COMP615 – Foundations of Data Science
ASSIGNMENT TWO
Group assignment
Semester 1, 2024

Bank Marketing
Dataset Analysis Report

Group 15	
Student ID	Student Names
22185652	Alexander Smokina
21156028	Min Thiha Ko Ko

PAPER CODE: COMP615

Due Date: 2nd Jun 2024 (midnight NZ time)

TOTAL MARKS: 100

INSTRUCTIONS:

1. **The following actions may be deemed to constitute a breach of the General Academic Regulations Part 7: Academic Discipline,**
   - Communicating with or collaborating with another person regarding the Assignment
   - Copying from any other student work for your Assignment
   - Copying from any third-party websites unless it is an open book Assignment
   - Uses any other unfair means
2. **Please email DCT.EXAM@AUT.AC.NZ if you have any technical issues with your Assessment/Assignment/Test submission on Canvas immediately**
3. **Attach your code for all the datasets in the appendix section**.

Table of Contents

List of Tables

List of Figures

1. Introduction

This report analyses a dataset concerning direct bank marketing campaigns based on phone calls conducted by Portuguese banks. The dataset includes 17 attributes, with the primary outcome being whether a client subscribes to a term deposit (yes/no). The goal is to use machine learning techniques to predict the outcome of these marketing campaigns.

1.1 Objective

The main objective of this research is to analyse the Bank Marketing Data Set to identify significant factors that influence the success of marketing campaigns. Through data exploration and modelling, this study aims to:

- Understand the key factors influencing client subscription to term deposits.
- Develop and evaluate classification models using K-Nearest Neighbours (KNN) and Naïve Bayes (NB) algorithms to predict campaign outcomes.
- Explore and optimise the architecture of Artificial Neural Networks (ANN) to improve prediction accuracy.

1.2 Key Questions

This study will focus on three main aspects:

- **Data Exploration and Feature Analysis:** What are the key characteristics and influential features in the dataset that impact the campaign outcomes?
- **Model Performance:** How do the KNN and Naïve Bayes models perform in predicting the success of the marketing campaigns? Compare their performance?
- **Neural Network Optimisation:** How does the architecture of an Artificial Neural Network (ANN), specifically the number of neurons and layers, affect the classification accuracy? How does the performance of the ANN compare to KNN and NB models?

Overall, the report aims to provide a comprehensive analysis of the factors influencing the success of bank marketing campaigns and to establish reliable models for predicting campaign outcomes using various machine learning techniques.

2. Part A: Predicting Bank Marketing Campaign Outcomes

2.1 KNN vs Naïve Bayes Algorithms

K-Nearest Neighbours (KNN) is a non-parametric, lazy learning algorithm used for classification and regression. It is a simple (little training involved), yet effective machine learning technique that help us to find the closest examples in the data to the new point we want to classify. Essentially, KNN assumes that similar things are usually found close together. This means it looks at the 'k' closest data points (neighbours) to figure out the category of a new point. To classify a new data point, KNN measures how far this point is from others in the dataset, picks the closest 'k' points, and then gives the new point the most common category among these neighbours. The choice of 'k' and how distance between data is measured (often using Euclidean distance) greatly affects how well the algorithm works and how it handles different kinds of data.

Meanwhile**, Naïve Bayes** is a fast and efficient classifier based on Bayes' Theorem and assumes that features are independent. It calculates the likelihood of each possible category of the target outcome based on given features, and assigns the most likely category to each new observation. Although this method is simple, the assumption of feature independence might not always hold true, Naïve Bayes can still be highly effective. This algorithm works fast and can handle large datasets with many features efficiently because it considers each feature separately when determining the likelihood of each category. However, when many features of the dataset are dependent, accuracy may significantly decrease, and it's advisable to use a more sophisticated model like Bayesian Network.

2.2 Exploratory Data Analysis (EDA)

2.2.1 Initial Data Exploration

This section will focus on the dataset and any features that we consider relevant to the analysis and modelling task. We plan to dive into the data to grasp the distribution of 'y' (deposit subscription), examine the interactions among various attributes, and identify any significant patterns that might surface.

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-1	0	unknown	no
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	339	4	failure	no
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	330	1	failure	no
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-1	0	unknown	no
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-1	0	unknown	no

Table 1 - Dataset Head Display

After modifying original Bank Marketing Dataset and giving a proper name to class target 'y', we can observe the following:

The dataset consists of 4,521 entries across 17 attributes. This indicates a robust data volume suitable for detailed analysis (Table 2). 16 of the attributes are features and 1 is the target to predict the outcomes of subscribing to a term deposit (yes/no).

The dataset has seven integer columns (numerical data), refer to Table 2:

- Age (Years)
- Balance (Average yearly balance)
- Day (Last contact day of the week)
- Duration (Last contact duration in seconds)
- Campaign (Number of contacts during this campaign)
- Pdays (Number of days since the client was last contacted)
- Previous (Number of contacts before this campaign)

Additionally, the dataset includes ten object columns, representing categorical variables (Table 2):

- Job (Type of job)
- Marital (Marital status)
- Education (Education level)
- Default (Credit in default: yes/no)
- Housing (Has housing loan: yes/no)
- Loan (Has personal loan: yes/no)
- Contact (Contact communication type)
- Month (Last contact month)
- Poutcome (Outcome of previous marketing campaign)
- Deposit Subscription (y) (Target variable: Has the client subscribed to a term deposit? yes/no)

Furthermore:

- There are no missing values in any of the columns, eliminating the need for data imputation (Table 3).
- There are no duplicate entries, which simplifies the preprocessing phase and ensures the uniqueness of the data for modelling (Table 3).
- The **deposit subscription** column, which serves as the class label, categorises clients into two groups: those who subscribed to a term deposit (**yes**) and those who did not (**no**). This allows for precise classification analysis aimed at predicting client behaviour concerning term deposit subscriptions.

---Dataset Basic Information---

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   age                  4521 non-null   int64 
 1   job                  4521 non-null   object
 2   marital              4521 non-null   object
 3   education            4521 non-null   object
 4   default              4521 non-null   object
 5   balance              4521 non-null   int64 
 6   housing              4521 non-null   object
 7   loan                 4521 non-null   object
 8   contact              4521 non-null   object
 9   day                  4521 non-null   int64 
 10  month                4521 non-null   object
 11  duration             4521 non-null   int64 
 12  campaign             4521 non-null   int64 
 13  pdays                4521 non-null   int64 
 14  previous             4521 non-null   int64 
 15  poutcome             4521 non-null   object
 16  deposit subscription 4521 non-null   object
dtypes: int64(7), object(10)
memory usage: 600.6+ KB
None
```

Table 2 - Dataset Basic Info

```
---Dataset Null Check---

age                     0
job                     0
marital                 0
education               0
default                 0
balance                 0
housing                 0
loan                    0
contact                 0
day                     0
month                   0
duration                0
campaign                0
pdays                   0
previous                0
poutcome                0
deposit subscription    0
dtype: int64

---Dataset Duplicates Check---

0
```

Table 3 - Dataset Null and Duplicates Check

Based on the summary statistics presented in Table 4 and Table 5 for the Bank Marketing Dataset the following patens are revealed

Numerical Attributes:

- Clients' age range from 19 to 87 years old, with an average age of 41, reflecting a diverse age group.
- Financial balances vary significantly from -3313 to 71,188, indicating a wide economic diversity among clients.
- Duration and frequency of contacts vary, highlighting differences in how clients interact with the marketing efforts.

Categorical Attributes:

- The dataset contains a range of job types, marital statuses, and educational backgrounds, with 'management' jobs and 'married' marital status being the most common.
- Most clients have housing loans and no credit defaults, typical characteristics that may affect their decisions on term deposits.
- Many contacts did not result in a deposit, with 'unknown' being a frequent previous outcome, suggesting areas for potential improvement in campaign strategies.

Dataset Summary Statistics for Numerical

	age	balance	day	duration	campaign	pdays	previous
count	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000	4521.000000
mean	41.170095	1422.657819	15.915284	263.961292	2.793630	39.766645	0.542579
std	10.576211	3009.638142	8.247667	259.856633	3.109807	100.121124	1.693562
min	19.000000	-3313.000000	1.000000	4.000000	1.000000	-1.000000	0.000000
25%	33.000000	69.000000	9.000000	104.000000	1.000000	-1.000000	0.000000
50%	39.000000	444.000000	16.000000	185.000000	2.000000	-1.000000	0.000000
75%	49.000000	1480.000000	21.000000	329.000000	3.000000	-1.000000	0.000000
max	87.000000	71188.000000	31.000000	3025.000000	50.000000	871.000000	25.000000

Table 4 - Dataset Summary Statistic for Numerical

Dataset Summary for Categorical

	job	marital	education	default	housing	loan	contact	month	poutcome	deposit subscription
count	4521	4521	4521	4521	4521	4521	4521	4521	4521	4521
unique	12	3	4	2	2	2	3	12	4	2
top	management	married	secondary	no	yes	no	cellular	may	unknown	no
freq	969	2797	2306	4445	2559	3830	2896	1398	3705	4000

Table 5 - Dataset Summary for Categorical

2.2.2 Data Distribution and Outlier Detection

Based on the table 6, figure 1 and figure 2 results, we can observe the following insights:

- Age data exhibits a moderate right skew with the majority of individuals under the age of 60. The skewness value of 0.6993 suggests a concentration of younger individuals. Some outliers are present above the age of 70, but these are within reasonable real-world possibilities and may represent older banking clients.
- Balance is highly positively skewed (skewness: 6.5942), indicating that most individuals have lower balances. Outliers are present at very high balance levels, which could represent high net worth individuals. These outliers could represent important customer segments.
- Day data is relatively evenly distributed with minimal skewness (0.0946), indicating that marketing contacts are evenly spread throughout the month. There are no outliers, as all data points fall within a normal range. No outlier treatment is needed.
- Duration shows a significant right skew (skewness: 2.7715), indicating most calls are short. Some extreme values suggest very long calls. Given the high skewness and presence of long-duration calls, these outliers might be particularly important. They could be occurring with key clients or in situations where the client is at a pivotal decision-making point, thus potentially more likely to convert into successful transactions.
- Campaign. Number of contacts per campaign is heavily right-skewed (skewness: 4.7423), with most customers contacted fewer than 10 times. Outliers exist as some individuals are contacted many more times. These outliers might indicate high effort on certain clients.
- Pdays is heavily skewed (skewness: 2.7162) with many customers not being contacted previously (pdays = -1). The data points above 400 days are outliers and rare. These outliers could represent a small, possibly distinct customer segment
- Previous. Number of contacts before this campaign is highly skewed (skewness: 5.8733), with most values clustered near zero. There are outliers with high values. These data points might represent high-contact individuals and could be important for certain model types.

After careful review and thorough analysis of the outliers in our dataset, we have decided to retain these data points, recognizing that they represent realistic and potentially insightful scenarios. By preserving these outliers, we capture a complete picture of customer interactions, which enhances our understanding of diverse client behaviours and supports more effective modelling. This approach not only strengthens our strategic decision-making but also ensures that our analysis remains comprehensive. However, we will continue to monitor these outliers and adjust our approach if they significantly affect the model's accuracy.

Detecting Outliers using Z Score Method

Numerical Data Only:

```
age          44
balance      88
day           0
duration     88
campaign     87
pdays       171
previous     99
dtype: int64
```

Table 6 - Dataset Outliers Z-Score Method

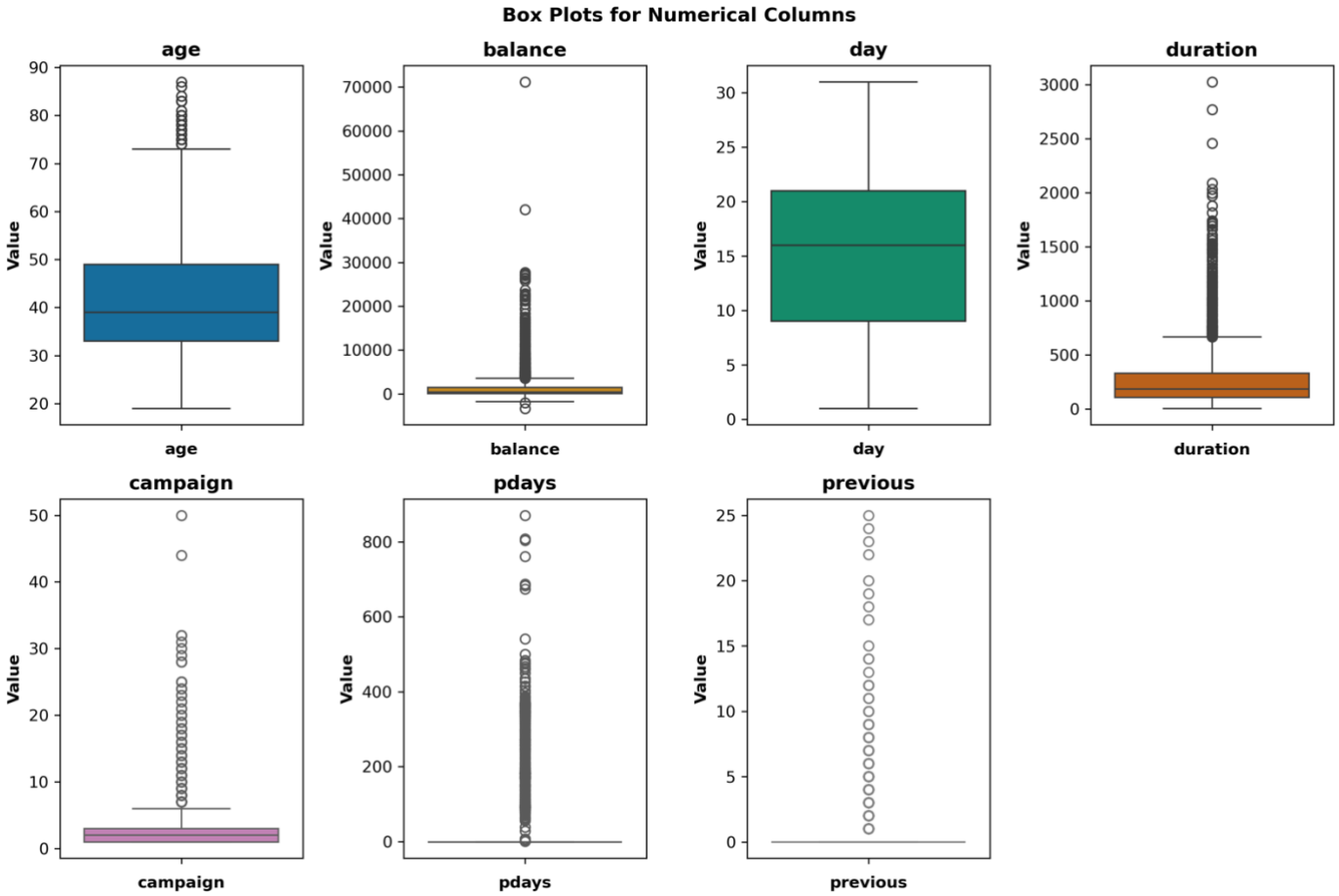

Figure 1 - Box Plots of Numerical Columns

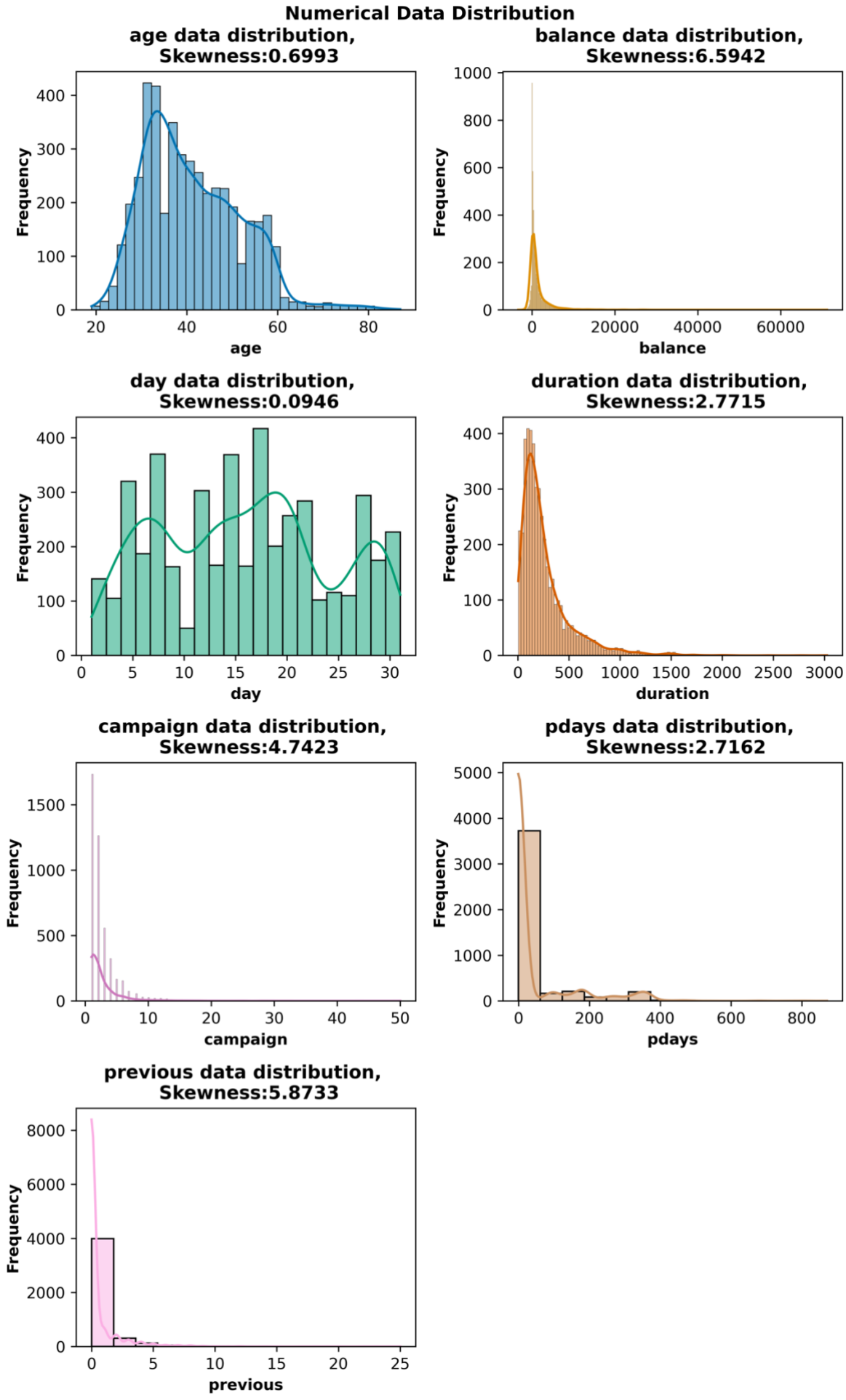

Figure 2 - Numerical Data Histograms

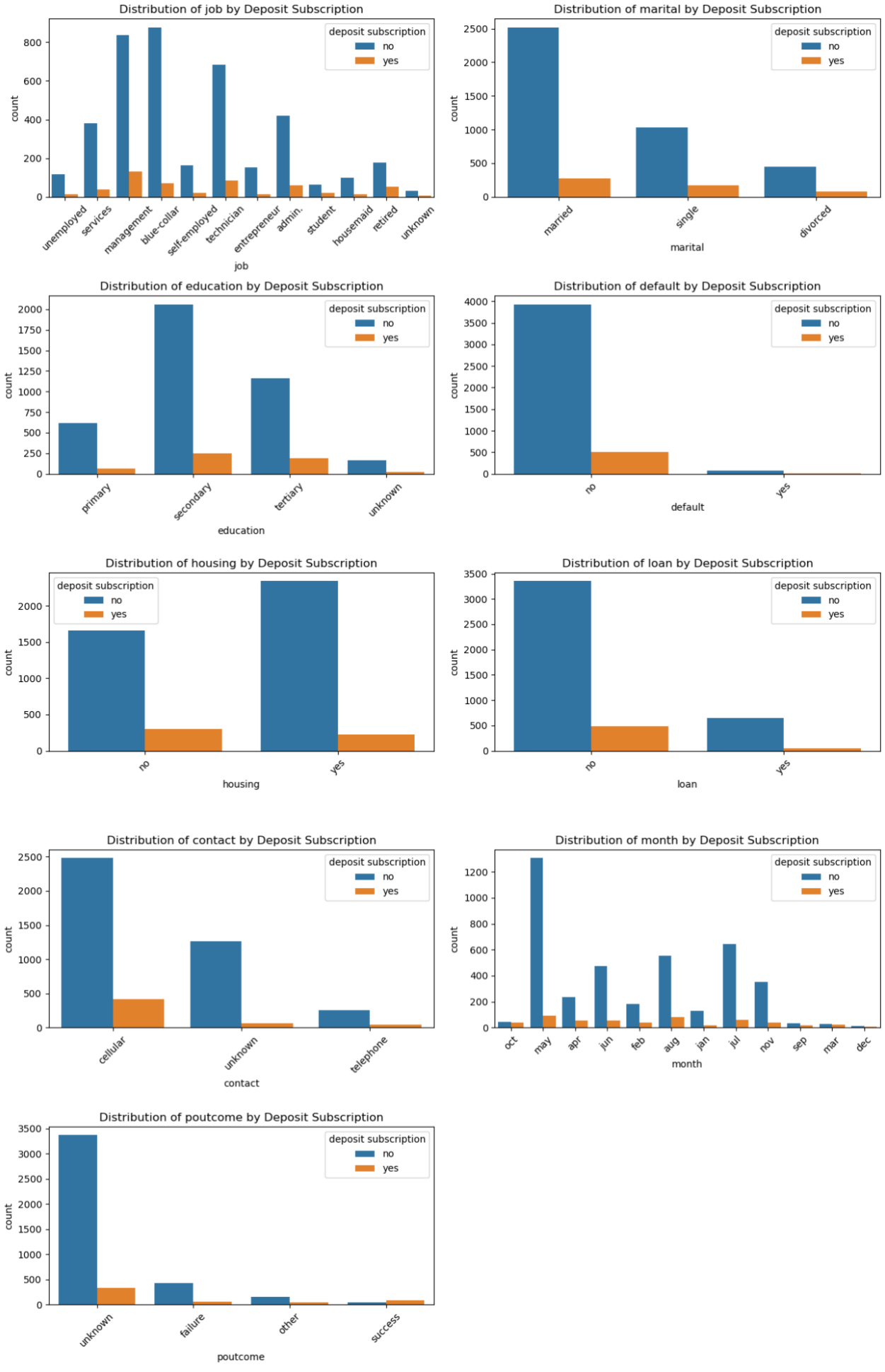

Figure 3 - Distribution of Categorical by Deposit Subscription

2.2.3 Categorical Data Analysis

Based on the categorical data plot shown in Figure 3 we can identify the following patterns:

- **Job**: The job category shows significant variation in deposit subscription rates, with management, technician, and blue-collar jobs showing higher counts, especially among those not subscribing. This suggests certain professions are more frequently targeted or less likely to subscribe.
- **Marital Status**: Married individuals are less likely to subscribe to deposits compared to single and divorced individuals, indicating that marital status could influence financial product decisions.
- **Education**: Clients with secondary education are the most targeted group, but they do not necessarily subscribe the most. Higher education levels (tertiary) show a relatively higher propensity to subscribe, suggesting education level impacts the likelihood of subscribing.
- **Default**: Clients without a credit default are far more likely to be targeted and to subscribe, highlighting credit history as a significant factor in marketing strategies.
- **Housing**: Individuals without a housing loan are more likely to subscribe to a deposit, indicating that financial liabilities could affect the decision to invest in deposits.
- **Loan**: Similarly, those without personal loans show a higher subscription rate, supporting the notion that fewer financial commitments correlate with higher subscription rates.
- **Contact**: Cellular contacts lead to more subscriptions than other methods, emphasising the effectiveness of direct communication.
- **Month**: While May has the highest contact rates, months like March, September, October, and December see better conversion rates, suggesting timing influences subscription success.
- **Previous Campaign Outcome ('potcome'):** Success in previous campaigns strongly predicts higher subscription rates in current campaigns.

Initial explorations of categorical data reveal significant factors influencing the outcomes of bank marketing campaigns. These insights are pivotal for guiding more focused and strategic campaign planning and customer targeting, enabling more personalised strategies that meet the specific needs of different clients.

2.2.4 Target Class Distribution

Figure 4 illustrates the dataset's distribution across various deposit subscription categories, with 'Deposit Subscription' considered as our target variable for analysis and predictive modelling. The bar chart clearly shows a significant imbalance between the categories, with a predominant number of samples in the 'no' category, totalling 4000 instances. On the contrary, the 'yes' category represents only 521 instances, indicating a much lower frequency of positive responses to deposit subscriptions. This pronounced imbalance poses a challenge for predictive modelling, emphasising the need for techniques to address the skewed nature of the dataset.

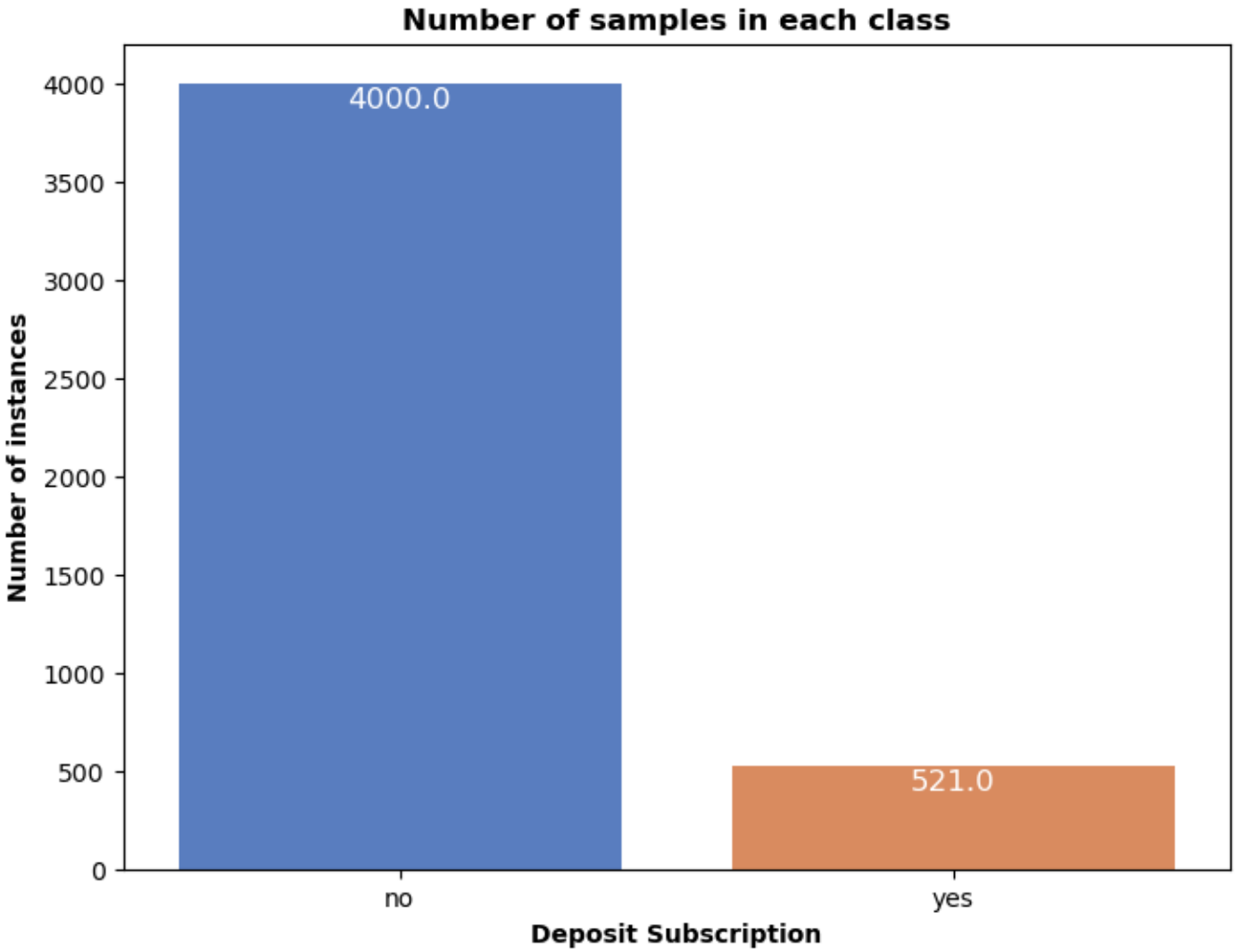

Figure 4 - Distribution of Deposit Subscription Outcomes

2.3 Feature Selection and Analysis

In this section, we aim to identify the most influential features for classifying the dataset. Using appropriate statistical and analytical methods, we will determine the top 5 features that have the highest impact on predicting whether a customer will subscribe to a term deposit.

2.3.1 Correlation Coefficients

To explore the relationships between the numerical features in our dataset, we generated a Pearson correlation heatmap. This visual representation, along with the correlation matrix, helps in understanding the strength and direction of relationships between pairs of features.

Based on Figure 5 and Table 7, the strongest correlation (0.58) is observed between the number of days since the client was last contacted (‘pdays’) and the number of contacts performed before this campaign (‘previous’). This moderately strong positive correlation indicates that clients who were contacted more frequently in the past tend to have a higher number of days since their last contact. The rest of the features show very weak correlations with each other, with coefficients close to zero. This suggests that these features are largely independent of each other.

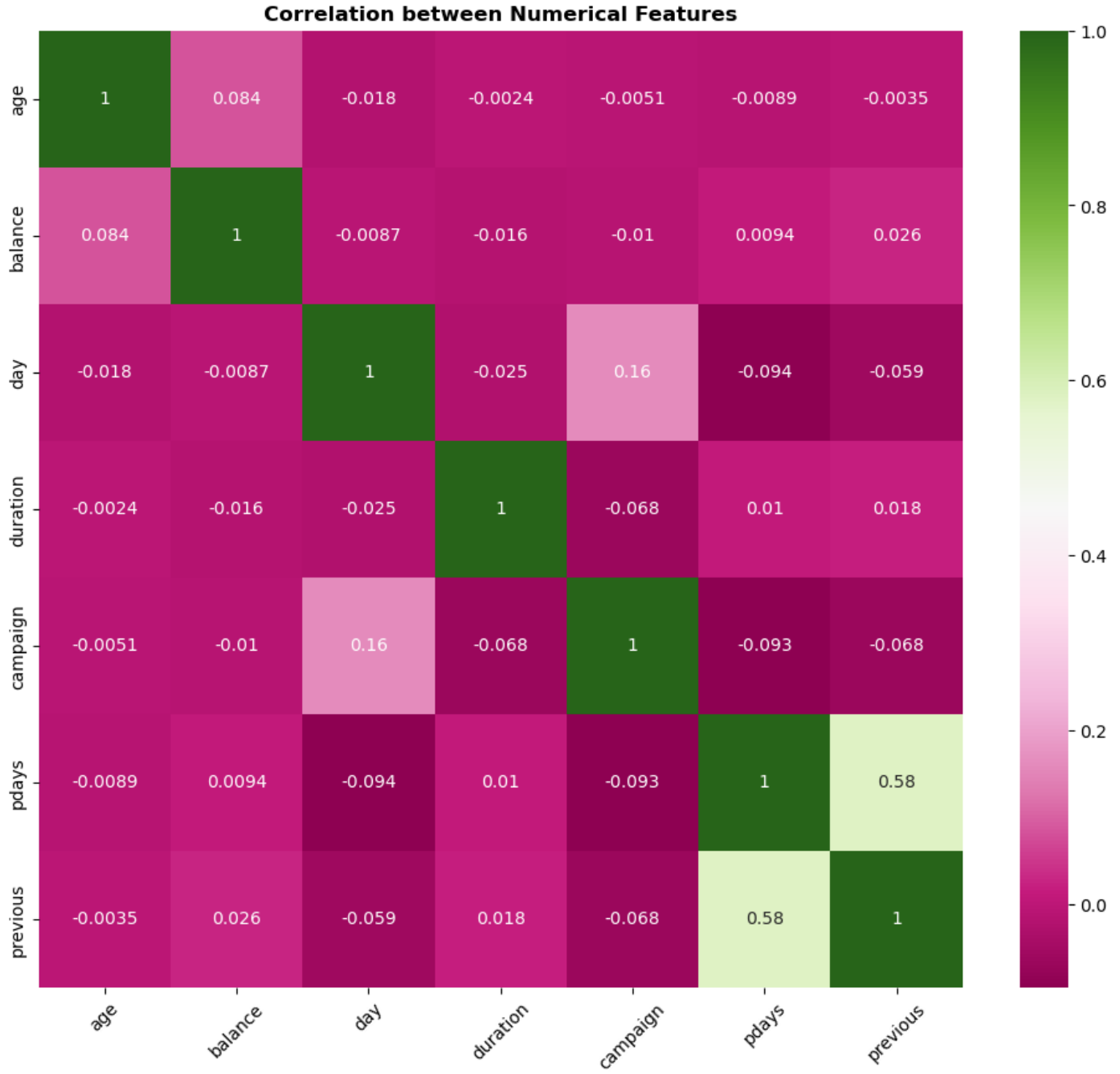

Figure 5 - Correlation Heatmap of Numerical Features

Pearson Correlation Coefficients

	age	balance	day	duration	campaign	pdays	previous
age	1.00	0.08	-0.02	-0.00	-0.01	-0.01	-0.00
balance	0.08	1.00	-0.01	-0.02	-0.01	0.01	0.03
day	-0.02	-0.01	1.00	-0.02	0.16	-0.09	-0.06
duration	-0.00	-0.02	-0.02	1.00	-0.07	0.01	0.02
campaign	-0.01	-0.01	0.16	-0.07	1.00	-0.09	-0.07
pdays	-0.01	0.01	-0.09	0.01	-0.09	1.00	0.58
previous	-0.00	0.03	-0.06	0.02	-0.07	0.58	1.00

Table 7 - Pearson Correlation Coefficients Matrix

2.3.2 ANOVA F-test

In this section, we employed the ANOVA F-test to identify the most significant features for predicting whether a customer will subscribe to a term deposit. The ANOVA F-test is a statistical method used to compare the means of different groups and determine whether the observed differences are statistically significant. By ranking all features based on their F-scores, we can highlight those that contribute most to the prediction task.

The F-scores for each feature, displayed in Figure 8, provide a visual representation of their significance. Features with higher F-scores (shown in Table 8) have a greater impact on the target variable and are thus considered more important. Notably, the 'duration' feature has the highest F-score, standing out among other features. This could be a reason for an unbiased model.

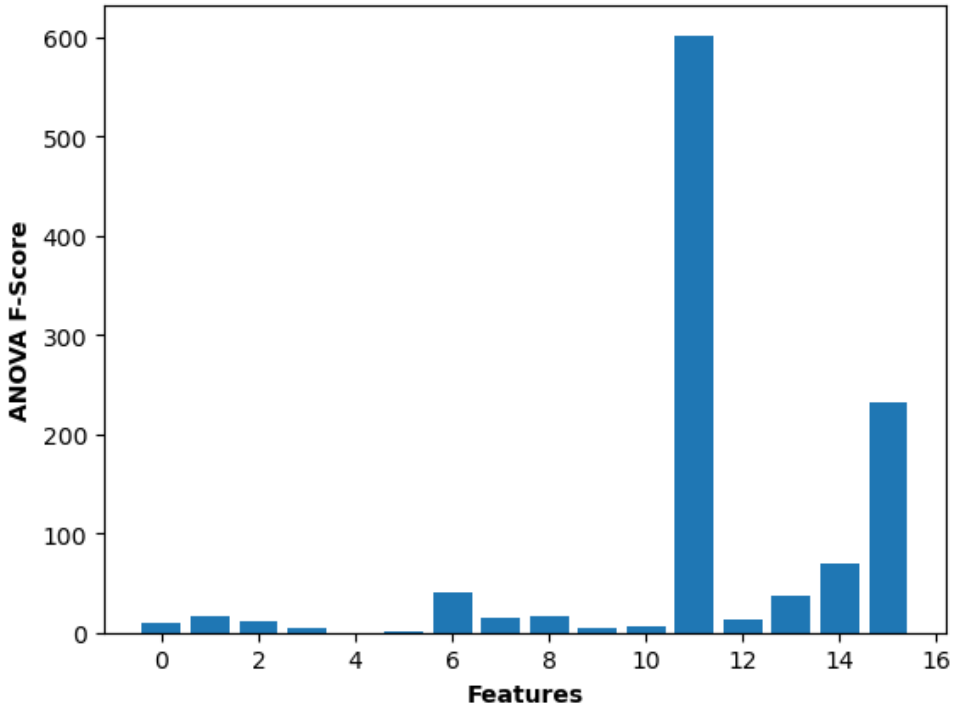

Figure 6 - Bar Plot of ANOVA F-Scores

ANOVA F-test

Feature 0:	9.287073
Feature 1:	16.078631
Feature 2:	11.160435
Feature 3:	4.683322
Feature 4:	0.241035
Feature 5:	1.305466
Feature 6:	40.320214
Feature 7:	14.512146
Feature 8:	17.036395
Feature 9:	4.197823
Feature 10:	6.104042
Feature 11:	601.815465
Feature 12:	13.141424
Feature 13:	37.537236
Feature 14:	70.061518
Feature 15:	231.977159

Table 8 - ANOVA F-Scores for Features

2.3.2 Top 5 Features

The correlation matrix shows that most features are largely independent, with the strongest correlation between **pdays** and **previous**. This suggests that each feature contributes uniquely to the prediction task without redundant information.

The ANOVA F-test ranked **duration**, **poutcome**, **previous**, **housing**, and **pdays** as the top five features:

- **Duration**: This feature has the highest F-score of 601.82, indicating that the duration of the last contact is the most influential factor in determining whether a customer will subscribe to a term deposit.
- **Poutcome:** The outcome of the previous marketing campaign also significantly affects the likelihood of subscription (F-score is 231.98).
- **Previous:** The number of contacts performed before this campaign is another crucial factor, reflecting customer engagement and interest. However, compared to the first two features, the F-score is only 70.06.
- **Housing:** Whether the customer has a housing loan influences their decision-making regarding term deposits (F-score is 40.32).
- **Pdays:** The number of days since the client was last contacted is also a key determinant, highlighting the importance of recent interactions (F-score is 37.54).

The features **previous** and **pdays** appear in both the correlation and ANOVA F-test results, indicating its significant role in the prediction task. The strong correlation between them aligns with their high F-scores, showing their combined influence. Additionally, **duration** is the most significant feature from the ANOVA F-test, which wasn't highlighted in the correlation matrix, demonstrating the value of using multiple methods for feature selection.

By combining insights from Pearson correlation coefficients and ANOVA F-scores (Table 9), we identified the top five influential features: **duration**, **poutcome**, **previous**, **housing**, and **pdays**. These features will be used in subsequent model building and evaluation tasks, providing a strong foundation for accurate predictions.

Top 5 Features based on F Score

feature_no	ANOVA F-Score	Feature Name
11	601.815465	duration
15	231.977159	poutcome
14	70.061518	previous
6	40.320214	housing
13	37.537236	pdays

Table 9 - Top 5 Features Ranked by ANOVA F-Scores

2.4 Independence Assumption in Naïve Bayes

In this section we are going to analyse the Pearson correlation coefficients among the top 5 features selected through the ANOVA F-test: **duration**, **poutcome**, **previous**, **housing**, and **pdays**.

Key Findings (Figure 7):

- **Duration**: Shows very weak correlation with other features, indicating its strong independence.
- **Poutcome**: Has moderate correlations with **previous** (0.62) and **pdays** (0.71), indicating that the outcome of the previous campaign is somewhat related to past interactions.
- **Previous**: Besides its moderate correlation with **poutcome**, it also shows a moderate correlation with **pdays** (0.58), highlighting its connection with past campaign interactions.
- **Housing**: Exhibits very weak correlations with all other features, suggesting its strong independence.
- **Pdays**: Displays moderate correlations with **poutcome** and **previous**, reinforcing the significance of past interactions.

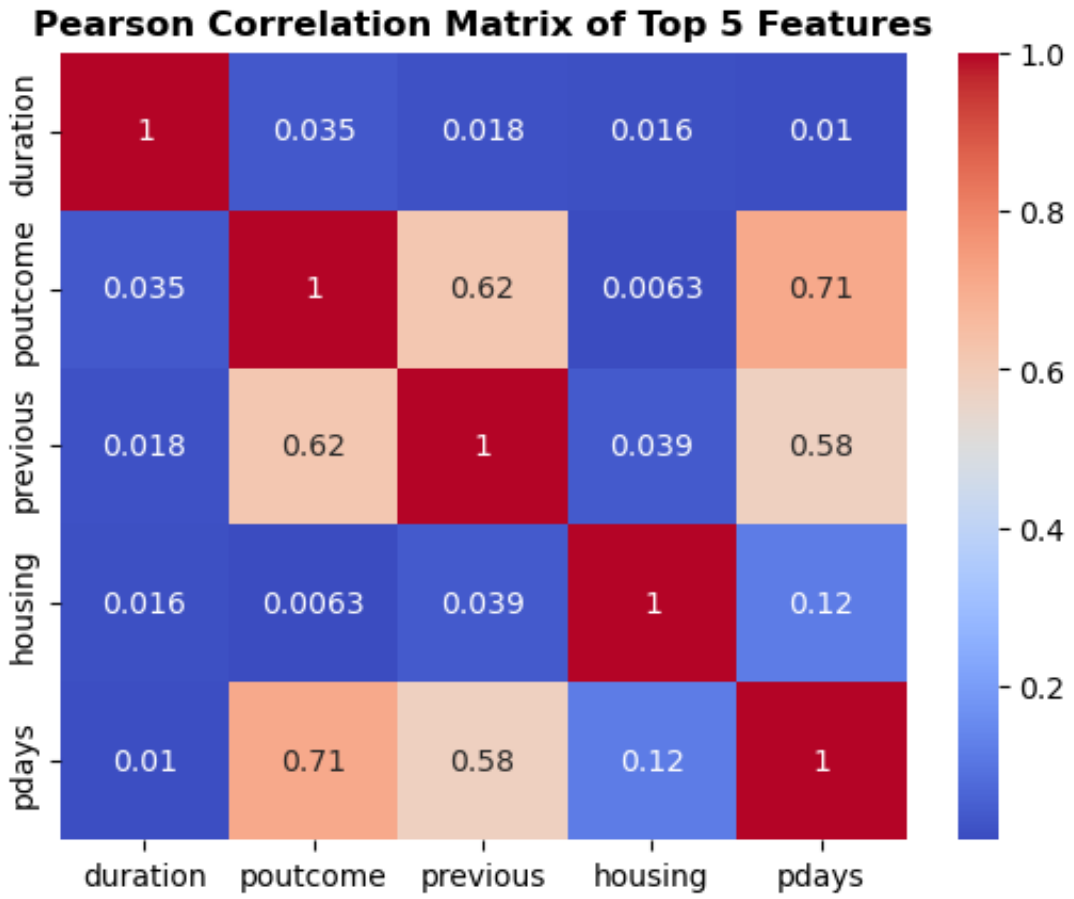

Figure 7 - Pearson Correlation Matrix of Top 5 Features

The independence assumption in the Naïve Bayes algorithm posits that features are independent given the class label. While **duration** and **housing** show very weak correlations with other features, suggesting they meet this assumption well, **poutcome**, **previous**, and **pdays** show moderate correlations with each other. Specifically, the correlations between **poutcome** and **previous** (0.62), **poutcome** and **pdays** (0.71), and **previous** and **pdays** (0.58) indicate some level of dependence. These dependencies suggest that while Naïve Bayes can still be viable, its performance might be influenced by these correlations.

To improve the model's prediction and mitigate the impact of these dependencies, we will employ resampling techniques such as Stratified K-Fold cross-validation. This approach helps ensure that each fold is representative of the overall class distribution, potentially leading to more robust and accurate model performance.

2.5 Naïve Bayes Model Building and Evaluation

In this section, we'll be evaluating the performance of the Naïve Bayes algorithm using the GaussianNB implementation with the selected features: **duration**, **poutcome**, **previous**, **housing**, and **pdays**.

We performed K-Fold cross-validation with 10 folds to ensure robust evaluation of the model. The average accuracy across all folds is 85.56%. The accuracy for each fold displayed in the Table 10.

K-Fold Cross-Validation with Gaussian Naive Bayes Classifier (k=10)

```
Average Accuracy: 85.56%
Fold 1: Accuracy = 84.55%
Fold 2: Accuracy = 86.73%
Fold 3: Accuracy = 86.06%
Fold 4: Accuracy = 83.85%
Fold 5: Accuracy = 84.29%
Fold 6: Accuracy = 84.51%
Fold 7: Accuracy = 88.50%
Fold 8: Accuracy = 86.95%
Fold 9: Accuracy = 84.73%
Fold 10: Accuracy = 85.40%
```

Table 10 - Cross-Validation Accuracy for Each Fold (Naïve Base)

The plot of accuracy per fold (Figure 8) shows some variability, with the highest accuracy reaching 88.50% in fold 7 and the lowest at 83.85% in fold 4. The average accuracy is marked with a red dashed line at 85.56%.

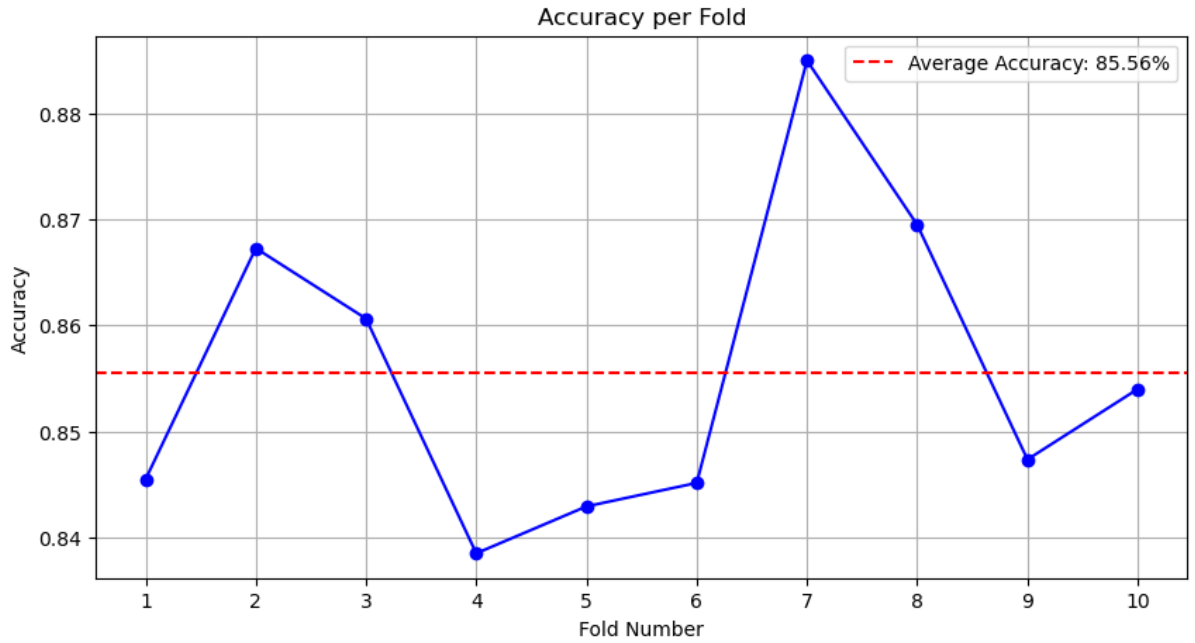

Figure 8 - Accuracy per Fold for Naïve Bayes Model

The confusion matrix (Figure 9) shows that the Naïve Bayes model correctly classified 1086 instances of the "no" class and 65 instances of the "yes" class. However, it misclassified 115 instances of the "no" class as "yes" and 91 instances of the "yes" class as "no". This indicates that while the model performs well in predicting the "no" class, it struggles more with accurately predicting the "yes" class.

After training the model on the entire training set and evaluating it on the test set, the overall accuracy was 84.82%.

The Naïve Bayes model displayed in Table 11 achieved an accuracy of 84.82%, with a weighted average F1-score of 0.85.The precision, recall, and F1-scores for the "no" class are high, indicating that the model performs well in identifying customers who do not subscribe to a term deposit. However, the scores for the "yes" class are lower, with a precision of 0.36 and a recall of 0.42, suggesting that the model struggles more with correctly identifying customers who will subscribe to a term deposit.

While the Naïve Bayes model shows good overall accuracy and performs well for the majority class ("no"), its performance for the minority class ("yes") indicates room for improvement. To address the biased prediction, we will explore alternative models that might better handle the dependencies among features and improve predictions for the minority class.

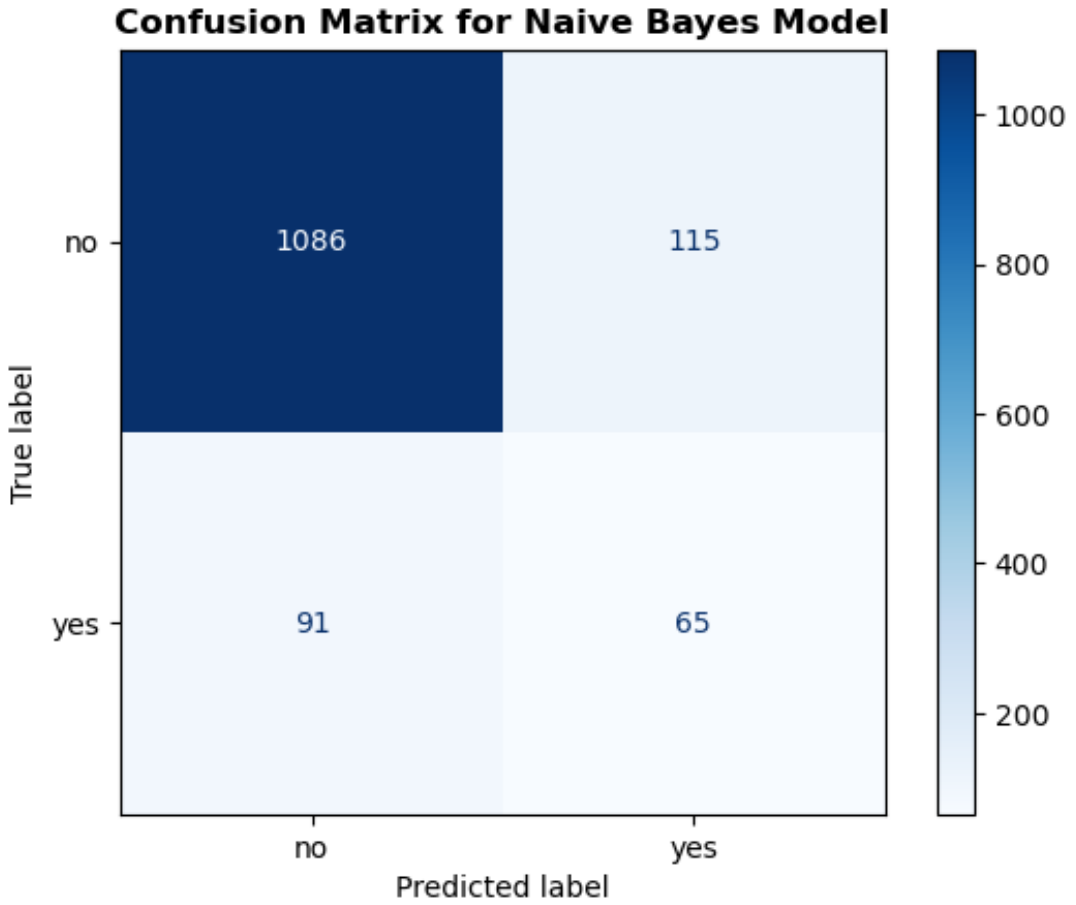

Figure 9 - Confusion Matrix for Naïve Bayes Model

Accuracy: 84.82%
Classification Report:

	precision	recall	f1-score	support
no	0.92	0.90	0.91	1201
yes	0.36	0.42	0.39	156
accuracy			0.85	1357
macro avg	0.64	0.66	0.65	1357
weighted avg	0.86	0.85	0.85	1357

Table 11 - Classification Report for Naïve Bayes Model

2.6 KNN Model Building and Evaluation

In this section, we'll be assessing the performance of the K-Nearest Neighbours (KNN) algorithm by fitting the model with various K values and analysing the resulting confusion matrix and classification report.

To determine the best K value, we trained the KNN model for K values ranging from 1 to 15. The accuracy for each K value is recorded for both the training and test sets (Table 12). The scores show some variability across different K values, helping us identify the optimal K for the model. The maximum test score of 89.24% was achieved with K = 9.

Finding the Optimal Value for K in K-nearest Neighbour Classifier (KNN)

```
K = 1: Train Score = 0.9646, Test Score = 0.8556
K = 2: Train Score = 0.9229, Test Score = 0.888
K = 3: Train Score = 0.9226, Test Score = 0.8873
K = 4: Train Score = 0.9125, Test Score = 0.8902
K = 5: Train Score = 0.9068, Test Score = 0.8873
K = 6: Train Score = 0.9023, Test Score = 0.888
K = 7: Train Score = 0.9046, Test Score = 0.8873
K = 8: Train Score = 0.9017, Test Score = 0.8858
K = 9: Train Score = 0.903, Test Score = 0.8924
K = 10: Train Score = 0.8992, Test Score = 0.8858
K = 11: Train Score = 0.9001, Test Score = 0.8909
K = 12: Train Score = 0.8995, Test Score = 0.8843
K = 13: Train Score = 0.896, Test Score = 0.8909
K = 14: Train Score = 0.8973, Test Score = 0.8814
K = 15: Train Score = 0.8973, Test Score = 0.8887
Max test score 89.24 % and k = [9]
```

Table 12 - Cross-Validation Accuracy for Each Fold (KNN)

The plot (Figure 10) shows the accuracy scores for both the training and test sets across different K values, highlighting the optimal K value of 9. Notably, compared to the Naïve Bayes model, the distribution of scores for KNN does not fluctuate much, indicating a more stable performance across different K values.

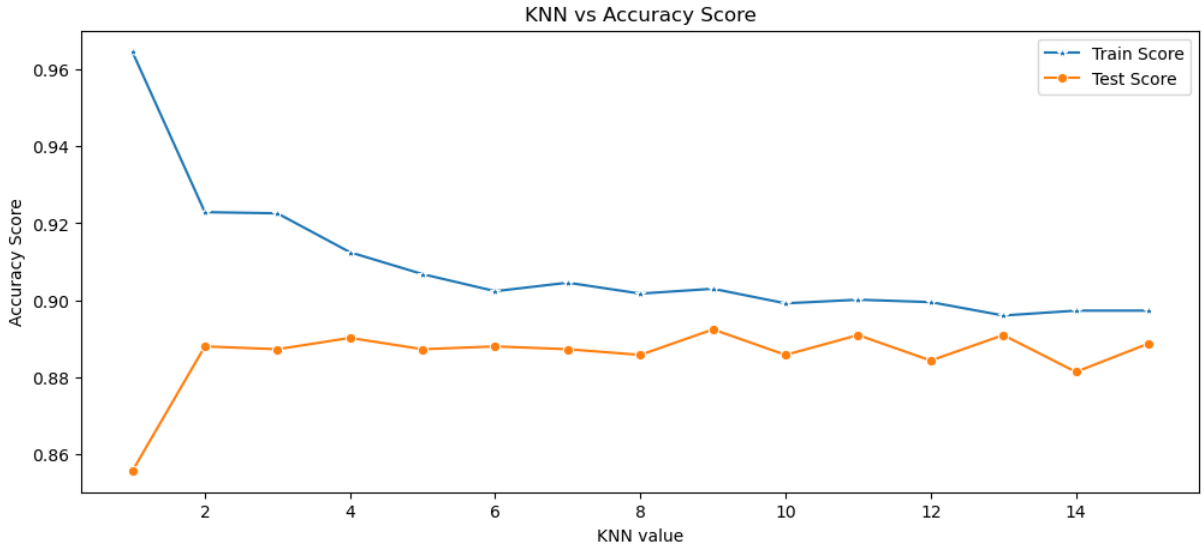

Figure 10 - Accuracy per Fold for KNN Model

The confusion matrix (Figure 11) shows that the KNN model correctly classified 1169 instances of the "no" class and 42 instances of the "yes" class. However, it misclassified 32 instances of the "no" class as "yes" and 114 instances of the "yes" class as "no". This indicates that the KNN model performs well in predicting the "no" class but has some difficulty with the "yes" class. It is worth mentioning that the KNN results in this matter are slightly better than those of the Naïve Bayes model.

The KNN model achieved an accuracy of 89.24%, with a weighted average F1-score of 0.88 (refer to Table 13). The precision, recall, and F1-scores for the "no" class are very high, indicating that the model performs well in identifying customers who do not subscribe to a term deposit. However, the scores for the "yes" class are lower, with a precision of 0.57 and a recall of 0.27, suggesting that the model still struggles with correctly identifying customers who will subscribe to a term deposit.

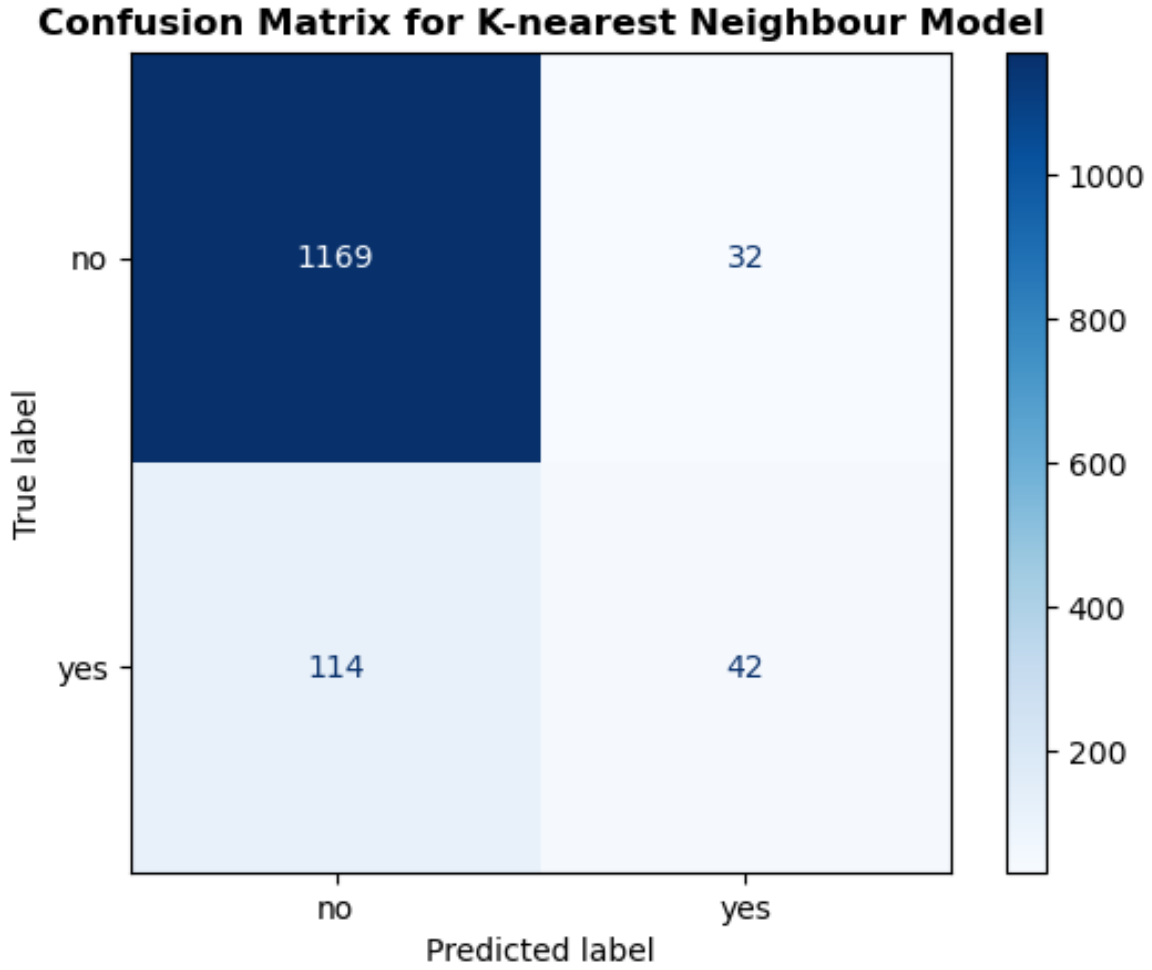

Figure 11 - Confusion Matrix for KNN Model

Accuracy: 89.24%
Classification Report:

	precision	recall	f1-score	support
no	0.91	0.97	0.94	1201
yes	0.57	0.27	0.37	156
accuracy			0.89	1357
macro avg	0.74	0.62	0.65	1357
weighted avg	0.87	0.89	0.88	1357

Table 13 - Classification Report for KNN Model

2.7 Model Comparison

Finally, we can compare the performance of the KNN and Naïve Bayes models in predicting customer subscription to term deposits. Both models were evaluated on the same dataset using the same set of features.

When comparing the KNN and Naïve Bayes models, the KNN model shows a slight improvement in overall accuracy (89.24% vs. 84.82%) and weighted average F1-score (0.88 vs. 0.85). The KNN model also has a higher precision for the "yes" class (0.57 vs. 0.36), although its recall for the "yes" class is lower (0.27 vs. 0.42) compared to the Naïve Bayes model.

Summary of Comparison:

- **Accuracy**: KNN (89.24%) > Naïve Bayes (84.82%)
- **Weighted F1-Score**: KNN (0.88) > Naïve Bayes (0.85)
- **Precision for "yes"**: KNN (0.57) > Naïve Bayes (0.36)
- **Recall for "yes"**: Naïve Bayes (0.42) > KNN (0.27)

Overall, the KNN model outperforms the Naïve Bayes model in terms of accuracy and precision for the "yes" class. Additionally, the KNN model shows more stability in its predictions with fewer misclassifications of the majority class ("no"). However, both models encounter challenges in accurately predicting the minority class ("yes"). Addressing these challenges may involve other resampling techniques, feature engineering, or experimenting with alternative models. Exploring other methods or adjusting model parameters could further enhance predictive performance.

3. Part B: Exploring Artificial Neural Networks

In this section, we will investigate different architectures for constructing an Artificial Neural Network (ANN). We will be employing 10-fold cross-validation for testing.

3.1 Activation Function and Learning Rate in MLP

The activation function in a Multilayer Perceptron (MLP) helps the model learn and understand complex data patterns. It adds non-linearity, allowing the model to handle more complicated relationships in the data. Common activation functions include:

- **Sigmoid**: Changes values to be between 0 and 1, useful for yes/no decisions.
- **Tanh**: Changes values to be between -1 and 1, often used to keep outputs balanced around zero.
- **ReLU (Rectified Linear Unit)**: Changes values to be between 0 and positive infinity, helping to avoid some problems during training.

Choosing the right activation function is important for the model to learn well.

The learning rate in MLP is a setting that controls how much the model's weights are adjusted during training. It influences how fast or slow the model learns.

- **High Learning Rate**: Makes the model learn quickly but can miss the best solution by jumping around too much.
- **Low Learning Rate**: Makes the model learn more slowly and carefully, but it can take a long time and might get stuck.

Finding the best learning rate is important for training the model effectively. It often requires trying different rates to see what works best.

3.2 Baseline Model with MLP Classifier

We are going to build an MLP Classifier with default parameter values and a single hidden layer with k neurons ($k \leq 25$). Table 14 displays the different configurations of k and max_iter along with their corresponding accuracies. We can see that the best k for a single hidden layer is 19, with the best number of iterations being 300, which provides the highest accuracy of 89.61%.

Optimizing best value for number of neurons (k)

and max_iter for Single Hidden Layer MLP Classifier

```
k=1, Best max_iter: 200, Highest accuracy: 88.50%
k=2, Best max_iter: 200, Highest accuracy: 88.50%
k=3, Best max_iter: 200, Highest accuracy: 88.50%
k=4, Best max_iter: 100, Highest accuracy: 88.58%
k=5, Best max_iter: 200, Highest accuracy: 88.87%
k=6, Best max_iter: 200, Highest accuracy: 89.24%
k=7, Best max_iter: 200, Highest accuracy: 89.24%
k=8, Best max_iter: 200, Highest accuracy: 89.24%
k=9, Best max_iter: 200, Highest accuracy: 89.24%
k=10, Best max_iter: 200, Highest accuracy: 89.24%
k=11, Best max_iter: 200, Highest accuracy: 89.24%
k=12, Best max_iter: 200, Highest accuracy: 89.24%
k=13, Best max_iter: 200, Highest accuracy: 89.24%
k=14, Best max_iter: 200, Highest accuracy: 89.24%
k=15, Best max_iter: 200, Highest accuracy: 89.24%
k=16, Best max_iter: 200, Highest accuracy: 89.39%
k=17, Best max_iter: 200, Highest accuracy: 89.39%
k=18, Best max_iter: 200, Highest accuracy: 89.39%
k=19, Best max_iter: 300, Highest accuracy: 89.61%
k=20, Best max_iter: 300, Highest accuracy: 89.61%
k=21, Best max_iter: 300, Highest accuracy: 89.61%
k=22, Best max_iter: 300, Highest accuracy: 89.61%
k=23, Best max_iter: 300, Highest accuracy: 89.61%
k=24, Best max_iter: 300, Highest accuracy: 89.61%
k=25, Best max_iter: 300, Highest accuracy: 89.61%

Best k for single hidden layer = 19, Overall Best max_iter: 300, Highest accuracy: 89.61%
```

Table 14 - Optimal Neurons & Iterations for MLP

After building the baseline MLP Model using the optimal parameters (k = 19 neurons and max_iter = 300), we output the confusion matrix and evaluate its accuracy.

Figure 12: Confusion Matrix for MLP Base Model

The confusion matrix shows that the MLP model correctly classified 1166 instances of the "no" class and 50 instances of the "yes" class. However, it misclassified 35 instances of the "no" class as "yes" and 106 instances of the "yes" class as "no". This indicates that the MLP model performs well in predicting the "no" class but still faces challenges with the "yes" class.

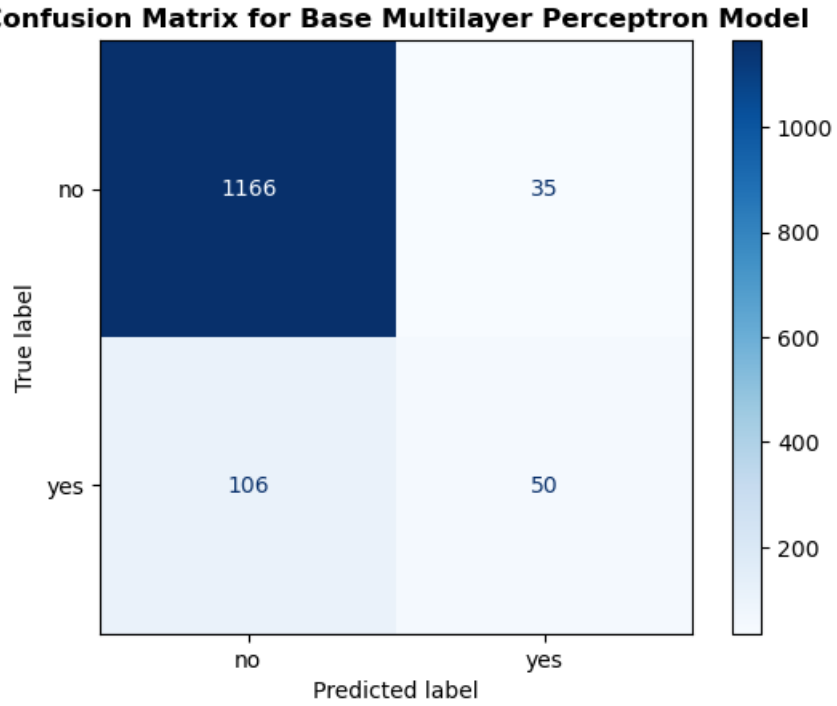

Figure 12 - Confusion Matrix for MLP Base Model

Table 15: Classification Report for Baseline MLP Model

The baseline MLP model achieved an accuracy of 89.61%. The precision, recall, and F1-scores for the "no" class are very high, indicating that the model effectively identifies customers who do not subscribe to a term deposit. However, the scores for the "yes" class are lower, with a precision of 0.59 and a recall of 0.32, suggesting the model struggles more with correctly identifying customers who will subscribe to a term deposit.

- Accuracy: 89.61%
- "No" Class: Precision = 0.92, Recall = 0.97, F1-score = 0.94
- "Yes" Class: Precision = 0.59, Recall = 0.32, F1-score = 0.41

Accuracy: 89.61%
Classification Report:

	precision	recall	f1-score	support
no	0.92	0.97	0.94	1201
yes	0.59	0.32	0.41	156
accuracy			0.90	1357
macro avg	0.75	0.65	0.68	1357
weighted avg	0.88	0.90	0.88	1357

Table 15 - Classification Report for Baseline MLP Model

The baseline MLP model performs well overall, particularly for the majority class ("no"). However, its performance for the minority class ("yes") indicates room for improvement.

3.3 Tracking Loss Value

After performance evaluation, we displayed the loss value during training and tracked how it changed with each iteration (Figure 13). The loss curve illustrates how the loss value changes over iterations during the training of the baseline MLP model.

The loss starts high, indicating a large difference between predicted and actual outputs at the beginning of training. As iterations increase, the loss value decreases rapidly, showing that the model is learning and improving its predictions. The curve flattens out as iterations progress, indicating that the model is converging and further training has little impact on reducing the loss.

The final training loss of 0.265 indicates that the model has achieved a relatively low error on the training data, supporting the good performance metrics observed in the previous section.

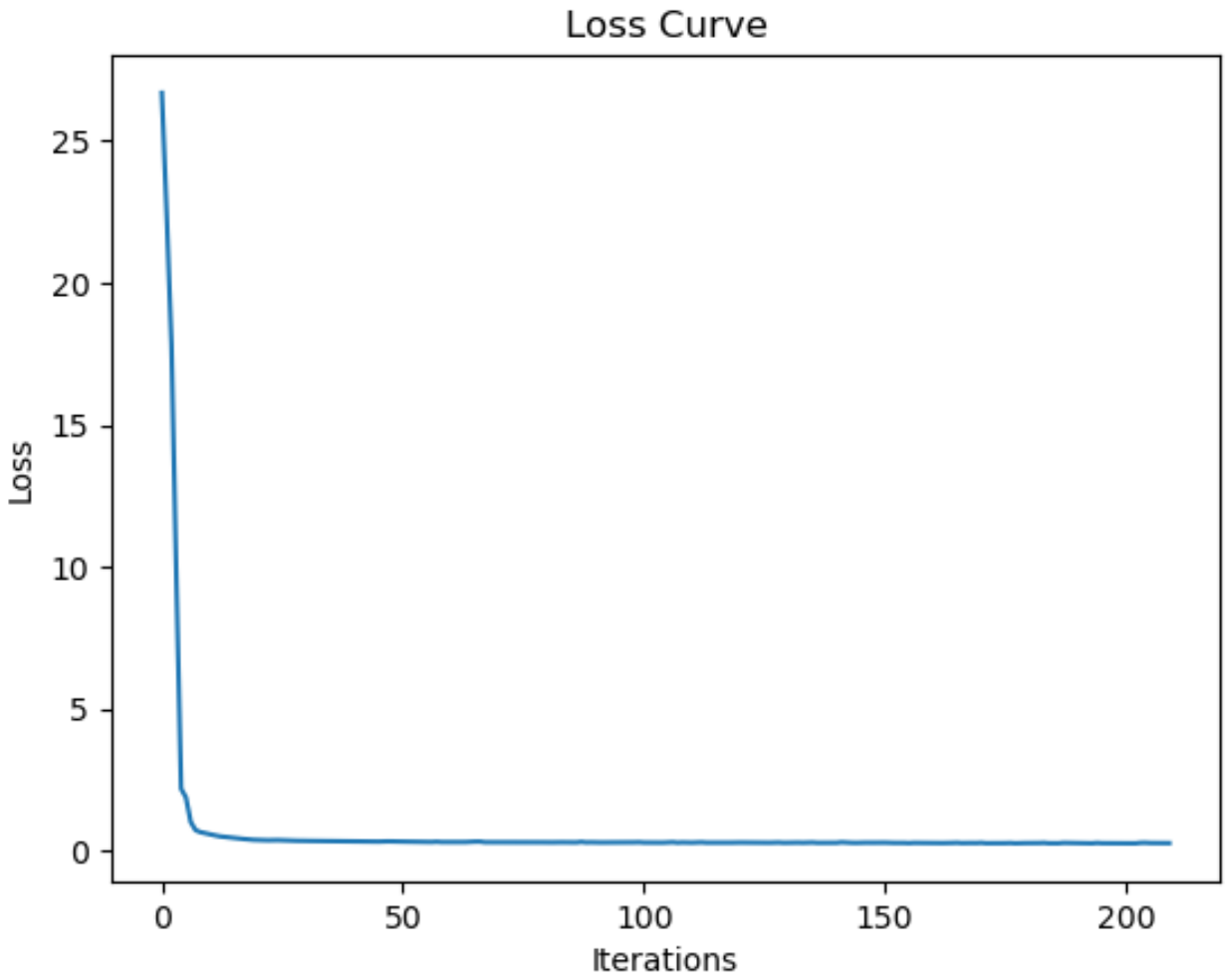

Figure 13 - Loss Curve for Baseline MLP Model

3.4 Experimenting with Two Hidden Layers

Using two hidden layers can improve the performance of a neural network by allowing it to learn more complex patterns and representations in the data. Each layer can capture different levels of abstraction, and combining these layers can lead to better generalisation and more accurate predictions.

After experimenting with two hidden layers, we find the best split of neurons across them that yields the highest classification accuracy (Table 16). The highest classification accuracy was achieved with the configuration (7, 18), resulting in an accuracy of 89.61%. Notably, some configurations like (12, 13) and (15, 10) also performed well, with accuracies close to the highest value.

This experiment shows that distributing neurons across two hidden layers can indeed enhance the model's performance. The optimal configuration identified is (7, 18), where the first hidden layer has 7 neurons and the second hidden layer has 18 neurons, achieving an accuracy of 89.61%. This indicates that a balanced distribution of neurons across two layers can effectively capture complex patterns in the data and improve classification accuracy.

Neurons Configuration and Accuracy Table:

Neurons Configuration	Accuracy
(24, 1)	0.885041
(23, 2)	0.885041
(22, 3)	0.885041
(21, 4)	0.878408
(20, 5)	0.879145
(19, 6)	0.885041
(18, 7)	0.871039
(17, 8)	0.882830
(16, 9)	0.890199
(15, 10)	0.893147
(14, 11)	0.883567
(13, 12)	0.885041
(12, 13)	0.894620
(11, 14)	0.875461
(10, 15)	0.886514
(9, 16)	0.885041
(8, 17)	0.868091
(7, 18)	0.896094
(6, 19)	0.867354
(5, 20)	0.892410
(4, 21)	0.885041
(3, 22)	0.885041
(2, 23)	0.866618
(1, 24)	0.885041

Table 16 - Neurons Configuration & Accuracy

3.5 Explaining Accuracy Variation

From the table 16 created in the previous part, we can observe the variation in classification accuracy with different splits of neurons across the two hidden layers. There are possible reasons for this variation such as:

- **Model Complexity**: Too many neurons in one layer can make the model too complex, leading to overfitting. This results in good performance on training data but poor performance on test data.
- **Balanced Feature Learning**: A balanced distribution of neurons allows the model to capture different levels of patterns in the data more effectively. Configurations like (7, 18) and (15, 10) show better performance due to this balance.
- **Interaction Between Layers:** Neurons in different layers work together to learn complex features. Balanced neuron distributions improve this interaction, leading to better model performance.

The variation in accuracy with different neuron splits across the two hidden layers highlights the importance of balanced configurations. Balanced neuron distributions facilitate efficient learning, effective feature representation, and better generalisation, leading to higher classification accuracies.

3.6 Comparing MLP Classifier Performance

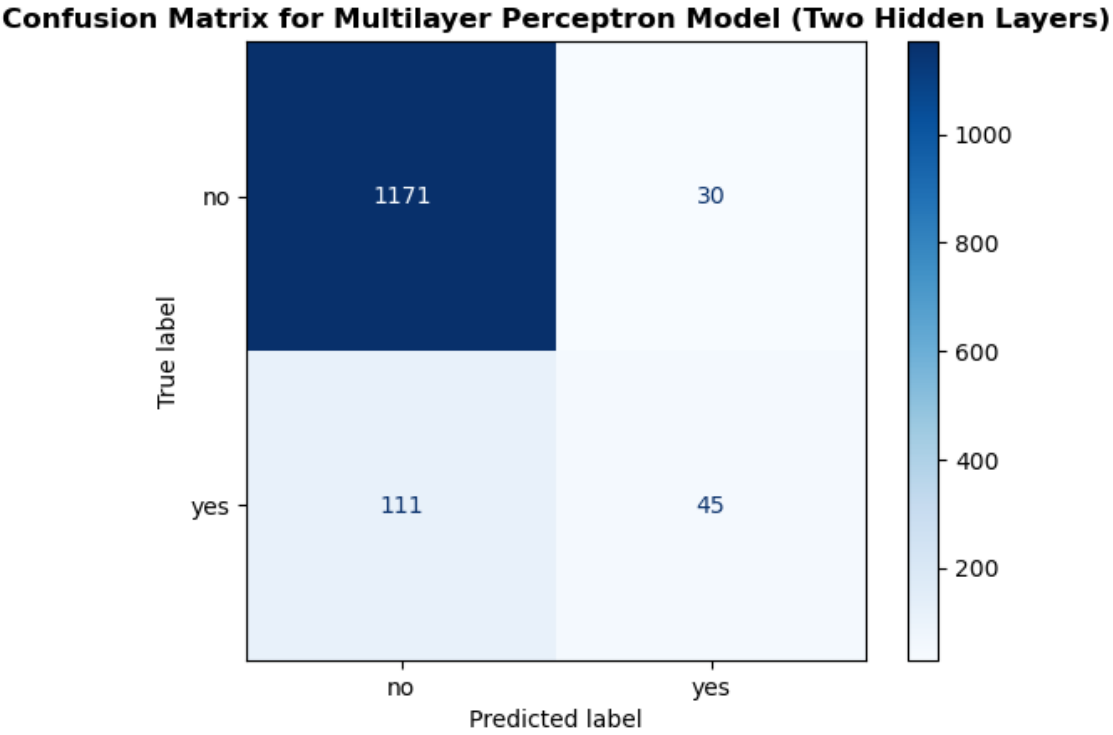

Figure 14 - Confusion Matrix for Two Layer MLP Model

The confusion matrix for the Multilayer Perceptron model with two hidden layers provides the following insights:

1171 instances of the "no" class and 45 instances of the "yes" class were correctly classified. However, the model misclassified 30 instances of the "no" class as "yes" and 111 instances of the "yes" class as "no". Similar to the base model, MLP Two Layer performs well in predicting the "no" class but still struggles with correctly identifying customers who will subscribe to a term deposit.

Accuracy: 89.61%
Classification Report:

	precision	recall	f1-score	support
no	0.91	0.98	0.94	1201
yes	0.60	0.29	0.39	156
accuracy			0.90	1357
macro avg	0.76	0.63	0.67	1357
weighted avg	0.88	0.90	0.88	1357

Table 17 - Classification Report for Two Layer MLP Model

Based on the accuracy performance (Table 15 and Table 17), both the baseline and the two-layer MLP models achieved the same accuracy of 89.61%. Therefore, we can conclude that the performance of the two models is the same in terms of overall accuracy. While the accuracy is the same, the slight differences in precision, recall, and F1-scores for the "yes" class may indicate nuanced performance variations, but these differences are not substantial enough to declare that one model performs better than the other one.

Comparing MLP Classifier Performance with other clasifiers

Naive Bayes Model Performance:

Accuracy: 84.82%
Classification Report:

	precision	recall	f1-score	support
no	0.92	0.90	0.91	1201
yes	0.36	0.42	0.39	156
accuracy			0.85	1357
macro avg	0.64	0.66	0.65	1357
weighted avg	0.86	0.85	0.85	1357

KNN Model Performance:

Accuracy: 89.24%
Classification Report:

	precision	recall	f1-score	support
no	0.91	0.97	0.94	1201
yes	0.57	0.27	0.37	156
accuracy			0.89	1357
macro avg	0.74	0.62	0.65	1357
weighted avg	0.87	0.89	0.88	1357

Best MLP (Two Hidden Layers) Model Performance:

Accuracy: 89.61%
Classification Report:

	precision	recall	f1-score	support
no	0.91	0.98	0.94	1201
yes	0.60	0.29	0.39	156
accuracy			0.90	1357
macro avg	0.76	0.63	0.67	1357
weighted avg	0.88	0.90	0.88	1357

Table 18 - Performance Comparison: MLP vs Naïve Bayes vs KNN

The final step of our analysis is to compare the performance of MLP, Naïve Bayes, and KNN classifiers. Table 18 presents a close comparison of performance metrics among these classifiers, helping us identify the best and least reliable prediction models.

The performance comparison of the MLP Classifier with Two Hidden Layers against Naïve Bayes and KNN on the same dataset reveals the following:

Naïve Bayes:

- **Strengths**: High precision and recall for the "no" class.
- **Weaknesses**: Low precision and recall for the "yes" class, indicating poor performance in identifying customers who will subscribe to a term deposit.

KNN:

- **Strengths**: High accuracy and strong performance for the "no" class.
- **Weaknesses**: Lower precision and recall for the "yes" class compared to the MLP model, indicating difficulty in correctly predicting the minority class.

MLP (Two Hidden Layers):

- **Strengths**: Highest accuracy among the three models and better precision for the "yes" class compared to KNN. It shows the highest recall for the "no" class, indicating strong performance in identifying customers who will not subscribe to a term deposit.
- **Weaknesses**: Although better than the other models, it still struggles with the "yes" class, showing moderate precision and recall. More time consuming to build than the other two.

Accuracy: MLP (89.61%) > KNN (89.24%) > Naïve Bayes (84.82%)

Based on the accuracy and classification reports, the MLP Classifier with two hidden layers is the best-performing model. It has the highest overall accuracy at 89.61% and offers a better balance between precision and recall for the "yes" class, followed by KNN. Conversely, the Naïve Bayes model presents the least reliable metrics, where feature dependency could lead to slightly poorer results. While all models perform well in predicting the majority class ("no"), the MLP model demonstrates a slight advantage in handling the minority class ("yes"), making it the most reliable choice for this dataset. However, it’s worth noting, that due to initial imbalanced dataset, all three models struggle with consistently identifying the "yes" class. Future improvements could focus on further enhancing the model's ability to predict the "yes" class, potentially through methods such as balancing the dataset, aiming for a less biased and more accurate result.

4. Conclusion

By conducting a comprehensive analysis of the Bank Marketing Dataset, we identified that the best-performing model out of the three was the MLP classifier. However, for practical usage, we also recommend the KNN classifier, which achieved nearly the same accuracy as the MLP model. While some of the KNN metrics were slightly lower, it requires less training time and is simpler and easier to implement.

KNN can serve as an initial identification model due to its simplicity and efficiency. Utilising these two models can help better understand marketing needs, identify the most influential features, and improve current marketing strategies. By focusing on the most important features and adjusting or eliminating the least important ones, organisations can enhance their subscription outcomes.

Overall, this analysis provides valuable insights and reliable models for predicting the success of bank marketing campaigns using various machine learning techniques.