and realities of their clientele, banks must have a sophisticated awareness of these dynamics. This method emphasizes the significance of constant attention to detail and flexibility in predictive modeling within the banking industry, particularly in view of the uncertainties surrounding the world economy.

The way banks and customers engage changes dramatically during a recession. Banks have historically concentrated on luring clients in for investments. But during a crisis, attention shifts to examining the creditworthiness of loan applicants. The bank's customer engagement and data analysis techniques are significantly impacted by this shift in dynamics. Banks must improve their predictive models, giving financial risk assessment and possible borrower stability first priority. This calls for shifting their methods from encouraging investments to managing risks, in addition to responding to the increasingly cautious and risk-averse habits of their customer base. Bank data analysts are therefore faced with the challenge of creating increasingly complex and flexible analytical techniques. These strategies need to effectively navigate the particulars of a recession, when financial stability and client dependability become critical.

Even with advanced ML techniques at their disposal, the role of data analysts in banks extends beyond mere technical proficiency. Their social skills, sociological background, and perception are essential for modifying strategies in response to the changing circumstances of a financial crisis. Data must be interpreted by analysts taking into account changes in society and human behavior, then incorporating these observations into their models. This comprehensive method guarantees that tactics are not just data-driven but also considerate of the needs of customers, resulting in more efficient and humane banking procedures—especially important in hard economic times.

The logistic regression model's output reveals a compelling narrative about the factors influencing customer decisions regarding term deposits. The 'housing' feature, with its high negative coefficient, suggests that customers with housing loans are less likely to subscribe to term deposits, possibly due to their existing financial burdens. Similarly, 'loan' as a negative influencer indicates a reluctance among those with personal loans to make additional financial commitments. The negative coefficient for 'contact' could reflect the impact of the communication method on customer decisions. On the other hand, 'education' and 'marital' status, although having smaller positive coefficients, hint at a demographic segment more inclined towards term deposits. Interestingly, the relatively low coefficients for 'age', 'balance', and 'duration' suggest these factors, while relevant, are

less decisive in this context. This detailed analysis underscores the importance of understanding customer profiles and their financial backgrounds in tailoring banking services and marketing strategies

References

[1] A data-driven approach to predict the success of bank telemarketing (2008)