BANK MARKETING SUCCESS PREDICTION

IE7300 - STATISTICAL LEARNING FOR ENGINEERING

Group 7
Keerthana Balswamy
Saiteja Reddy Gajula
Suja Ganesh Murugan

Abstract

In this study, we evaluate a bank marketing dataset to forecast the performance of marketing initiatives using classification algorithms. Our dataset covers a variety of attributes such as age, job, degree, balance, loan status, and contact information.

We begin by performing feature engineering to convert categorical data to numerical representation. Then, we use Exploratory Data Analysis (EDA) to obtain insight into the dataset. Following EDA, we clean up the data by removing outliers and duplicates.

To prepare the data for modeling, we separated it into training and testing sets. Given the high dimensionality of our feature space, we use techniques like Principal Component Analysis (PCA) to reduce dimensionality. Furthermore, we solve class imbalance by under sampling.

We then train other classification algorithms, such as Logistic Regression and Gaussian Naive Bayes. We use essential metrics, including F1-score, accuracy, precision, and recall, to evaluate model performance. These criteria help us identify the best-performing method for our statistical study.

Introduction

In this research, we look at the direct marketing efforts run by a Portuguese banking institution that predominantly used phone calls to reach out to potential customers. These commercials attempted to promote a single financial product: bank term deposits. Unlike traditional marketing methods, these campaigns frequently necessitate repeated interactions with the same client to determine whether they would subscribe to the product. These interactions had a binary conclusion of 'yes' or 'no,' indicating whether the client had subscribed to the bank term deposit.

Our investigation focuses on determining the efficiency of these marketing activities and identifying significant elements that influence client bank term deposit subscriptions. By evaluating the dataset containing campaign information, we hope to identify insights that can inform future marketing plans and improve the effectiveness of customer outreach activities. Throughout this report, we will examine many components of marketing efforts, such as the characteristics of the target clientele, the frequency and duration of contact, and the results of these interactions. Using data-driven methodologies and statistical analysis, we aim to better grasp the dataset's underlying patterns and trends.
Finally, we want to deliver actionable insights that will assist the banking institution in enhancing its marketing tactics, increase client engagement, and maximize the success rate of future campaigns aimed at bank term deposit subscriptions.

Data Description

The dataset relates to direct marketing initiatives carried out by a Portuguese banking organization in which phone calls were the major mode of engagement with customers. The goal of these advertisements was to promote a particular financial product: bank term deposits. Each observation in the dataset corresponds to a client contacted during the marketing activities.

The dataset includes input variables that provide information about bank clients and specifics of marketing interactions, as well as an output variable that indicates whether the client subscribed to a term deposit.

Column	Type	n_unique	min	max	sample_unique
age	int64	75	18	95	[55, 39, 51, 38, 36, 41, 37, 35, 57, 23, 33, ...]
job	object	12	admin.	unknown	[admin., self-employed, services, housemaid, ...]
balance	int64	3153	-6847	66653	[1662, -3058, 3025, -87, 205, 76, 4803, 911, ...]
housing	object	2	no	yes	[no, yes]
loan	object	2	no	yes	[no, yes]
contact	object	3	cellular	unknown	[cellular, telephone, unknown]
month	object	12	apr	sep	[jun, apr, may, nov, jan, sep, feb, mar, aug, ...]
campaign	int64	32	1	63	[2, 3, 1, 4, 5, 6, 7, 30, 8, 9, 11, 14, 10, 28, ...]
pdays	int64	422	-1	854	[-1, 352, 21, 91, 186, 263, 96, 355, 294, 412, ...]
poutcome	object	4	failure	unknown	[unknown, other, failure, success]
marital	object	3	divorced	single	[married, single, divorced]
education	object	4	primary	unknown	[secondary, tertiary, primary, unknown]
default	object	2	no	yes	[no, yes]
day	int64	31	1	31	[4, 17, 7, 13, 18, 16, 15, 25, 9, 20, 30, 10, ...]
duration	int64	1310	2	3881	[94, 882, 476, 531, 176, 263, 396, 117, 613, ...]
previous	int64	32	0	58	[0, 1, 3, 4, 5, 7, 2, 6, 9, 11, 27, 8, 22, 10, ...]
deposit	object	2	no	yes	[yes, no]

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a fundamental step in the Data Analysis process, especially critical in preparing the dataset for classification tasks such as ours. EDA helps us understand the structure of data and the relationships with variables showing potential collinearity issues, which could impact model performance. It also helps us to find variables with missing values, which we handle by imputation, removal, or other methods, which we will display in our EDA process.

I. Data Structure:
   The dataset we have has **45211** rows and **17** attributes, out of which 7 are Numerical variables and 10 are Categorical.
   - ➢ Numerical Variables: age, balance, duration, campaign, pdays, previous, day
   - ➢ Categorical:
     - Nominal: Education, Month

- Ordinal: Job, Housing, Loan, Contact, POutcome, Deposit, Default, Marital Status

II. Null Values:

We found that our data contains a significant number of null variables, predominantly in Poutcome (81.75% null) and Contact (28.80% null).

```
#percentage of null values
df.isna().mean().round(4) * 100
```

```
age             0.00
job             0.64
marital         0.00
education       4.11
default         0.00
balance         0.00
housing         0.00
loan            0.00
contact        28.80
day_of_week     0.00
month           0.00
duration        0.00
campaign        0.00
pdays           0.00
previous        0.00
poutcome       81.75
y               0.00
dtype: float64
```

***Fig. 1**. Proportion of null values*

i. POutcome:

The variable **poutcome** contains information on the outcome of the previous marketing campaign (categorical: 'failure',' nonexistent',' success'). Since it has a high proportion of null values, 81.75%, imputation can significantly alter the variable's original distribution, leading to misrepresentations in our analysis. Thus, we decided to drop this feature.

ii. Contact:

The variable **contact** contains information on the communication type of the marketing campaigns (categorical: 'cellular,' 'telephone'). There are various methods of handling this missing data with imputation, but to avoid skewing the variable's distribution, we have decided to name all the missing values as a seperate category called 'unknown'.

iii. Education and Job:

The variables **education** and **job** contains information on the education level (categorical: 'primary','secondary','tertiary') and the Occupation (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown') of the customer. They both have 4.11% and 0.64% null values, respectively, since the proportions of null values are smaller, we decided to perform imputation with the help of mode. The null values of education were replaced with 'secondary' and job with 'blue-collar'.

III. Duration:

The variable **duration** is the duration of the last contact in seconds (numeric). Our dataset has mentioned in an important note that this attribute highly affects the output target (e.g., if duration=0, then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call, y is obviously known. And had concluded that, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model. Thus, we dropped this variable for our prediction problem.

IV. Data Visualization:

After handling the null values, we proceeded with visualizing the distributions of both numerical and categorical variables to better understand them. This will help us identify outliers and recognize any existing relationships or trends between two numeric variables.

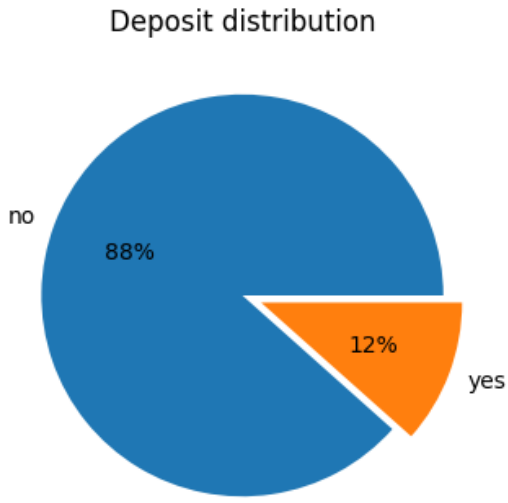

***Fig. 2**. Distribution of Term Deposit Subscription after the campaign*

Fig 2 shows that 88% of our data belongs to class '0'. To address this imbalance, we have employed techniques like Undersampling and Oversampling while training our models to prevent biased predictive outcomes.

We also performed comparative visualizations between variables such as contact, month, housing, loan and default (has credit in default?) with the target variable to understand it's influence on the customer's decision.

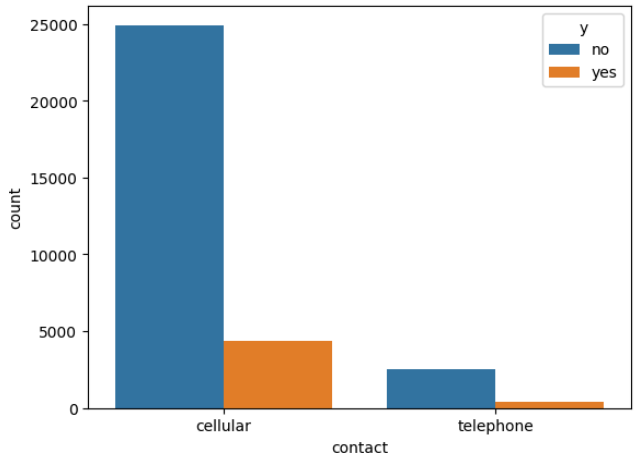

***Fig. 3**. Contact and deposit subscription status*

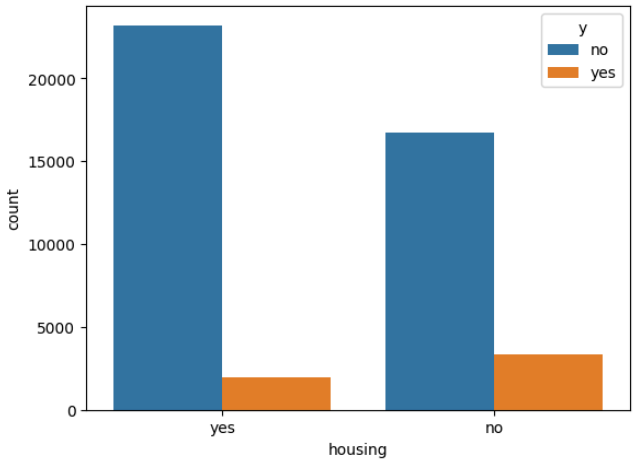

***Fig. 4**. Housing and deposit subscription status*

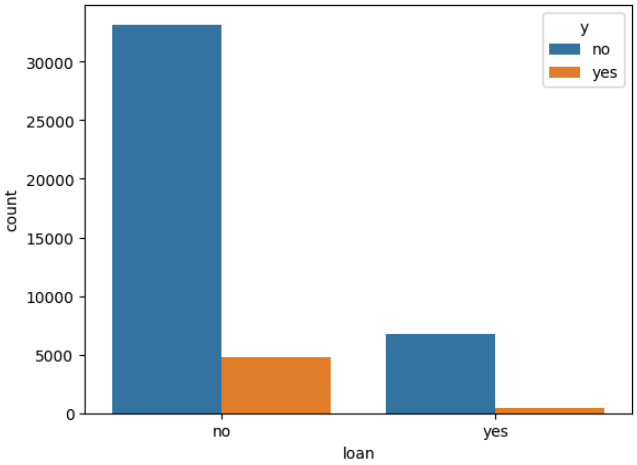

***Fig. 5**. Loan and deposit subscription status*

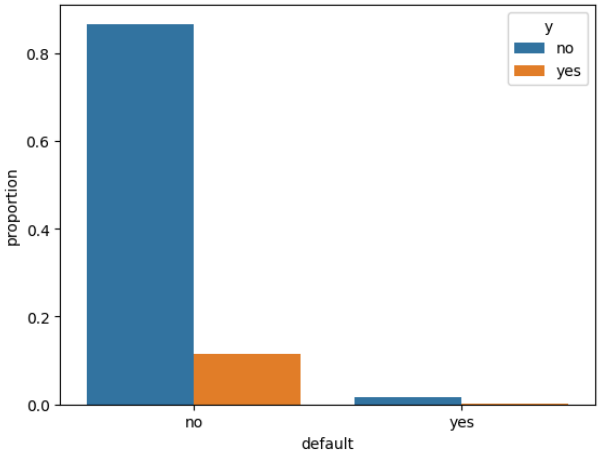

***Fig. 6**. Default and deposit subscription status*

From Fig 4, we can infer that people with housing have less tendency to subscribe to term deposit accounts, which mostly could be due to possible home debt. Fig 5 also shows similar reasoning where customers with loans are less likely to subscribe to term deposits. From Fig 6, we can understand that an overwhelming majority of customers who did not subscribe to a term deposit (y = no) are those who have no credit default (default = no). This suggests that customers without a default are significantly less likely to subscribe to a term deposit compared to those with a default.

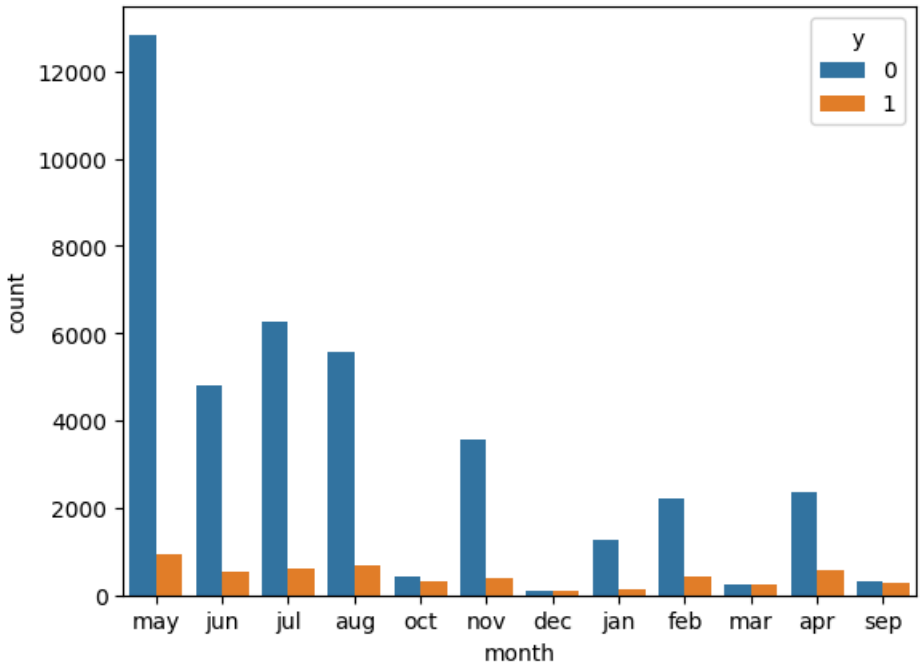

Fig. 7*. Month-wise deposit subscription status*

The majority of contacts (and thus marketing activity) seem to be concentrated in a few specific months, with May showing the highest level of activity, as shown in Fig 7. The frequency of contacts significantly drops in the months after August, which shows that the decisions to subscribe to a term deposit might be influenced by seasonal factors.

In addition to these, we also visualized categorical variables like age, education, marital status, and job to understand customer demographics.

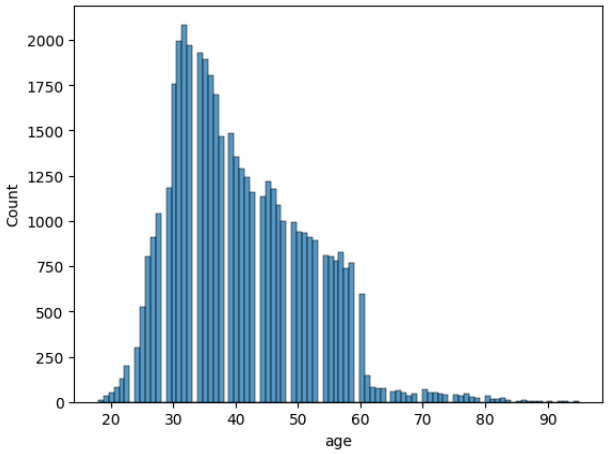

Fig. 8*. Age distribution among Customers*

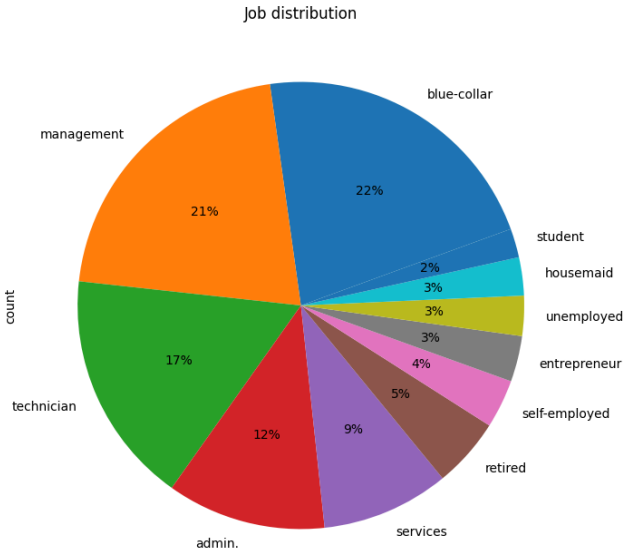

Fig. 9*. Job distribution among Customers*

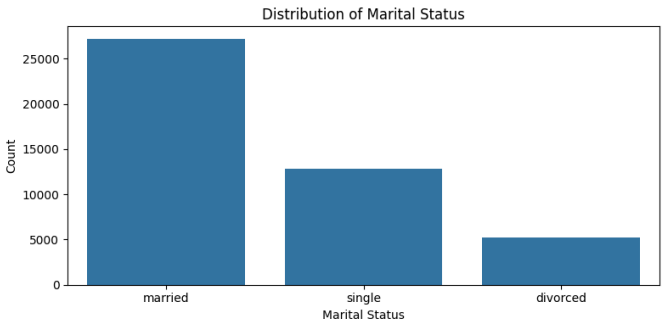

***Fig. 10**. Marital Status distribution among Customers*

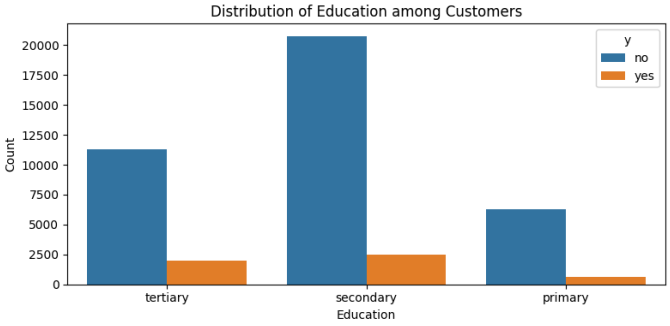

***Fig. 11.** Education level distribution among Customers*

The majority of the customers contacted by the campaign are between the ages of 30 and 60 and are married. The major education level of the customer database seems to be Secondary, with most of them in the blue-collar and management sectors.

To check if our features are independent of each other, we performed feature correlation analysis using the '.corr(method='pearson')' function and plotted a heatmap. Fig 11. shows that all the variables have no underlying correlation between them. There is a low correlation between pdays and previous but not very significant.

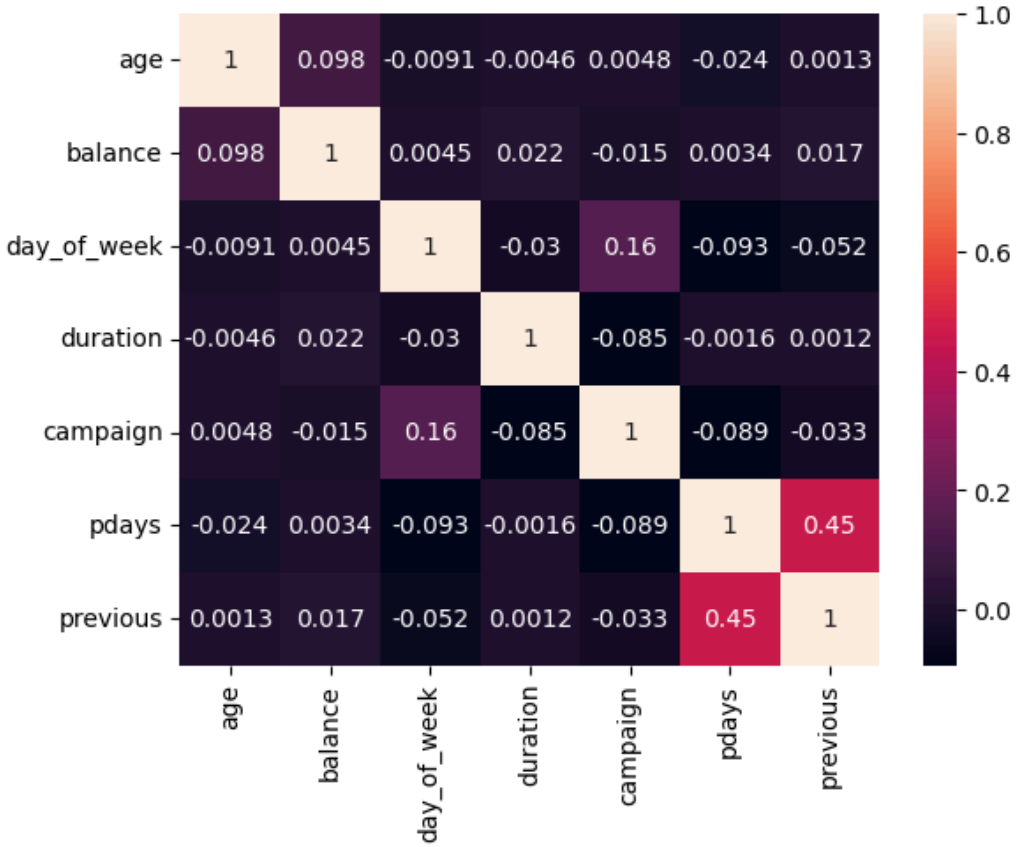

***Fig. 11.** Heatmap of correlation between numerical variables*

Feature Engineering:

Our dataset has a mixture of numerical and categorical variables. Thus, we performed data transformation on continuous numerical variables like age by binning it and further encoding all the binary categorical variables using Label encoding and multi-class variables using Target encoding and one-hot encoding and compared it's performances.

i. Outliers:

Before proceeding with the data transformation step, we noticed that some of the variables, such as campaign(number of contacts performed during this campaign(numeric)) and previous(number of contacts performed before this campaign(numeric)), had outliers with extreme values. For example, the campaign had a max value of 275, which is unusually high. Thus, we removed the outliers of these variables that lie beyond the 0.99 percentile to improve the model's performance.

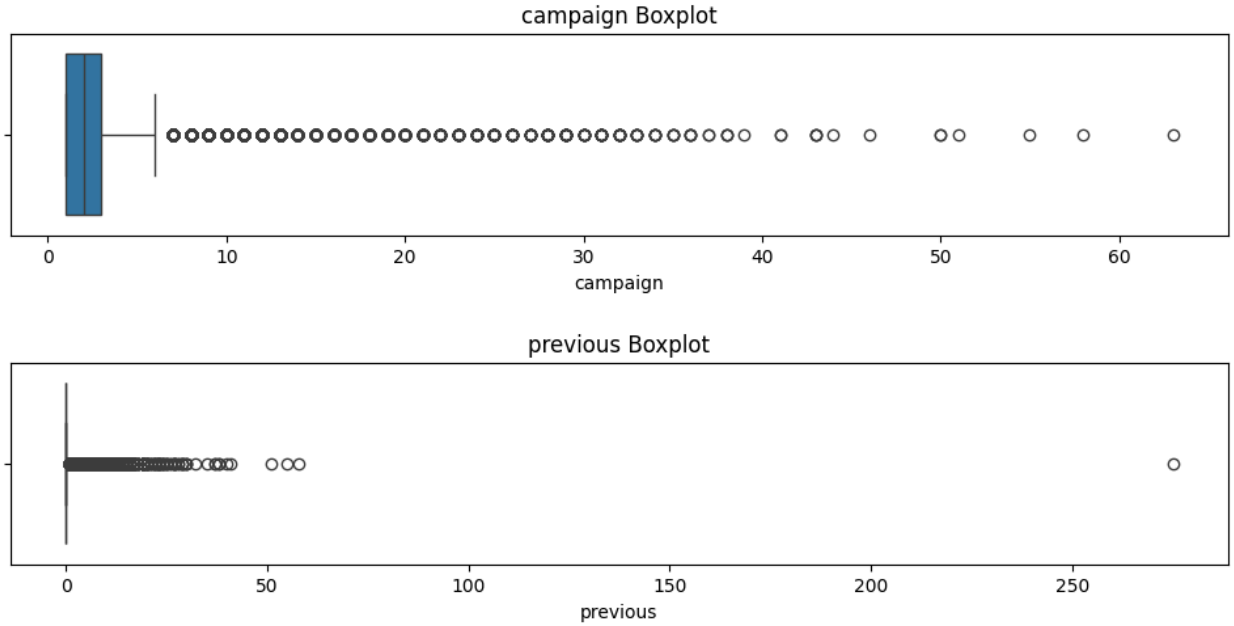

Fig. 12. *Boxplots of the variables campaign and previous*

ii. pdays:

The variable pdays has occurrences of -1, which indicates that the bank did not previously contact the customer. This value can be problematic later when classifying the dataset; thus, we replaced the values of -1 with 0.

iii. Binning:

Since the variable Age is continuous, we decided to bin it into a different column as age_categories using the following custom bin ['<20', '20-30', '30-40', '40-50', '50-60',

'60-70', '70-80', '80<'] to increase computational efficiency. This can also help to mitigate the influence of outliers in this variable by grouping extreme values together into higher or lower bins.

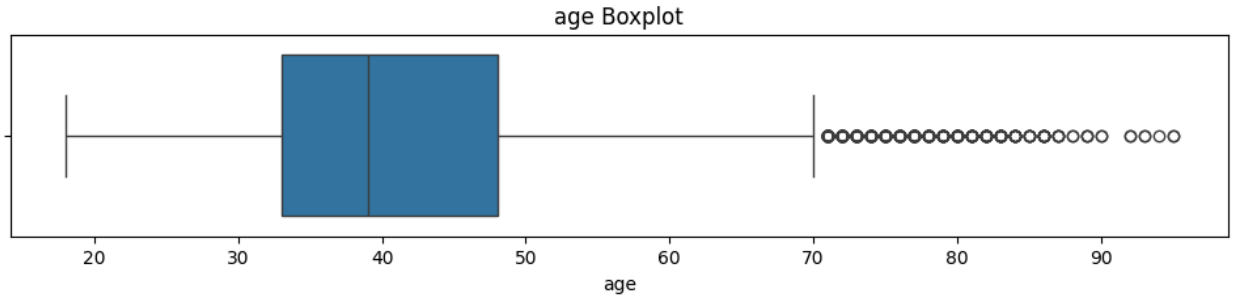

Fig. 13. *Boxplot of the variable Age*

iv. Target Encoding:

Since our categorical feature has many unique values, we decided to implement target encoding since it helps avoid the dimensionality issue that comes with one-hot encoding.

Results

MODEL 1: LOGISTIC REGRESSION

In this part of our project, we explored Logistic Regression, a powerful statistical technique used for binary classification. We used a combination of our own implementation and the popular scikit-learn library to study predictive analytics. We also investigated how standardization and Principal Component Analysis (PCA) affect the model's performance.

Model Foundation:

We developed our model by creating a logistic regression algorithm from scratch. The foundation of our model is the logistic function:

$$P(y \mid \theta, x) = 1 / (1 + e^{(-\theta^T x)})$$

For the task of binary classification, this function predicts the probability that a certain input x belongs to the positive class y=1, based on the weights θ.

Pre-processing:
Before training our model, we carefully processed the data to ensure accuracy. We standardized the data to prevent bias from feature scales and used SMOTE to address the class imbalance, improving the diversity of our training dataset for a fairer learning experience.

Training and Tuning and Validation:
We fine-tuned our algorithm using Gradient Descent, closely monitoring the cost function:

$$J(\theta)=-[y\log(h\theta(x)) +(1-y)\log(1-h\theta(x))]$$

We utilized both L1 and L2 regularization techniques for cost reduction, thus enhancing the sophistication of our model.
After the training, we evaluated the model's performance using precision. We used confusion matrices and AUC scores to gauge the model's performance. We have also deployed a function called 'find_best_threshold,' which determines the threshold that maximizes the F1 score on the validation set.

PCA- Enhanced Logistic Regression:
The implementation of PCA simplified the data, allowing us to better understand our features and make our model more efficient. PCA helped us identify the key factors influencing our predictions, highlighting the importance of our feature selection process.

Without PCA:

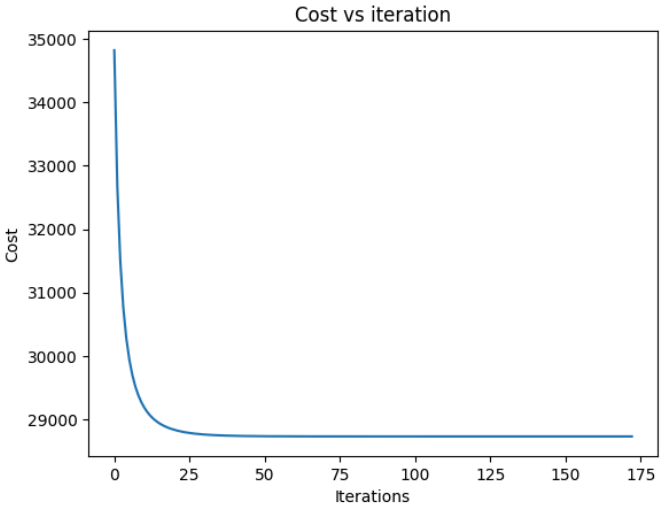

***Fig. 13**. Training data cost trend*

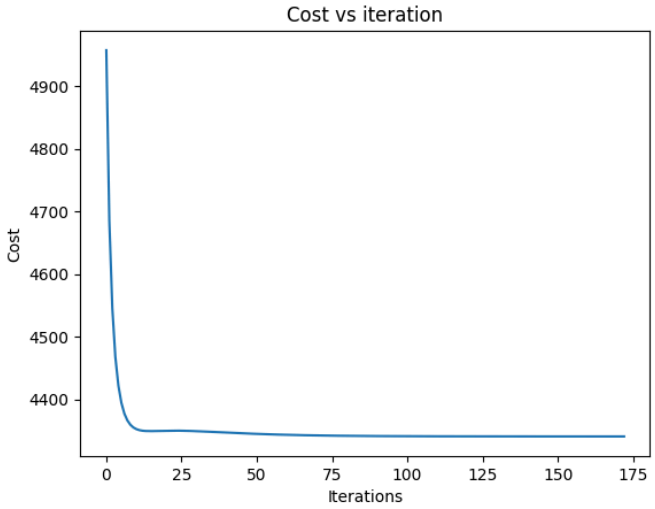

***Fig. 14**. Validation data cost trend*

The model finished learning with the Best threshold at 0.6930303030303. The cost trend from Training and validation shows that there was no overfitting issue.

Metrics	
Accuracy	0.845495
Error rate	0.154505
Sensitivity/Recall/True Positive Rate	0.415385
False Negative Rate	0.584615
Specificity/True Negative Rate	0.902551
F1 score	0.568929

Fig. 15. *LR without PCA model performance metrics*

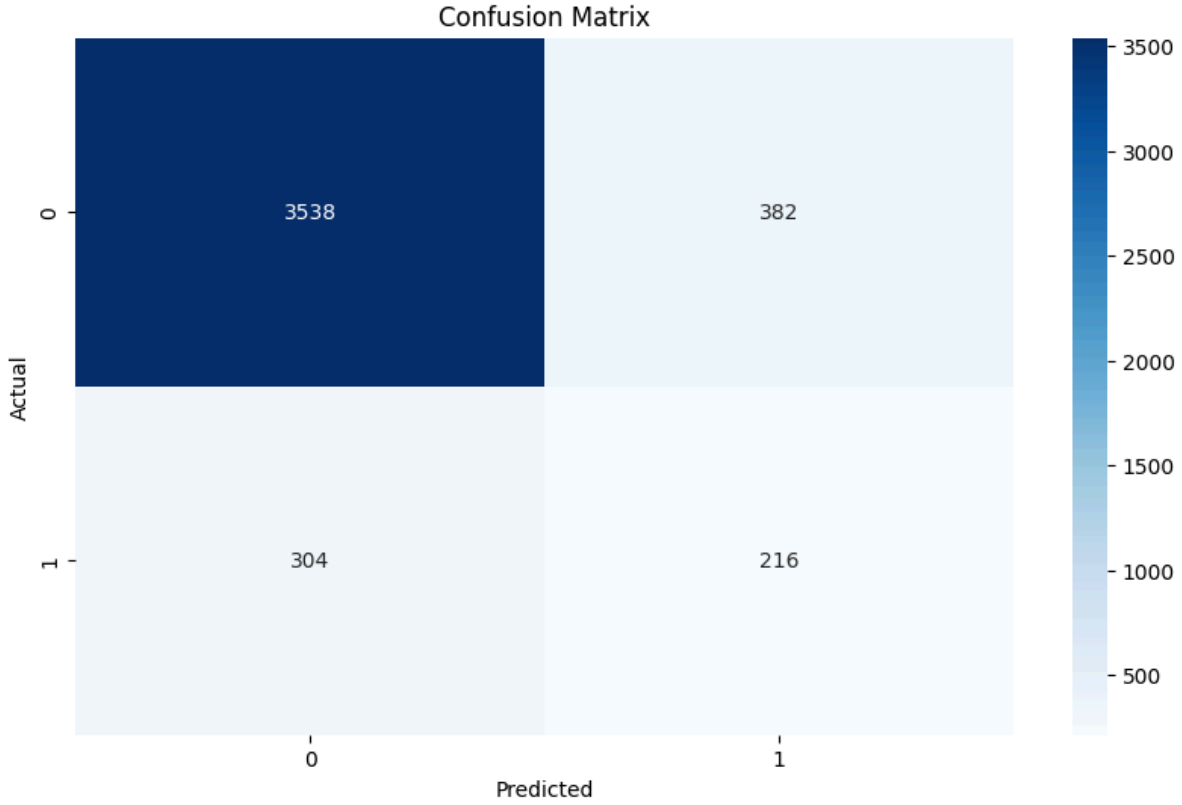

Fig. 16. *LR without PCA confusion matrix*

With PCA:

Metrics	
Accuracy	0.744595
Error rate	0.255405
Sensitivity/Recall/True Positive Rate	0.563462
False Negative Rate	0.436538
Specificity/True Negative Rate	0.768622
F1 score	0.650243

***Fig. 17**. LR with PCA model performance metrics*

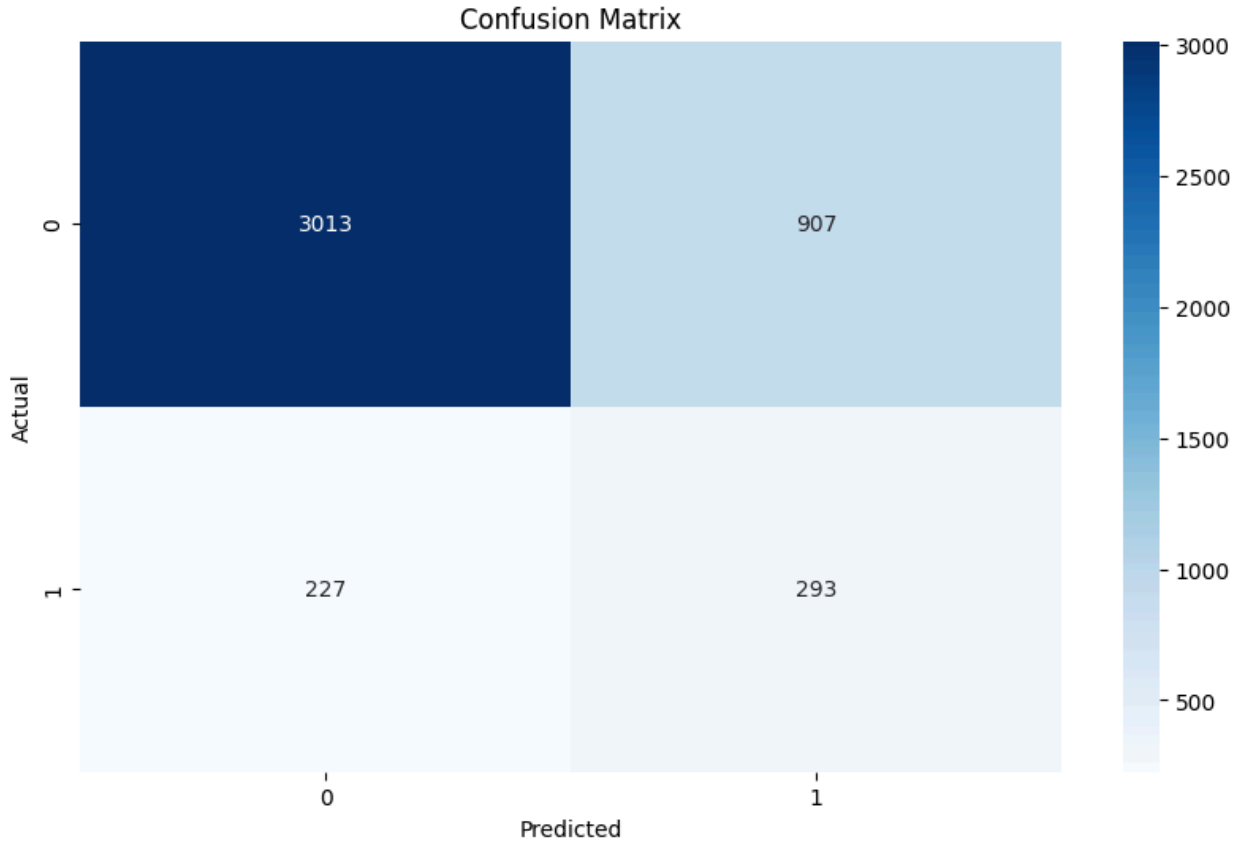

***Fig. 18**. LR with PCA Confusion matrix*

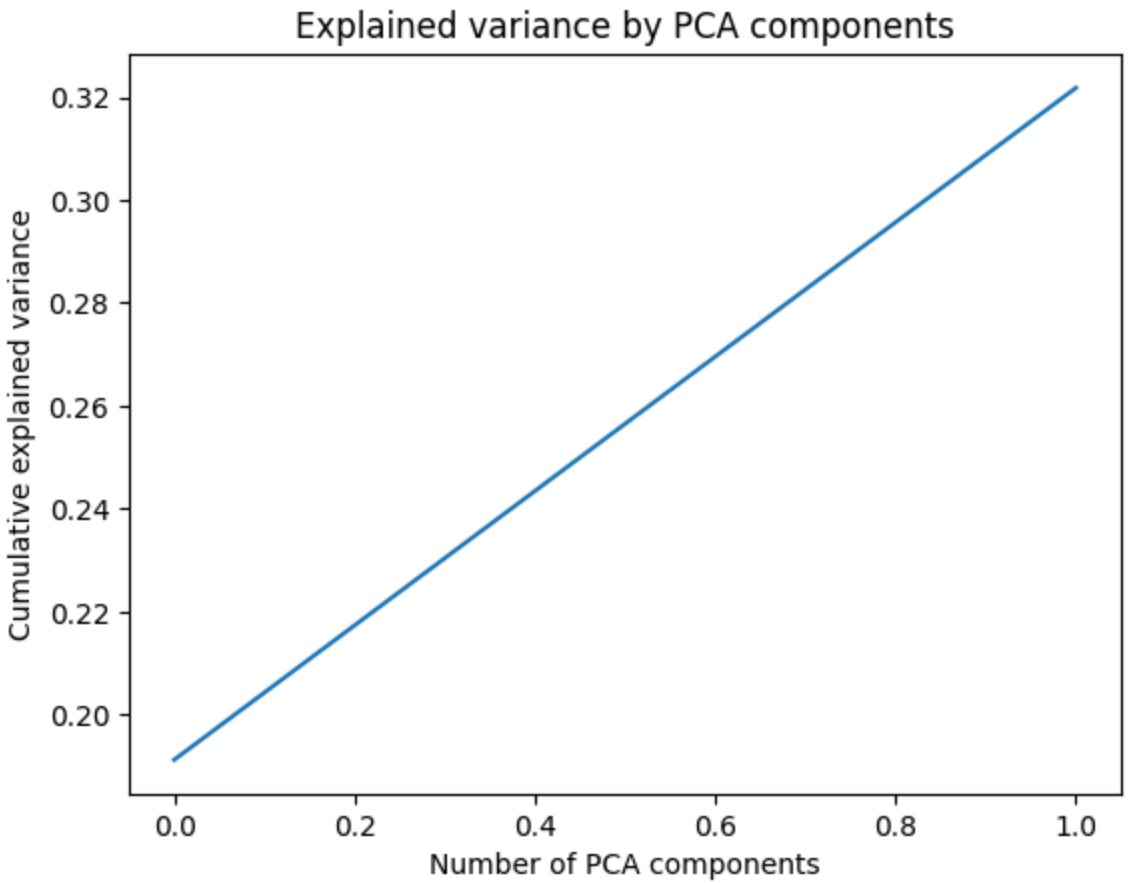

***Fig. 19**. LR with PCA Confusion matrix*

The Logistic Regression model performed well without PCA, indicating that models will indeed perform better with the rich, original feature set rather than a reduced one. PCA reduces dimensionality by projecting the data onto fewer dimensions, which can lead to a loss of information. If the discarded components contain information that is important for prediction accuracy, the model's performance can suffer.

MODEL 2: Gaussian Naïve Bayes

This section introduces the Gaussian Naive Bayes (NB) algorithm, a probabilistic model commonly used for classification applications. NB is a generative model that makes predictions based on assumptions about the data's underlying distribution and Bayes' theorem. While implementing Naive Bayes we have considered that the dataset has an Independent Identical Distribution, and the numerical features follow the Gaussian distribution. The following are significant points and explanations for the Gaussian Naive Bayes algorithm:

Model Foundation:

Gaussian Naive Bayes assumes that the characteristics in each class follow a Gaussian (normal) distribution. This means that for each class, the chance of detecting a feature value is estimated using the probability density function (pdf) of a Gaussian distribution with a specific mean and variance.

The Gaussian pdf is given as:

$$f(x_i|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

x_i is the value of variable x.

μ is the mean of the distribution.

σ^2 is the variance of the distribution.

Pre-processing:

Before training the model, we preprocess the data to guarantee that it is consistent with the Gaussian distribution assumption. This may entail normalizing the characteristics to have a mean of zero and a standard deviation of 1.

Training and tuning:

During the training procedure, the parameters of the Gaussian distribution are estimated for each feature in each class. This includes determining the mean and standard deviation for each feature

within each class. These parameters are used in prediction to calculate the probability of seeing a specific feature value given the class.

Visualization and validation:

Following training, we assess the model's performance using various metrics, including accuracy, precision, and recall. Visualization approaches, such as confusion matrices, can provide information on the model's behavior and effectiveness.

One distinguishing feature of Gaussian naïve Bayes is the naïve assumption of feature independence within each class. This means that the presence of one feature has no bearing on the presence of another, given the class label. While this assumption may not always be valid in practice, Gaussian NB frequently performs well even when it is broken.

Overall, Gaussian Naive Bayes provides a simple but successful technique for classification tasks, especially when the data follows the Gaussian distribution assumption. Understanding the principles and assumptions of this adaptable method allows us to create accurate predictions and obtain insights into the underlying data distribution.

Metrics	
Accuracy	0.644820
Error rate	0.355180
Sensitivity/Recall/True Positive Rate	0.717308
False Negative Rate	0.282692
Specificity/True Negative Rate	0.635204
F1 score	0.321136

***Fig. 20**. Naive Bayes with PCA model performance metrics*

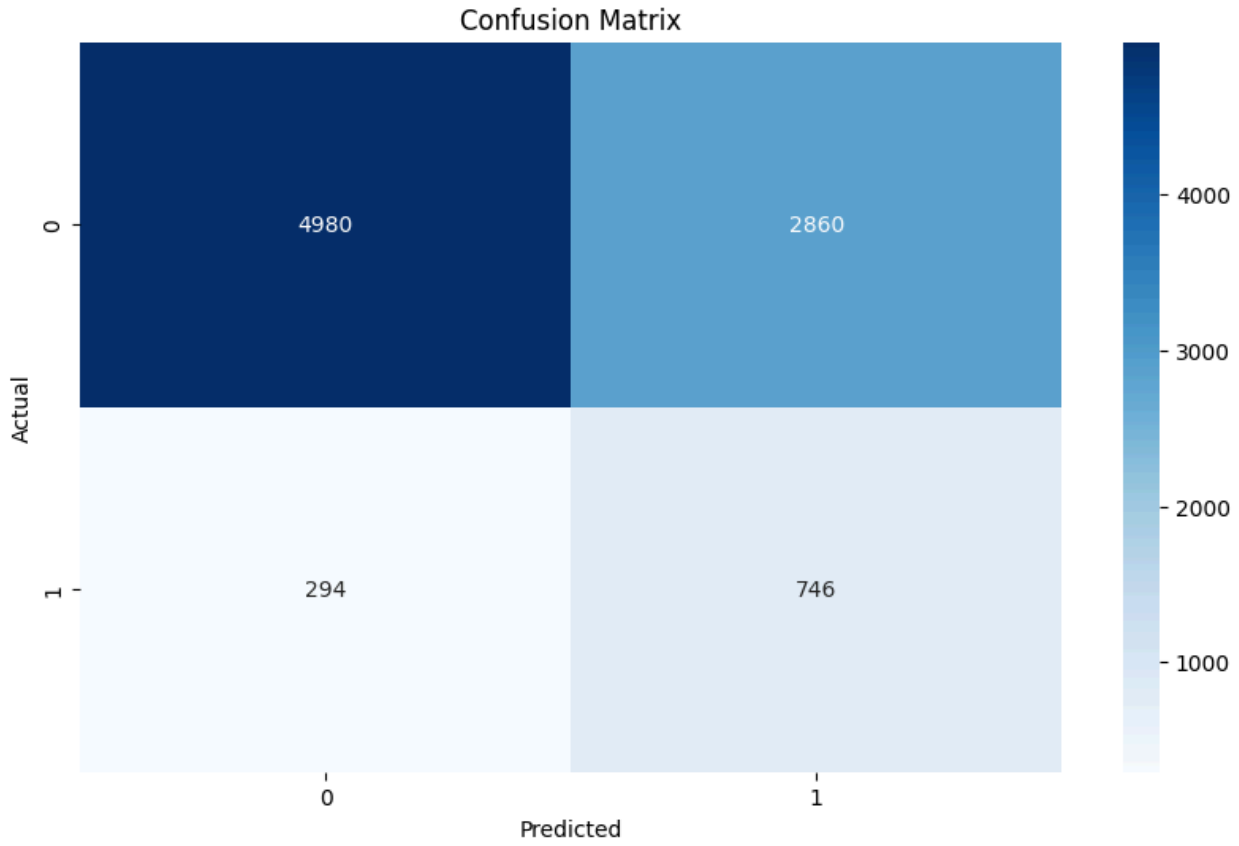

Fig. 21. Naive Bayes with PCA Confusion matrix

Summary of Models

Model	Accuracy	Error rate	F1 score
LR	0.845	0.15	0.56
LR with PCA	0.74	0.25	0.65
Naive Bayes	0.64	0.1235	0.32

Conclusion

In our investigation into the efficacy of various classification models for predicting customer subscription to term deposits in bank marketing campaigns, we have employed a range of techniques to address challenges inherent in the dataset. These challenges include imbalanced classes, outliers, and the need for feature selection and transformation.

We observed that models without Principal Component Analysis (PCA) outperformed those with PCA, likely due to the information loss that PCA introduces. This underscores the importance of the original features in predicting the target variable, which may contain subtle yet critical patterns that PCA can obscure.

Experimenting with different machine learning models, such as Naive Bayes and Logistic Regression, has provided diverse perspectives on the dataset:

- Naive Bayes: Known for its simplicity and speed, this model assumes feature independence and can quickly deliver baseline results. It is particularly useful when the dataset is large, though its assumption of feature independence can be a significant limitation when this condition is not met.
- Logistic Regression: A straightforward and interpretable model, logistic regression performs well when the relationship between the features and the log odds of the outcomes is linear. Its interpretability is a strong suit, though its performance may falter with non-linear relationships unless feature engineering is employed to capture these complexities.

Overall, each model brings its strengths and weaknesses. Our exploration into the classification of bank marketing campaign data demonstrates the nuanced trade-offs in model selection and the critical role of data preprocessing. Future work should aim to refine these approaches and explore cutting-edge techniques to enhance model accuracy and interpretability further.