Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Bank Customer Analysis

Group 2

M.Sc. Business Analytics

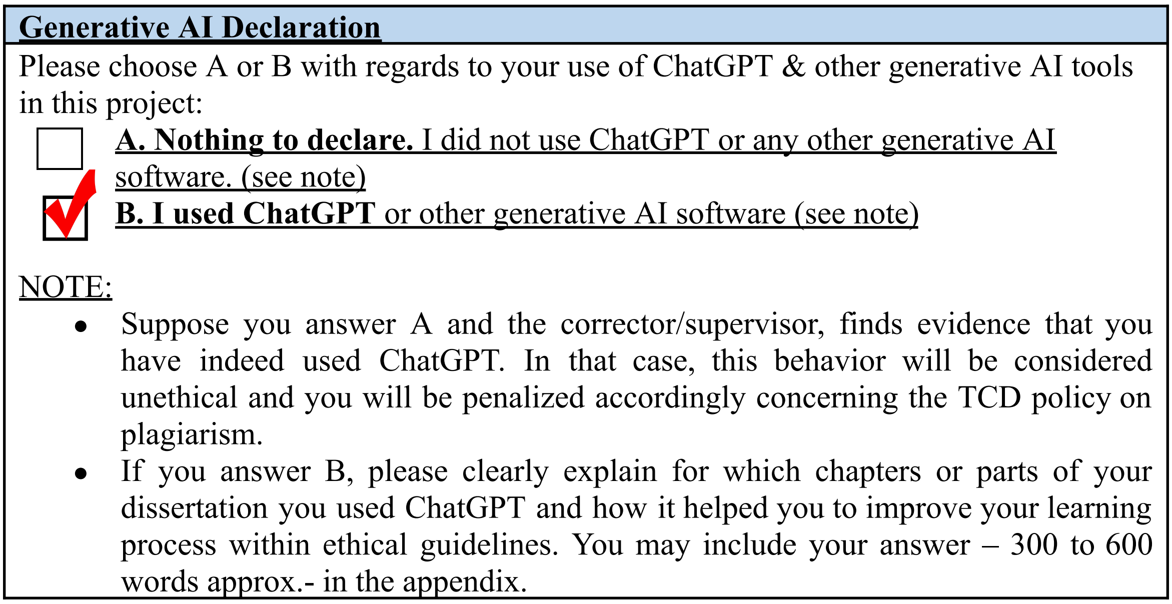
BU7154-202425 Foundations of Business Analytics
Professor Baidyanath Biswas

Trinity Business School
TRINITY COLLEGE
UNIVERSITY OF DUBLIN

15 November, 2024

DECLARATION

This work has not been submitted as an exercise for a degree at this or any other university and is entirely our own. We have read and understood the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at http://www.tcd.ie/calendar.



Generative AI Declaration

Please choose A or B with regards to your use of ChatGPT & other generative AI tools in this project:

□ **A. Nothing to declare.** I did not use ChatGPT or any other generative AI software. (see note)

☑ **B. I used ChatGPT** or other generative AI software (see note)

NOTE:

- Suppose you answer A and the corrector/supervisor, finds evidence that you have indeed used ChatGPT. In that case, this behavior will be considered unethical and you will be penalized accordingly concerning the TCD policy on plagiarism.
- If you answer B, please clearly explain for which chapters or parts of your dissertation you used ChatGPT and how it helped you to improve your learning process within ethical guidelines. You may include your answer – 300 to 600 words approx.- in the appendix.

Signed:

Anubhav Kumar-24339164

Kara Downing- 24365457

T.R Vijay Srivatsan- 24340271

Vidya Jaiswal-24335598

Xueying Ouyang-24332556

Usage of AI-

- Used GPT to deepen our understanding of the analytical methods
- Used AI to explore the research question and diagnose the root cause for bug reports on our code
- Used GPT to understand the logical relations between key concepts, the pros and cons of applying different methods
- Used AI to seek guidance on academic writing in the aspects of logic flow, clarity, structure, and relevance to the topic

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Banks are a crucial component of people’s daily lives. It provides financial services, investments, and protection of financial assets to individuals and businesses. It relies on a steady influx of deposits to maintain stable cash flow and profitability. This study built its analysis based on a dataset from a Portuguese bank with over 40,000 user records to identify the key factors that influence the propensity of term deposit subscriptions most, when users are more likely to make term deposits and pinpoint the most prospective user segment for long-term targeted campaigns.

To achieve these goals, the study took reference from previous studies and utilized multiple analytical and machine-learning methods, including PCA (Principal Component Analysis), LR (logistic regression), DT (decision tree), and k-means clustering. Different approaches were applied to assess the correlation across variables, determine the weights of each variable, capture nonlinear relations, and discover the high-potential user group.

Our findings have uncovered several distinct patterns for future actionable decisions. First, a moderate to strong correlation was discovered between ‘duration’ and ‘deposit’ and ‘previous’ and ‘pdays’. This demonstrates the strategic importance of conducting an effective telemarketing campaign. Second, users tend to have a preference for making term deposits in the first half of the week (Tuesdays and Wednesdays in particular) and at the beginning and end of the month. That suggests the importance of aligning promotional activities with that unique timing pattern. Third, with k-means clustering, the study identified the user profile of the most high-potential users, who are wealthy and risk-averse with high financial stability and bank balances significantly higher than the average.

PROJECT GOAL

The project is organized around three major research questions as follows:
1) Which factors influence customers' decision to subscribe to term deposits?

Precise investments on targeted campaigns: this research question taps into the whole list of predictors by examining which ones impact deposit behaviours the most. Addressing this question facilitates a well-rounded overview of the influencing factors behind term deposit subscriptions by understanding the underlying motivations. This would eventually lead to the bank's sustainable business growth.

2) On what days of the week or times of the month do users show a higher likelihood of making a term deposit?

People receive unsolicited calls and promotional messages every day, from day to night, weekdays to weekends. Is there a pattern on which days of the week or which times of the month work the best? Exploring this research question helps reveal unique patterns behind users' financial behaviours and preferences. It fosters strategic, precise planning of promotional telemarketing activities that align with the discovered timing pattern.

3) For future telemarketing campaigns, which user segment should the bank target?

Telemarketing requires substantial financial investment. Banks need to allocate those investments wisely to maximise the ROI. This last research question focuses on pinpointing the most prospective user segmentation. Addressing this question shall allow the business stakeholders to concentrate on the most high-potential customers, with matching tailored services, and nurture long-term trust and collaboration.

LITERATURE REVIEW

Looking a few years back, Portugal's banking industry has encountered severe challenges since the collapse of Banco Espirito Santo in 2014. The industrial landscape, despite increasing stabilities over the past decade, is now highly competitive due to surging influence from overseas investment and the entry of foreign entities (Jochen, 2015). The report released by the European Commission in 2024 pointed out that Portugal's banking industry still faces high debt. Additionally, the growing housing loan interest rates (4.3%) may have hindered individuals' propensity to make term deposits given the pressure on the household finances (European Union, 2024).

Aside from those challenges, marketing strategies nowadays are more data-driven to warrant effective and precise customer acquisition. In 2023, the global big data analytics market in banking is expected to reach $745.16 billion by 2030 with a 13.5% CAGR growth rate (Puja, 2024). In Europe, ad spending in the telemarketing market in Europe is forecasted to reach 2.32 billion dollars in 2024 (Telemarketing - Worldwide, 2024). All those trends demonstrate the importance of data analytics and how it transforms traditional marketing tactics to drive continuous business growth.

Based on the empirical findings, there has been a prevalent adoption of data analytics in the banking industry. In previous studies, machine learning methods such as k-NN, Random Forest, and Decision Trees were commonly used techniques to segment users for targeted marketing and evaluate their likelihood of desired behaviours. The featured predictors include job type, marital status, and education level (Asare-Frempong, J. and Jayabalan, M, 2017). Moreover, Logistic Regression was also often utilised to identify key variables alone. Key statistical metrics such as accuracy, precision, and recall rate were used to evaluate the model's performance (Ahmad Ilham et al 2019).

The appliance of those methods from the previous studies was used as reference to guide this study.

METHODOLOGY

Python was the major analytical language used in this project for data processing and visualization. Before diving deep into the analysis, the study used R to generate descriptive statistics on the dataset. Among the predictors, ‘poutcome’ has 81.75% null values (36,959 out of 45,211). It was excluded from the analysis given its limited usability. ‘Contact’, with 28.8% null values, was also dropped for irrelevance. For other variables with an insignificant ratio of null values (4% for ‘education’ and less than 1% for ‘job’), all the rows with null values were removed. The study also utilised encoding techniques to transform binary variables into numerical ones. The above actions were conducted to ensure effective data processing on a clean dataset.

To gain a better understanding of the factors influencing customers’ decisions to subscribe to term deposits, 3 methods were applied: PCA (Principal Component Analysis), LR (logistic regression), and a DT (decision tree) model. PCA was used to identify the most impactful variables such as "day" from PC1 and "pdays" from PC2 based on their weights. As the supplementary analysis, a logistic regression model was applied to capture potential non-linear relationships. Key metrics such as precision and recall rate were used to assess the model’s performance along with a more in-depth analysis regarding the strongly imbalanced pattern in the original dataset. Finally, DT was utilised mainly to capture node patterns to generate complementary visual insights.

The second research question builds on the results of the first research question, in which the days and times of the month were investigated to reveal when higher-than-average term deposit subscriptions occurred. This investigation was accomplished through a series of statistical analyses, iteration and clustering, feature engineering and pattern identification of term deposit subscriptions over time.

Finally, k-means clustering was applied by classifying users into 3 groups based on shared traits. The likelihood of subscription was computed for each cluster to identify which cluster carries the most high-potential user segment.

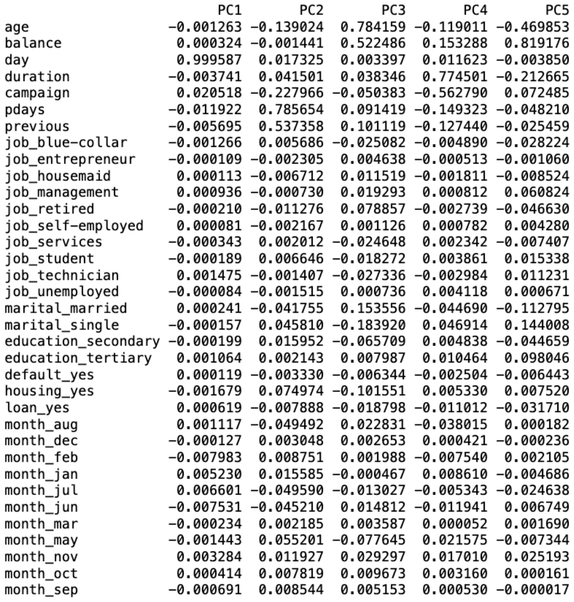
In generating k-means clusters, 3 different n-values were tested. Each of the corresponding cluster plots is shown above. According to the visual representation, when n = 10, there was significant overlap between the 3 clusters. After increasing the n-value from 10 to 25 and 50, the boundaries between the clusters became clearer with much less overlapping, indicating an improved stability and quality of the cluster. For this study, 3 clusters were generated at n = 50 to ensure stable and reliable clustering results. More details can be found in the Appendix.

The study also included Python-generated visualisations such as heatmaps, bar charts, line charts, cluster plots, and box plots to demonstrate the discovered patterns and facilitate interpretation.

DISCUSSION OF FINDINGS

The first research question aims to understand the factors influencing customers' decision to subscribe to term deposits. To explore the question in depth, various statistical methods and machine learning models were applied, such as: Principal Component Analysis, Heatmap, Logistic Regression and a Decision Tree.

In applying Principal Component Analysis (PCA), PC1 stands out as the most informative component as the comparison between plots shows some separation between the term deposit subscribers and non-subscribers. As shown in the below in the PCA table, the variable 'day of the month' in PC1 has an exceedingly high weight coefficient of 0.9996, followed by the variable 'pdays' (0.7856) from PC2 and the variable 'balance' (0.5225).

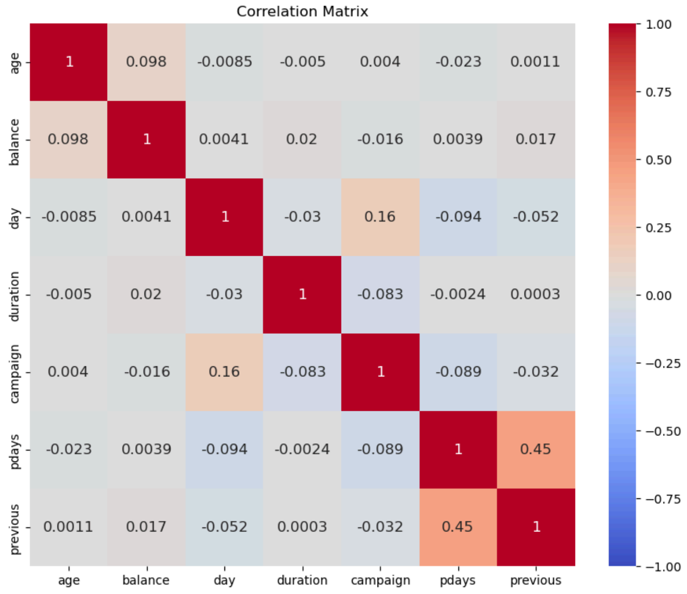


```
                         PC1       PC2       PC3       PC4       PC5
age                -0.001263 -0.139024  0.784159 -0.119011 -0.469853
balance             0.000324 -0.001441  0.522486  0.153288  0.819176
day                 0.999587  0.017325  0.003397  0.011623 -0.003850
duration           -0.003741  0.041501  0.038346  0.774501 -0.212665
campaign            0.020518 -0.227966 -0.050383 -0.562790  0.072485
pdays              -0.011922  0.785654  0.091419 -0.149323 -0.048210
previous           -0.005695  0.537358  0.101119 -0.127440 -0.025459
job_blue-collar    -0.001266  0.005686 -0.025082 -0.004890 -0.028224
job_entrepreneur   -0.000109 -0.002305  0.004638 -0.000513 -0.001060
job_housemaid       0.000113 -0.006712  0.011519 -0.001811 -0.008524
job_management      0.000936 -0.000730  0.019293  0.000812  0.060824
job_retired        -0.000210 -0.011276  0.078857 -0.002739 -0.046630
job_self-employed   0.000081 -0.002167  0.001126  0.000782  0.004280
job_services       -0.000343  0.002012 -0.024648  0.002342 -0.007407
job_student        -0.000189  0.006646 -0.018272  0.003861  0.015338
job_technician      0.001475 -0.001407 -0.027336 -0.002984  0.011231
job_unemployed     -0.000084 -0.001515  0.000736  0.004118  0.000671
marital_married     0.000241 -0.041755  0.153556 -0.044690 -0.112795
marital_single     -0.000157  0.045810 -0.183920  0.046914  0.144008
education_secondary -0.000199  0.015952 -0.065709  0.004838 -0.044659
education_tertiary  0.001064  0.002143  0.007987  0.010464  0.098046
default_yes         0.000119 -0.003330 -0.006344 -0.002504 -0.006443
housing_yes        -0.001679  0.074974 -0.101551  0.005330  0.007520
loan_yes            0.000619 -0.007888 -0.018798 -0.011012 -0.031710
month_aug           0.001117 -0.049492  0.022831 -0.038015  0.000182
month_dec          -0.000127  0.003048  0.002653  0.000421 -0.000236
month_feb          -0.007983  0.008751  0.001988 -0.007540  0.002105
month_jan           0.005230  0.015585 -0.000467  0.008610 -0.004686
month_jul           0.006601 -0.049590 -0.013027 -0.005343 -0.024638
month_jun          -0.007531 -0.045210  0.014812 -0.011941  0.006749
month_mar          -0.000234  0.002185  0.003587  0.000052  0.001690
month_may          -0.001443  0.055201 -0.077645  0.021575 -0.007344
month_nov           0.003284  0.011927  0.029297  0.017010  0.025193
month_oct           0.000414  0.007819  0.009673  0.003160  0.000161
month_sep          -0.000691  0.008544  0.005153  0.000530 -0.000017
```

Figure 1.1 PCA Table of All the Predictors

The heatmap in Figure 1.2 visualises the relationship between the variables in the dataset, where stronger red colours indicate the positive correlation and stronger blue indicates the negative correlation. While assessing the heatmap, we noticed that pdays (days since last contact) and previous (number of contacts performed before this campaign and for this client) has a positive correlation of 0.45.

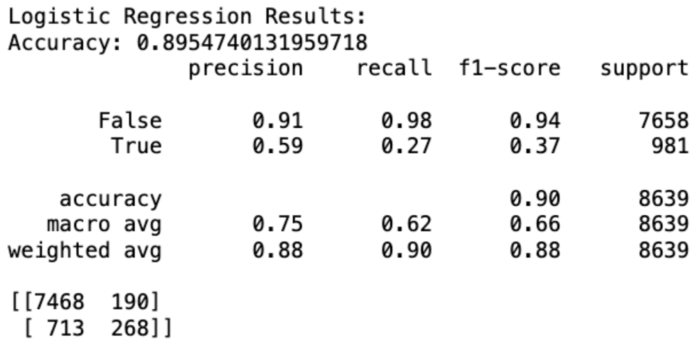
Figure 1.2


Correlation Matrix

	age	balance	day	duration	campaign	pdays	previous
age	1	0.098	-0.0085	-0.005	0.004	-0.023	0.0011
balance	0.098	1	0.0041	0.02	-0.016	0.0039	0.017
day	-0.0085	0.0041	1	-0.03	0.16	-0.094	-0.052
duration	-0.005	0.02	-0.03	1	-0.083	-0.0024	0.0003
campaign	0.004	-0.016	0.16	-0.083	1	-0.089	-0.032
pdays	-0.023	0.0039	-0.094	-0.0024	-0.089	1	0.45
previous	0.0011	0.017	-0.052	0.0003	-0.032	0.45	1

To deepen our understanding of the variable impact of subscribing to a term deposit, we applied a Logistic Regression for predictive modelling. In Figure 1.3 below, we can see that the model has an accuracy of ~89.55%, meaning that it correctly predicted the outcome in the test data set 89.55% of the time.

Figure 1.3

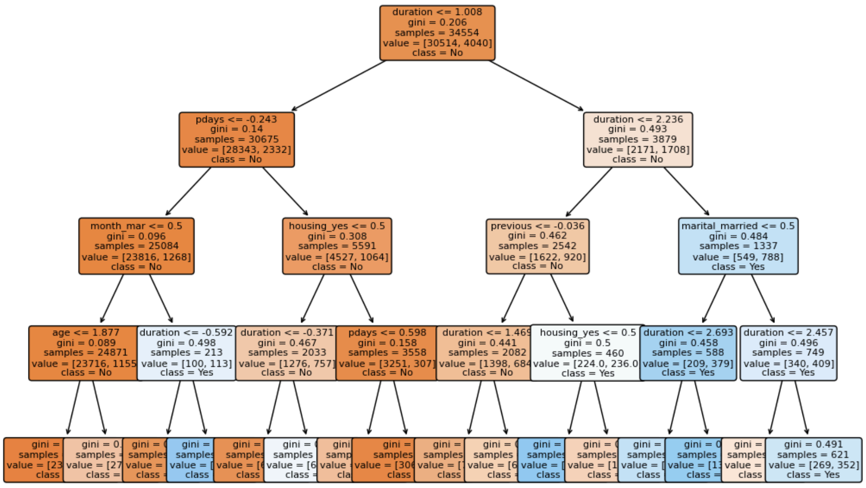


```
Logistic Regression Results:
Accuracy: 0.8954740131959718
              precision    recall  f1-score   support

       False       0.91      0.98      0.94      7658
        True       0.59      0.27      0.37       981

    accuracy                           0.90      8639
   macro avg       0.75      0.62      0.66      8639
weighted avg       0.88      0.90      0.88      8639

[[7468  190]
 [ 713  268]]
```

While assessing the precision score and F1 score for the False group, the non-subscribers and the True group, the subscribers, we can see that the Logistic Regression model predicts the False group more often. We can see that the True group has a precision score of .59, meaning the model is only correctly predicting this group 59% of the time and the recall, which correctly captures the number of subscribers is low, at 27%. This is likely due to an oversampling of the non-subscribers. The confusion matrix offers even more insights, where there is a large False Negative value of 713, where the model is incorrectly predicting non-subscription when it should have been subscription.

Finally, to capture any potential non-linear relationships, we created a Decision Tree in which the model achieved an accuracy of 87.32% (just below that of the Linear Regression). Overall, the decision tree captures more of the actual subscribers.

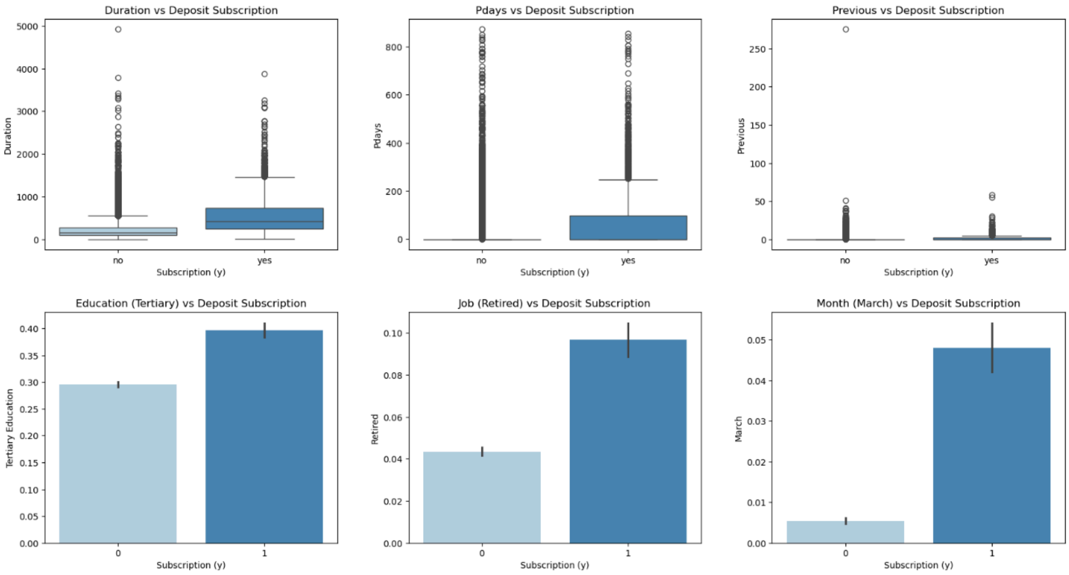
Figure 1.4


duration <= 1.008 gini = 0.206 samples = 34554 value = [30514, 4040] class = No
pdays <= -0.243 gini = 0.14 samples = 30675 value = [28343, 2332] class = No
duration <= 2.236 gini = 0.493 samples = 3879 value = [2171, 1708] class = No
month_mar <= 0.5 gini = 0.096 samples = 25084 value = [23816, 1268] class = No
housing_yes <= 0.5 gini = 0.308 samples = 5591 value = [4527, 1064] class = No
previous <= -0.036 gini = 0.462 samples = 2542 value = [1622, 920] class = No
marital_married <= 0.5 gini = 0.484 samples = 1337 value = [549, 788] class = Yes
age <= 1.877 gini = 0.089 samples = 24871 value = [23716, 1155 class = No
duration <= -0.592 gini = 0.498 samples = 213 value = [100, 113] class = Yes
duration <= -0.371 gini = 0.467 samples = 2033 value = [1276, 757] class = No
pdays <= 0.598 gini = 0.158 samples = 3558 value = [3251, 307] class = No
duration <= 1.469 gini = 0.441 samples = 2082 value = [1398, 684 class = No
housing_yes <= 0.5 gini = 0.5 samples = 460 value = [224.0, 236.0 class = Yes
duration <= 2.693 gini = 0.458 samples = 588 value = [209, 379] class = Yes
duration <= 2.457 gini = 0.496 samples = 749 value = [340, 409] class = Yes
gini = 0.491 samples = 621 value = [269, 352] class = Yes

Reviewing the Decision Tree in Figure 1.4, duration is the top node and appears to be an important factor. To understand the Decision tree, we can see that if the duration (of a call) is less than or equal to 1.008 seconds, the model would suggest non-subscription. This is fairly intuitive in that a shorter call would likely mean a quick 'no' from a customer.

On the opposite side, if the duration is greater than 1.008 seconds then more variables are introduced such as; pdays (days since last contact), recent contact and less recent contact. Here, we can understand that if pdays is less than or equal to -0.243, then the model would suggest non-subscription. Alternatively, if the customer was contacted less recently (greater than -0.243 to be exact) then variables like housing loans, marital status, month of the campaign and age can contribute to the prediction.

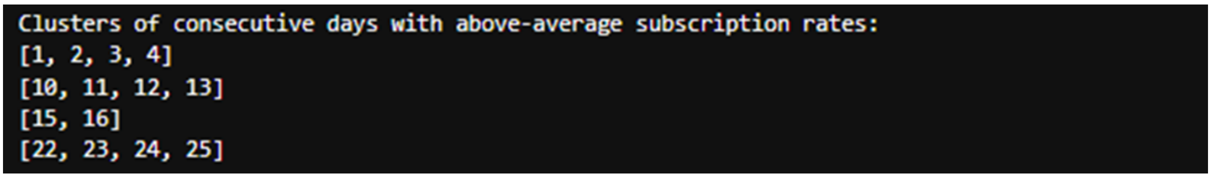
Duration also shows the strongest correlation with deposit by showing a 0.39 correlation coefficient, as seen in Figure 1.5. This interesting finding reiterates that longer call durations positively influence the term deposit.



In conclusion of the first research question, the analysis investigated the factors influencing customers' decision to subscribe to term deposits, and found that call duration is an important factor in predicting term deposit subscriptions. Additionally, the timing of the contact, both day and pdays, also has significance, which could mean that there is greater success for subscriptions on certain days of the month and days between contacts could be influential for marketing campaigns.

The second question researched is: *"When running marketing campaigns, What are the days and times of the month that show higher-than-average term deposit subscriptions?"* By analysing historical campaign data, the study uncovered the impact of both the day of the month and the day of the week on subscription behaviour. The insights generated from this analysis will allow the bank to strategically target high-performing periods and maximise campaign effectiveness.

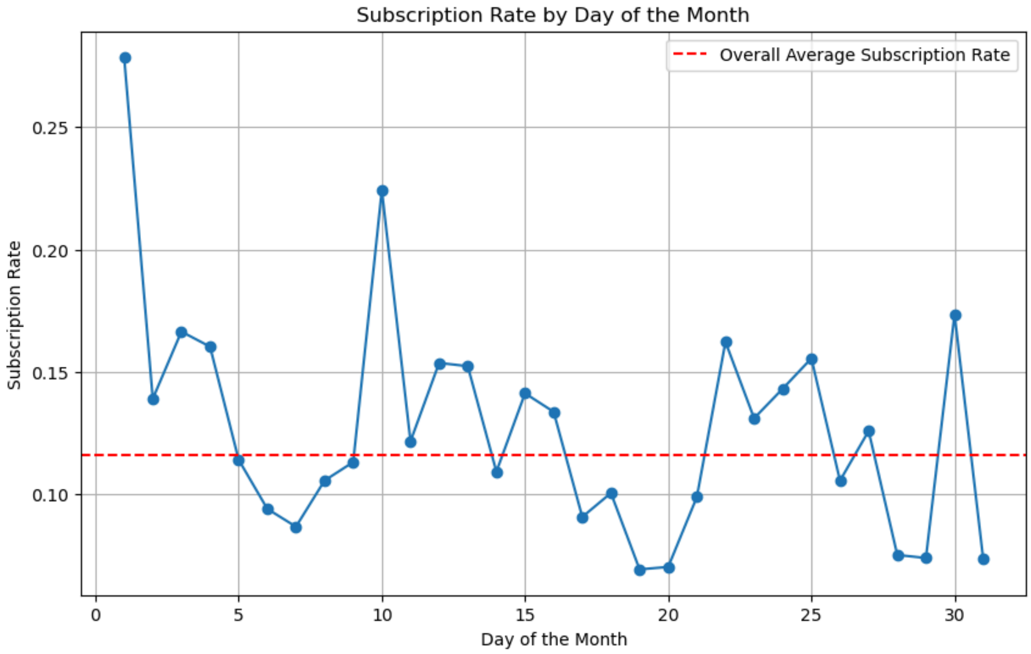
Figure 2.1



```
Clusters of consecutive days with above-average subscription rates:
[1, 2, 3, 4]
[10, 11, 12, 13]
[15, 16]
[22, 23, 24, 25]
```

1. Day of the Month Analysis

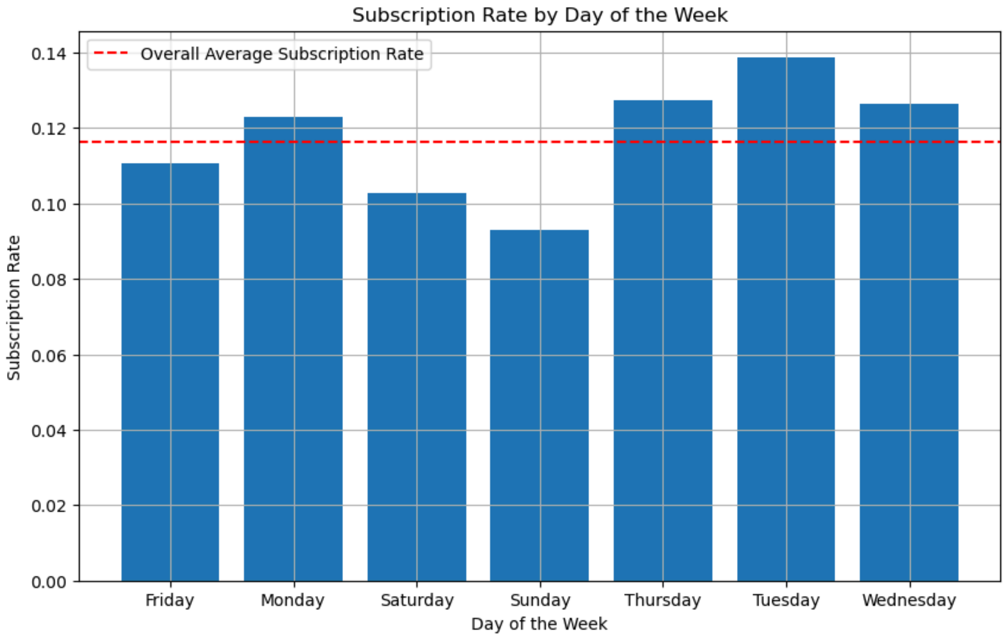
Figure 2.2

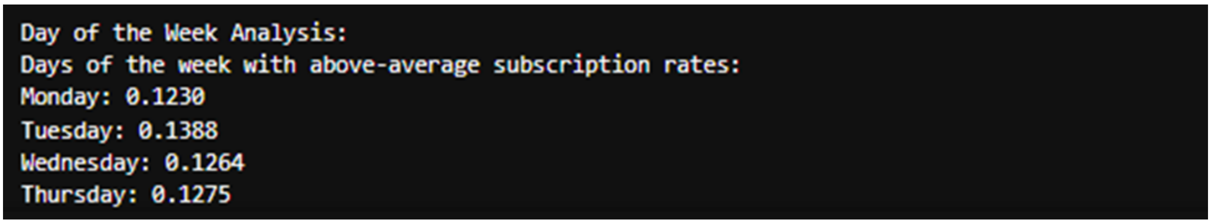


The above line plot reveals trends in subscription rates across different days of the month. The overall average subscription rate, indicated by the red dashed line, provides a benchmark to evaluate daily performance. This analysis reveals several noteworthy patterns, particularly in how certain days of the month consistently outperform others in terms of subscription rates. Key insights include

- **Clusters of Consecutive Above-Average Days:** The clusters of consecutive days with above-average subscription rates were classified based on the calculated daily subscription rates exceeding the overall average subscription rate. Days with consecutive above-average performance were grouped to form these clusters.
    - **[1, 2, 3, 4]:** Early days of the month show significantly higher subscription rates, possibly due to customer financial readiness post-salary.
    - **[10, 11, 12, 13]:** A mid-month peak suggests effective campaign targeting during this period.
    - **[15, 16]:** Mid-month performance remains promising, though with shorter clusters.
    - **[22, 23, 24, 25]:** Late-month clusters suggest another uptick in customer engagement.

Figure 2.3





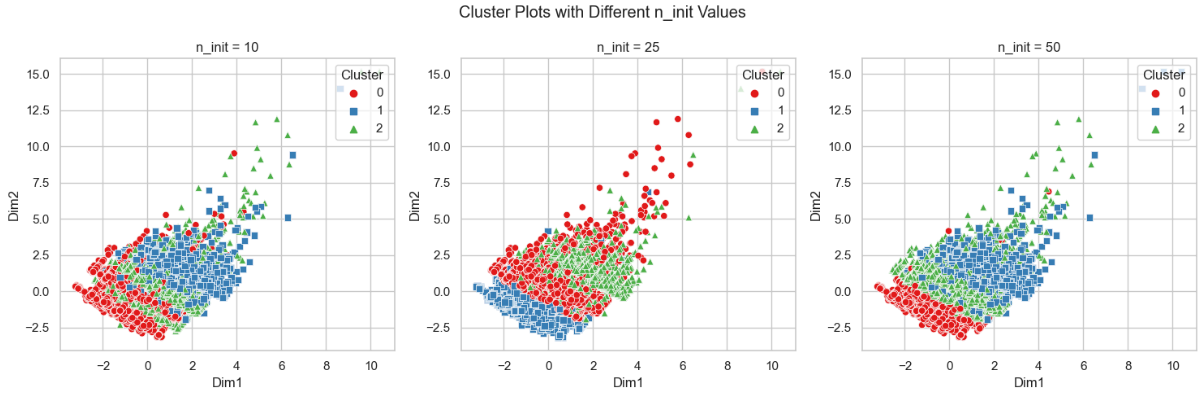
```
Day of the Week Analysis:
Days of the week with above-average subscription rates:
Monday: 0.1230
Tuesday: 0.1388
Wednesday: 0.1264
Thursday: 0.1275
```

2. Day of the Week Analysis

The subscription rates aggregated by day of the week provide further insight into weekday performance, as visualised in the bar chart below:

Based on the statistical results, there is a clear variation in user responsiveness across different days of the week. Tuesday stands out as the peak day with the highest subscription rate at 13.88%, followed by Thursday, Wednesday, and Monday with subscription rates as 12.75%, 12.64%, and 12.30% respectively. These four weekdays all have subscription rates that go above the benchmark line, the overall average.

This demonstrates that users have a stronger propensity to make a term deposit in the first half of the week. Conversely, Friday, Saturday, and Sunday exhibit subscription rates below the benchmark line. That may indicate a noticeable decline in interest, engagement, or commitment from customers as the weekend approaches. This intriguing pattern, therefore, highlights the strategic importance of planning and organising the marketing campaigns at an optimal schedule.



Finally, for the third research question focused on future targeted marketing, k-means clustering was applied to pinpoint the user segment that carries the highest potential. A total of 3 clusters were created based on finance-related attributes after calculating the best k-value and locating the most suitable n-value (n=50). Unique user profiles can be found among the 3 clusters. Users from Cluster 1 are characterised as "customers with limited resources". They have minimal account balances, heavy personal or housing loans, and the lowest likelihood of subscribing to term deposits. Users from cluster 3 have significant housing loan commitments and moderate balances. They present a low but marginally higher subscription likelihood compared to Cluster 1. Conversely, users from cluster 2 are characterised as "wealthy, risk-averse customers". Those users hold the highest balances exceeding the average, no or insignificant loans, and the highest propensity to subscribe.

The above findings highlight how an individual's financial status and external debt pressure can directly impact their personal financial decision. Capability of making term deposits usually indicates a stable cash flow. In that case, users will have extra funds for savings. Those who have to handle the stress from low cash flow and the high pressure from personal or housing debts, on the contrary, may be much more inclined to allocate their cash flow toward daily expenses rather than term deposits. This reveals the importance of targeting wealthy, risk-averse customers as the most prospective user segment who possess the financial capacity for long-term investments with tailored services dedicated to their needs and expectations.

BUSINESS RECOMMENDATIONS

1. Find the Best Timing for Campaigns

Our findings, supported by previous studies (Xie et al., 2023), suggest that running campaigns on Tuesdays and Thursdays and at the beginning or 10th day of each month leads to higher response rates. Therefore, planning and implementing campaigns at optimal times can maximise the success of deposit subscriptions.

2. Increase Call Engagement

As indicated in our findings, longer call duration is positively associated with a greater propensity for deposit description. Moreover, customers show a higher acceptability to get contacted in the new campaign if they were contacted before. Thus, to ensure subscription success, it is crucial to elevate the call engagement. The study recommends running pre-call surveys to understand customer needs and pain points, having personalised call scripts to ensure engagement, and building trust through follow-ups via emails or newsletters with tailored offers (KIVO, 2023). Ensuring strict compliance with GDPR is also essential to avoid penalties and violation of user privacy (Intelemark,. 2024).

3. Precise Customer Segmentation

Utilising precise customer segmentation can significantly enhance the effectiveness of telemarketing efforts by allowing tailored messaging for each segment. For Cluster 1, focus on promoting financial education and low-risk savings options to build trust. For Cluster 2, prioritise personalised outreach with premium term deposit and investment offerings that align with their wealth and risk aversion. For Cluster 3, target messaging around debt management solutions and moderate-growth investment products that suit their mid-level financial stability. By matching offers to each cluster's needs, telemarketing campaigns can achieve higher engagement, conversion rates, and customer satisfaction.

LIMITATIONS AND FUTURE CONSIDERATIONS

The dataset that this study uses dates back to the early 2000s. That may limit the applicability of the findings given the recent changes in the business landscape and evolvement of the marketing techniques. Moreover, since the dataset is strongly imbalanced with over 77% of the data focusing on non-subscriptions, the analytical implications, and the model could be biased and unable to capture some of the patterns found in recent studies. For instance, previous empirical studies have shown a high association between married, mid-aged employed or senior-aged retired customers with no personal or house loans with term deposit subscription behaviours. Although this pattern is not directly found in our analysis, we recommend the bank keep leveraging analytical methods to understand its target users and curate tailored services.

However, the insights regarding optimised campaign timing and precise customer segmentation remain valuable. Future studies should focus more on validating these findings with more recent data and exploring possibilities to combine telemarketing with other promotional techniques for business growth.

REFERENCES

Xie, C., Zhang, J.-L., Zhu, Y., Xiong, B. and Wang, G.-J. (2023). How to improve the success of bank telemarketing? Prediction and interpretability analysis based on machine learning. *Computers & Industrial Engineering*, 175, p.108874. doi:https://doi.org/10.1016/j.cie.2022.108874.

Ilham, A., Khikmah, L., Indra, Ulumuddin and Bagus Ary Indra Iswara, I. (2019). Long-term deposits prediction: a comparative framework of classification model for predicting the success of bank telemarketing. *Journal of Physics: Conference Series*, 1175, p.012035. doi:https://doi.org/10.1088/1742-6596/1175/1/012035.

Asare-Frempong, J. and Jayabalan, M. (2017). Predicting customer response to bank direct telemarketing campaigns. *2017 International Conference on Engineering Technology and Technopreneurship (ICE2T)*. [online] doi:https://doi.org/10.1109/ice2t.2017.8215961.

Welle, D. (2015). The fight for Portugal's banks. [online] dw.com. Available at: https://www.dw.com/en/the-fight-for-portugals-banks/a-18328627 [Accessed 15 Nov. 2024].

IBS Intelligence. (2024). Big data analytics set to transform banking with projected $745bn market by 2030. [online] Available at: https://ibsintelligence.com/ibsi-news/big-data-analytics-set-to-transform-banking-with-projected-745bn-market-by-2030/.

Statista. (2024). Telemarketing - Worldwide | Statista Market Forecast. [online] Available at: https://www.statista.com/outlook/amo/advertising/direct-messaging-advertising/telemarketing/worldwide?currency=usd#global-comparison [Accessed 15 Nov. 2024].

KIVO (2023). If you're looking for a powerful sales tool to help you generate leads, qualify prospects, and convert customers, then telemarketing is worth considering. By using telephone communication to create direct and personalized conversations with your target audience, telemarketing allows you to reach out. [online] Linkedin.com. Available at: https://www.linkedin.com/pulse/how-enhance-your-sales-strategy-telemarketing-mtckivo/.

Intelemark. (2024). About Us. [online] Available at: https://www.intelemark.com/blog/the-impact-of-gdpr-on-b2b-telemarketing-strategies/.

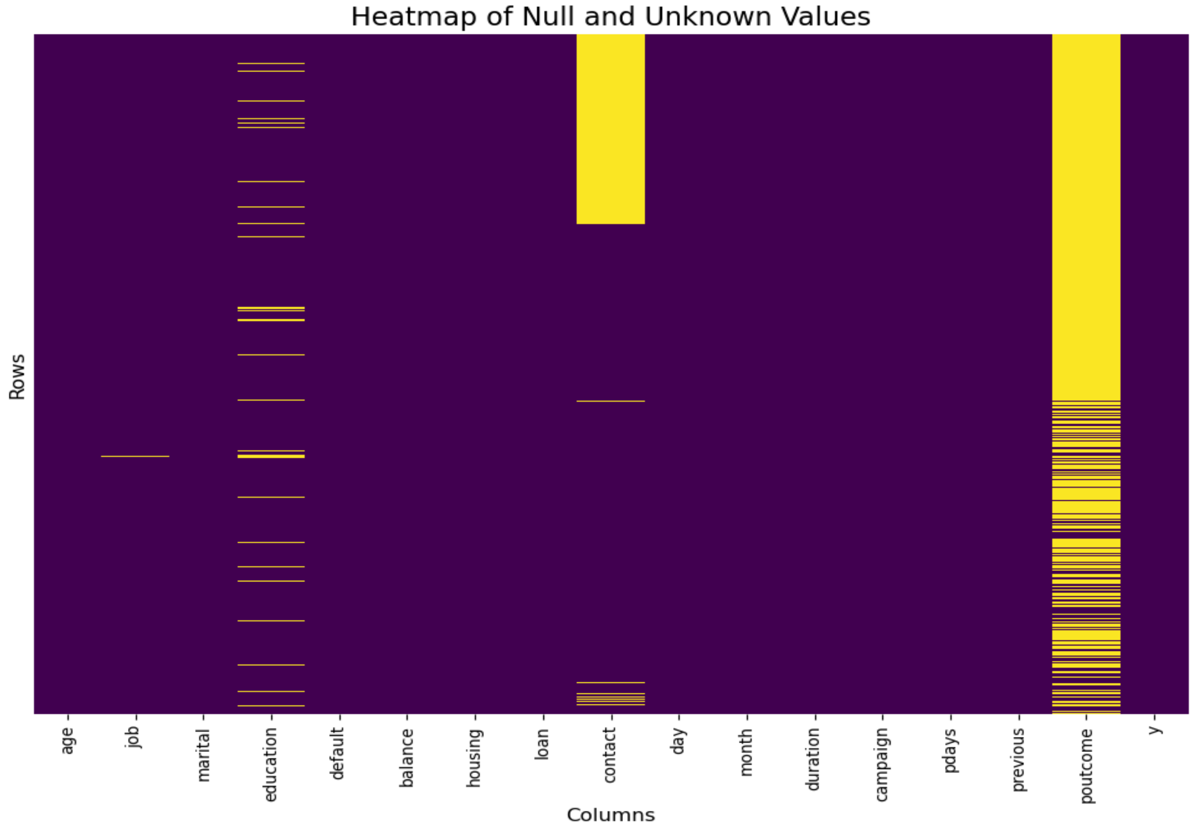
APPENDIX

Data Cleaning and Preprocessing

```
import pandas as pd

# Load the dataset

file_path = 'Bank Data.csv'

data = pd.read_csv(file_path)

# Check for null values

null_values = data.isnull().sum()

# Check for 'unknown' values in each column

unknown_values = data.apply(lambda x: (x == 'unknown').sum())

# Combine the results into a single DataFrame for clarity

null_and_unknown = pd.DataFrame({

      'Null Values': null_values,

      'Unknown Values': unknown_values

})

# Display the results

print("Null and Unknown Values in the Dataset:")

print(null_and_unknown)

# Creating a boolean matrix for null and 'unknown' values

boolean_matrix = data.isnull() | data.applymap(lambda x: x == 'unknown')
```

```
# Display the boolean matrix (optional, you can remove this if not needed
in the output)

print("Boolean Matrix for Null and Unknown Values:")

print(boolean_matrix)

# Plot the heatmap with the boolean matrix

plt.figure(figsize=(12, 8))

sns.heatmap(boolean_matrix, yticklabels=False, cbar=False, cmap="viridis")

# Add titles and labels

plt.title('Heatmap of Null and Unknown Values', fontsize=16)

plt.xlabel('Columns', fontsize=12)

plt.ylabel('Rows', fontsize=12)

plt.show()
```

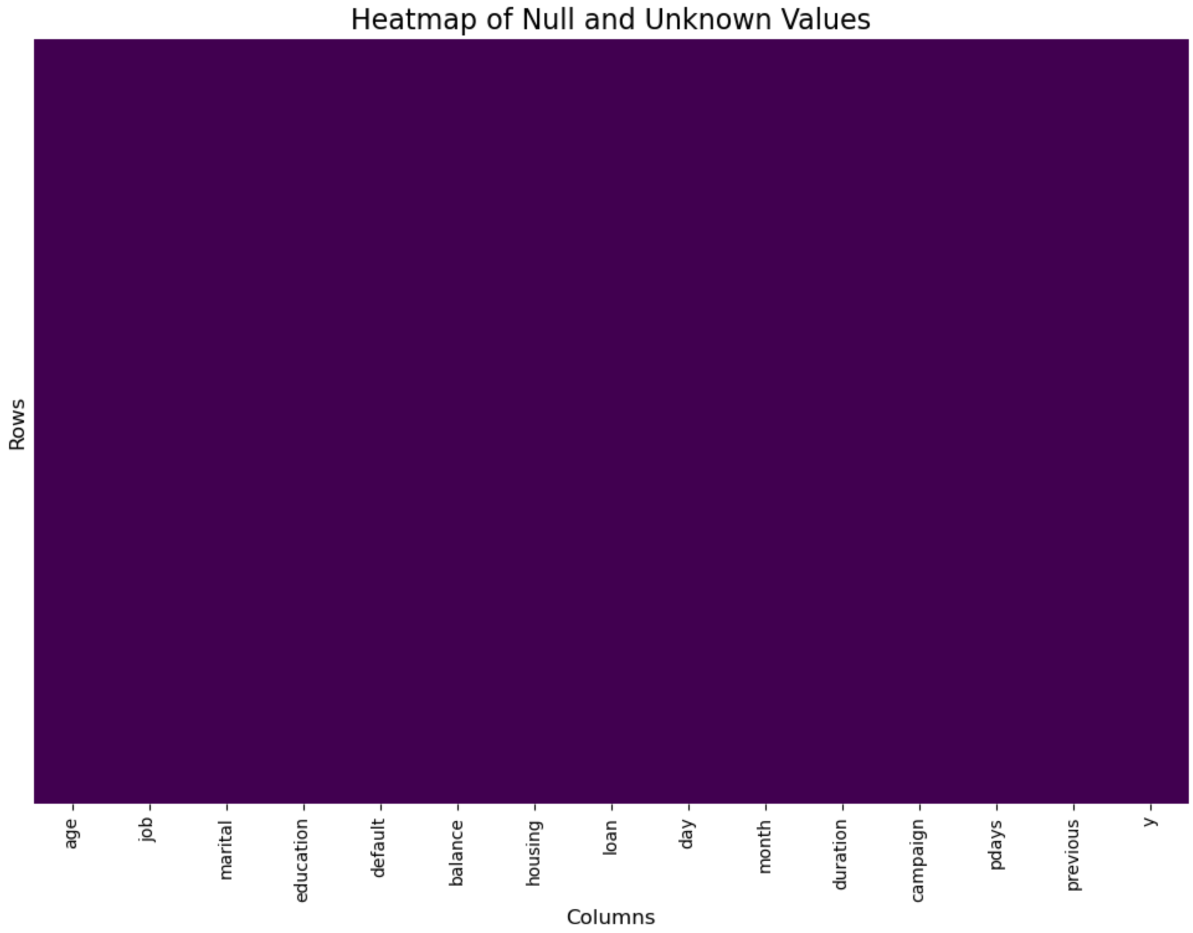
Output:



```
# Drop the 'poutcome' column as it has many missing or 'unknown' values
data_cleaned = data.drop(columns=['poutcome'])

# Display the updated dataset structure
data_cleaned.info()

# Creating a boolean matrix for null and 'unknown' values
boolean_matrix = data_cleaned.isnull() | data_cleaned.applymap(lambda x: x
== 'unknown')

# Display the boolean matrix (optional, you can remove this if not needed
in the output)
print("Boolean Matrix for Null and Unknown Values:")
print(boolean_matrix)

# Plot the heatmap with the boolean matrix
plt.figure(figsize=(12, 8))
sns.heatmap(boolean_matrix, yticklabels=False, cbar=False, cmap="viridis")
```

```python
# Drop the 'poutcome' column
data_cleaned = data.drop(columns=['poutcome'])

# Display the updated dataset structure
data_cleaned.info()

data_cleaned = data_cleaned[
    (data_cleaned['education'] != 'unknown') & (data_cleaned['job'] !=
'unknown')
]

# Display the updated dataset structure to confirm changes
data_cleaned.info()

data_cleaned = data_cleaned[
    (data_cleaned['education'] != 'unknown') & (data_cleaned['job'] !=
'unknown')
]

# Drop the 'contact' column
data_cleaned = data_cleaned.drop(columns=['contact'])

# Plot the heatmap with the boolean matrix
plt.figure(figsize=(12, 8))
sns.heatmap(boolean_matrix, yticklabels=False, cbar=False, cmap="viridis")

# Add titles and labels
plt.title('Heatmap of Null and Unknown Values', fontsize=16)
plt.xlabel('Columns', fontsize=12)
plt.ylabel('Rows', fontsize=12)

plt.show()
```

Output post data cleaning



Which factors influence customers' decision to subscribe to term deposits?

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler, OneHotEncoder

from sklearn.decomposition import PCA

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
```

```
from sklearn.tree import plot_tree


# Load the data

data=pd.read_csv("Cleaned_Bank_Data.csv")


# Check for missing values

print(data.isnull().sum())

# Convert categorical variables to numerical

categorical_cols = ['job', 'marital', 'education', 'default', 'housing',
'loan', 'month', 'y']

data = pd.get_dummies(data, columns=categorical_cols, drop_first=True)

# Standardize the data

scaler = StandardScaler()

numerical_cols = ['age', 'balance', 'duration', 'campaign', 'pdays',
'previous']

X[numerical_cols] = scaler.fit_transform(X[numerical_cols])

# Split data into training and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Principal Component Analysis with added components

pca = PCA(n_components=5)  # Increase the number of components

principal_components = pca.fit_transform(X_train)

pca_df = pd.DataFrame(data=principal_components, columns=['PC1', 'PC2',
'PC3', 'PC4', 'PC5'])

pca_df['y_yes'] = y_train

# Visualize for multiple components

sns.pairplot(pca_df, hue='y_yes', palette='colorblind', vars=['PC1',
'PC2', 'PC3'], diag_kind='kde')

plt.suptitle('Pair Plot of Principal Components', y=1.02)

plt.show()
```

```
# Analyze Feature Importance

loadings = pd.DataFrame(pca.components_.T, columns=['PC1', 'PC2', 'PC3',
'PC4', 'PC5'], index=X.columns)

print(loadings)

# Visualize feature importance for PC1

plt.figure(figsize=(10, 6))

plt.bar(loadings.index, loadings['PC1'])

plt.xticks(rotation=45, ha='right')

plt.xlabel('Features')

plt.ylabel('Loadings on PC1')

plt.title('Feature Importance for PC1')

plt.show()

# Visualize all PCA results

plt.figure(figsize=(8, 6))

sns.scatterplot(x='PC1', y='PC2', hue='y_yes', data=pca_df,

                palette='colorblind', style='y_yes', markers=['o', '^'],
alpha=0.6)

sns.kdeplot(x='PC1', y='PC2', hue='y_yes', data=pca_df, levels=5,
alpha=0.3)

plt.xlim(-10, 10)

plt.ylim(-5, 10)

plt.title('PCA of Bank Marketing Data')

plt.xlabel('Principal Component 1')

plt.ylabel('Principal 333Component 2')

plt.show()

# Create a heatmap

numeric_data = data.select_dtypes(include=[np.number])

corr_matrix = numeric_data.corr()

plt.figure(figsize=(10, 8))
```

```
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1,
annot_kws={"fontsize": 12})

plt.title("Correlation Matrix")

plt.show()

# Logistic Regression

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

logreg = LogisticRegression(max_iter=1000)

logreg.fit(X_train, y_train)

y_pred_logreg = logreg.predict(X_test)

# Print results

print("Logistic Regression Results:")

print("Accuracy:", accuracy_score(y_test, y_pred_logreg))

print(classification_report(y_test, y_pred_logreg))

print(confusion_matrix(y_test, y_pred_logreg))

# Create and train the model

logreg = LogisticRegression(max_iter=1000)

logreg.fit(X_train, y_train)

# Make predictions on the test data

y_pred = logreg.predict(X_test)

# Evaluate model

print("Accuracy:", accuracy_score(y_test, y_pred))

print(classification_report(y_test, y_pred))

# Decision Tree

tree = DecisionTreeClassifier()

tree.fit(X_train, y_train)

y_pred_tree = tree.predict(X_test)

# Print Decsion Tree ouput

print("\nDecision Tree Results:")
print("Accuracy:", accuracy_score(y_test, y_pred_tree))

print(classification_report(y_test, y_pred_tree))

print(confusion_matrix(y_test, y_pred_tree))

# Visualize Decision Tree

plt.figure(figsize=(12, 8))

plot_tree(tree, feature_names=X.columns, class_names=['No', 'Yes'],
filled=True, rounded=True)

plt.show()

# Decision Tree with Pruning

tree = DecisionTreeClassifier(max_depth=4, min_samples_leaf=50)  # Adjust
parameters

tree.fit(X_train, y_train)

y_pred_tree = tree.predict(X_test)

print("\nDecision Tree Results:")

print("Accuracy:", accuracy_score(y_test, y_pred_tree))

print(classification_report(y_test, y_pred_tree))

print(confusion_matrix(y_test, y_pred_tree))

# Visualize Decision Tree

plt.figure(figsize=(12, 8))

plot_tree(tree, feature_names=X.columns, class_names=['No', 'Yes'],

          filled=True, rounded=True, fontsize=8)  # Adjust fontsize

plt.show()
```

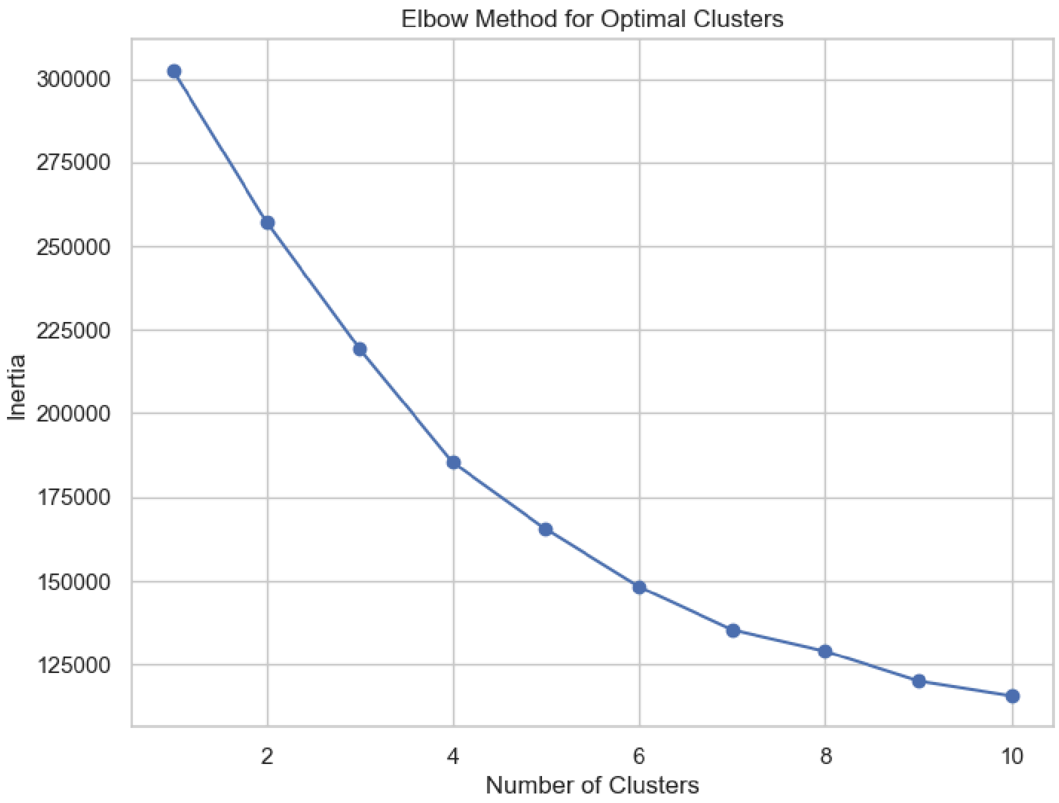
What are the days and times of the month that show higher-than-average term deposit subscriptions

```
# Investigating the timing of term subscriptions

import pandas as pd
import matplotlib.pyplot as plt


#Convert 'day' to day of the week

data['day_of_week'] = pd.to_datetime(data['day'],
format='%d').dt.day_name()


# Aggregate subscription rate by day of the month

daily_subscriptions = data.groupby('day')['y_yes'].mean()


# Calculate the overall average term deposit subscription rate

overall_avg_subscription = data['y_yes'].mean()


# Days with above-average subscription rates

above_average_days = daily_subscriptions[daily_subscriptions >
overall_avg_subscription]


# Visualize the results as day of the month

plt.figure(figsize=(10, 6))

plt.plot(daily_subscriptions.index, daily_subscriptions.values,
marker='o', linestyle='-')

plt.axhline(y=overall_avg_subscription, color='r', linestyle='--',
label='Overall Average Subscription Rate')

plt.xlabel('Day of the Month')

plt.ylabel('Subscription Rate')
plt.title('Subscription Rate by Day of the Month')
plt.legend()
plt.grid(True)

# Highlight the above-average days
for day in above_average_days.index:
    plt.text(day, above_average_days[day], str(day), ha='center', va='bottom')

plt.show()

# Aggregate subscription rate by day of the week
weekly_subscriptions = data.groupby('day_of_week')['y_yes'].mean()

# Calculate the overall average subscription rate for day of the week
overall_avg_subscription_weekly = data['y_yes'].mean()

# Identify days of the week with above-average subscription rates
above_average_days_weekly = weekly_subscriptions[weekly_subscriptions > overall_avg_subscription_weekly]

#  Visualise the results in day of the week format
plt.figure(figsize=(10, 6))
plt.bar(weekly_subscriptions.index, weekly_subscriptions.values)
plt.axhline(y=overall_avg_subscription_weekly, color='r', linestyle='--',
label='Overall Average Subscription Rate')
plt.xlabel('Day of the Week')
plt.ylabel('Subscription Rate')
plt.title('Subscription Rate by Day of the Week')
plt.legend()
plt.grid(True)

# Sort in sequential order
weekly_subscriptions.index = pd.Categorical(weekly_subscriptions.index,
                                            categories=['Monday',
'Tuesday', 'Wednesday',
                                                        'Thursday',
'Friday', 'Saturday', 'Sunday'],
                                            ordered=True)
weekly_subscriptions = weekly_subscriptions.sort_index()

plt.bar(weekly_subscriptions.index, weekly_subscriptions.values)
plt.axhline(y=overall_avg_subscription_weekly, color='r', linestyle='--',
label='Overall Average Subscription Rate')
plt.xlabel('Day of the Week')
plt.ylabel('Subscription Rate')
plt.title('Subscription Rate by Day of the Week')
plt.legend()
plt.grid(True)

# Find clusters of consecutive days with above-average subscription rates
consecutive_days = []
current_cluster = []
for i in range(len(daily_subscriptions)):
    day = daily_subscriptions.index[i]
    if daily_subscriptions[day] > overall_avg_subscription:
        current_cluster.append(day)
    else:
        if len(current_cluster) > 1:
            consecutive_days.append(current_cluster)
        current_cluster = []
if len(current_cluster) > 1:
    consecutive_days.append(current_cluster)

if consecutive_days:
    print("Clusters of consecutive days with above-average subscription rates:")
    for cluster in consecutive_days:
        print(cluster)
else:
    print("No clusters of consecutive days with above-average subscription rates found.")


# Day of the week analysis

print("\nDay of the Week Analysis:")

# Identify specific days of the week with above-average subscription rates

above_average_days_weekly = weekly_subscriptions[weekly_subscriptions > overall_avg_subscription_weekly]

if above_average_days_weekly.any():

    print("Days of the week with above-average subscription rates:")

    for day, rate in above_average_days_weekly.items():

        print(f"{day}: {rate:.4f}")

else:

    print("No days of the week with above-average subscription rates found.")
```

For future telemarketing campaigns, which user segment should the bank target?

```
# Import libraries
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.decomposition import PCA
```

```
# Load and select columns
data = pd.read_csv('C:/Users/vijai/desktop/Cleaned_Bank_Data.csv')
columns = ['balance', 'housing', 'loan', 'y', 'job', 'marital',
'education']
data = data[columns]

# Encode categorical features
label_enc = LabelEncoder()
for col in ['housing', 'loan', 'y', 'job', 'marital', 'education']:
    data[col] = label_enc.fit_transform(data[col])

# Standardise the data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

# Different n_init values for KMeans
n_init_values = [10, 25, 50]

# Create plot figure
fig, axes = plt.subplots(1, len(n_init_values), figsize=(15, 5))
fig.suptitle('Cluster Plots with Different n_init Values')

# Reduce data to 2D
pca = PCA(n_components=2)
data_pca = pca.fit_transform(data_scaled)

# Loop through n_init values
for i, n_init in enumerate(n_init_values):
    # KMeans clustering
    kmeans = KMeans(n_clusters=3, n_init=n_init, random_state=42)
    clusters = kmeans.fit_predict(data_scaled)

    # Prepare plot data
    data_pca_df = pd.DataFrame(data_pca, columns=['Dim1', 'Dim2'])
    data_pca_df['Cluster'] = clusters

    # Plot the clusters
    sns.scatterplot(
        ax=axes[i],
        x='Dim1', y='Dim2',
        hue='Cluster',
        data=data_pca_df,
        palette='Set1',
        style='Cluster',
        markers=['o', 's', '^']
    )
    axes[i].set_title(f'n_init = {n_init}')
    axes[i].legend(title='Cluster', loc='upper right')

# Show plot
plt.tight_layout()
plt.subplots_adjust(top=0.85)
plt.show()
```



```
int = 10

Cluster 1 Insights:
Average Balance: 761.7915259009009
Housing Loan Ratio: 0.6040259009009009
Personal Loan Ratio: 1.0
Subscription Likelihood: 0.06644144144144144
Average Job Level: 4.088400900900901
Average Marital Status: 1.1019144144144144
Average Education Level: 1.1068412162162162

Cluster 2 Insights:
Average Balance: 1763.1281127740172
Housing Loan Ratio: 0.0009315034465627522
Personal Loan Ratio: 0.00018630068931255045
Subscription Likelihood: 0.18257467552629944
Average Job Level: 4.814879215053096
Average Marital Status: 1.1854312860957585

Average Education Level: 1.2302676519903124


Cluster 3 Insights:

Average Balance: 1234.9186430501352

Housing Loan Ratio: 1.0

Personal Loan Ratio: 0.0

Subscription Likelihood: 0.08050635444811367

Average Job Level: 3.9577204042829983

Average Marital Status: 1.174171920344241
Average Education Level: 1.0998198739117382


int = 25


Cluster 1 Insights:

Average Balance: 761.7915259009009

Housing Loan Ratio: 0.6040259009009009

Personal Loan Ratio: 1.0

Subscription Likelihood: 0.06644144144144144

Average Job Level: 4.088400900900901

Average Marital Status: 1.1019144144144144

Average Education Level: 1.1068412162162162


Cluster 2 Insights:

Average Balance: 1763.1281127740172

Housing Loan Ratio: 0.0009315034465627522

Personal Loan Ratio: 0.00018630068931255045

Subscription Likelihood: 0.18257467552629944

Average Job Level: 4.814879215053096

Average Marital Status: 1.1854312860957585

Average Education Level: 1.2302676519903124
```

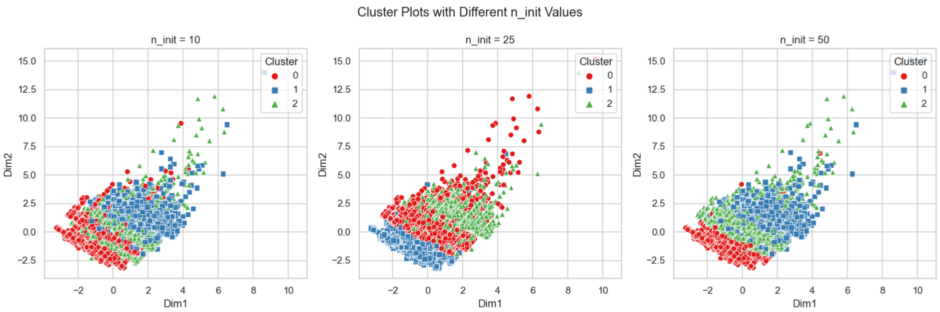
```
Cluster 3 Insights:
Average Balance: 1234.9186430501352
Housing Loan Ratio: 1.0
Personal Loan Ratio: 0.0
Subscription Likelihood: 0.08050635444811367
Average Job Level: 3.9577204042829983
Average Marital Status: 1.174171920344241
Average Education Level: 1.0998198739117382

int = 50

Cluster 1 Insights:
Average Balance: 761.7915259009009
Housing Loan Ratio: 0.6040259009009009
Personal Loan Ratio: 1.0
Subscription Likelihood: 0.06644144144144144
Average Job Level: 4.088400900900901
Average Marital Status: 1.1019144144144144
Average Education Level: 1.1068412162162162

Cluster 2 Insights:
Average Balance: 1763.1281127740172
Housing Loan Ratio: 0.0009315034465627522
Personal Loan Ratio: 0.00018630068931255045
Subscription Likelihood: 0.18257467552629944
Average Job Level: 4.814879215053096
Average Marital Status: 1.1854312860957585
Average Education Level: 1.2302676519903124

Cluster 3 Insights:
Average Balance: 1234.9186430501352
Housing Loan Ratio: 1.0
```

```
Personal Loan Ratio: 0.0
Subscription Likelihood: 0.08050635444811367
Average Job Level: 3.9577204042829983
Average Marital Status: 1.174171920344241
Average Education Level: 1.0998198739117382
```

```
# Load and select columns
data = pd.read_csv('C:/Users/vijai/desktop/Cleaned_Bank_Data.csv')
columns = ['balance', 'housing', 'loan', 'y', 'job', 'marital',
'education']
data = data[columns]

# Encode categorical variables
label_enc = LabelEncoder()
for col in ['housing', 'loan', 'y', 'job', 'marital', 'education']:
    data[col] = label_enc.fit_transform(data[col])

# Standardise the data
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

# Elbow method for optimal clusters
inertia = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(data_scaled)
    inertia.append(kmeans.inertia_)
```

```
plt.figure(figsize=(8, 6))
plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method')
plt.show()

# Define initial centroids
initial_centroids = np.array([
    [765.49, 0.60, 1.00, 0.07, 1, 1, 1],
    [1748.44, 0.00, 0.00, 0.18, 2, 0, 2],
    [1245.73, 1.00, 0.00, 0.08, 0, 2, 0]
])
initial_centroids_scaled = scaler.transform(initial_centroids)

# Experiment with n_init values
for n_init_val in [10, 25, 50, 100]:
    print(f"\nClustering with n_init = {n_init_val}")
    kmeans = KMeans(n_clusters=3, init=initial_centroids_scaled,
n_init=n_init_val, random_state=42)
    data['Cluster'] = kmeans.fit_predict(data_scaled)

    # Cluster summary
    cluster_summary = data.groupby('Cluster').mean()
    print("Cluster Summary:")
    print(cluster_summary)

    # Pairplot for clusters
    sns.pairplot(data, hue='Cluster', palette='Set1')
    plt.suptitle(f'Cluster Visualization (n_init = {n_init_val})', y=1.02)
    plt.show()
```

```
# Cluster insights
for i in range(3):
    print(f"\nCluster {i+1} Insights:")
    print(f"Avg Balance: {cluster_summary.loc[i, 'balance']}")
    print(f"Housing Loan: {cluster_summary.loc[i, 'housing']}")
    print(f"Personal Loan: {cluster_summary.loc[i, 'loan']}")
    print(f"Subscription Likelihood: {cluster_summary.loc[i, 'y']}")
    print(f"Avg Job Level: {cluster_summary.loc[i, 'job']}")
    print(f"Avg Marital Status: {cluster_summary.loc[i, 'marital']}")
    print(f"Avg Education Level: {cluster_summary.loc[i, 'education']}")
```


Cluster Plots with Different n_init Values
n_init = 10
n_init = 25
n_init = 50
Cluster
Dim2
Dim1