\*\*Student Name:\*\* [Your Name]

\*\*Student ID:\*\* [Your ID]

\*\*Course:\*\* [Course Name]

\*\*Date:\*\* [Date]

**Objectives**

The primary objectives of this project are as follows:

1. **Share of Attracted Clients:** Determine the proportion of clients attracted in the source data.

2. **Mean Values of Numerical Features:** Calculate the mean values of numerical features among attracted clients.

3. **Average Call Duration:** Identify the average call duration for attracted clients.

4. **Average Age Among Attracted Clients:** Analyze the average age among attracted clients, with a particular focus on unmarried clients.

5. **Average Age and Call Duration by Job:** Explore the average age and call duration for different types of client employment.

**Methodology**

* **Data Source**

The dataset used in this project was sourced from Kaggle by Henrique Yamahata. It includes a comprehensive set of attributes related to clients, their interactions with the bank, and other relevant information.

* **Data Preprocessing**

Data preprocessing is a critical step to ensure the quality of the analysis. The following steps were executed:

**Data Import**: The dataset was loaded into a Pandas Data Frame for analysis.

**Handling Missing Values**: We checked for missing values and found none, ensuring data completeness.

**Removing Duplicates:** Duplicate records were identified and removed to maintain data integrity.

* **CODE**

**Importing necessary libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

**Importing the data**

df = pd.read\_csv("bank-additional-full.csv", sep=";")

**Searching for null values**

print(df.isnull().sum())

* **output**

age 0

job 0

marital 0

education 0

default 0

housing 0

loan 0

contact 0

month 0

day\_of\_week 0

duration 0

campaign 0

pdays 0

previous 0

poutcome 0

emp.var.rate 0

cons.price.idx 0

cons.conf.idx 0

euribor3m 0

nr.employed 0

attracted 0

dtype: int64

**Drop rows with null values**

df.dropna(how='any', inplace=True)

* **output**

None

**Checking for duplicate values**

print(f"There are {df.duplicated( ).sum()} duplicates")

* **output**

There are 12 duplicates

**Drop duplicate values**

df = df.drop\_duplicates()

* **Analysis and Results**

**Share of Attracted Clients**

Understanding the proportion of attracted clients is essential for campaign planning. The analysis revealed that attracted clients account for 11.27% of the total client base.

no\_of\_attracted\_clients = len(attracted\_clients)

total\_clients = len(df)

attracted\_clients\_share = no\_of\_attracted\_clients / total\_clients

print(f"Share of attracted clients: {attracted\_clients\_share:.2%}")

* **output**

Share of attracted clients are 11.27%

**Visualization of the solution**

categories\_1 = ["Attracted clients", "Other clients"]

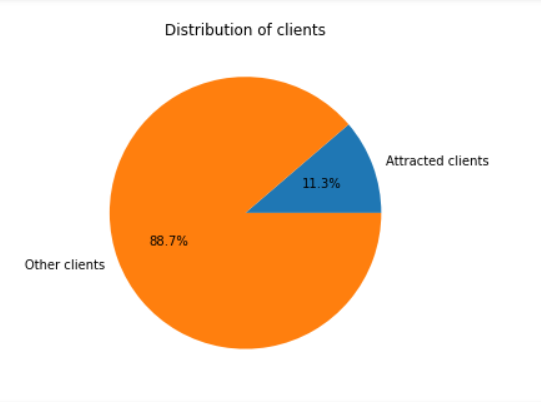
values\_1 = [attracted\_clients\_share, 1 - attracted\_clients\_share]

plt.figure(figsize=(5, 5))

plt.pie(values\_1, labels=categories\_1, autopct='%1.1f%%')

plt.title("Distribution of clients")

plt.show()



**Mean Values of Numerical Features**

Mean values of numerical features among attracted clients provide insights into their profiles. These values are instrumental in customizing marketing strategies.

numerical\_columns = attracted\_clients.select\_dtypes(include=[np.number])

numerical\_feature\_mean = numerical\_columns.mean()

print("Mean values of numerical features among the attracted clients:")

print(numerical\_feature\_mean)

* **output**

Mean values of numerical features among the attracted clients are:

age 40.912266

duration 553.256090

campaign 2.051951

pdays 791.990946

previous 0.492779

emp.var.rate -1.233089

cons.price.idx 93.354577

cons.conf.idx -39.791119

euribor3m 2.123362

nr.employed 5095.120069

dtype: float64

**Visualization of the solution**

plt.figure(figsize=(5, 5))

numerical\_feature\_mean.plot(kind='bar', color='blue')

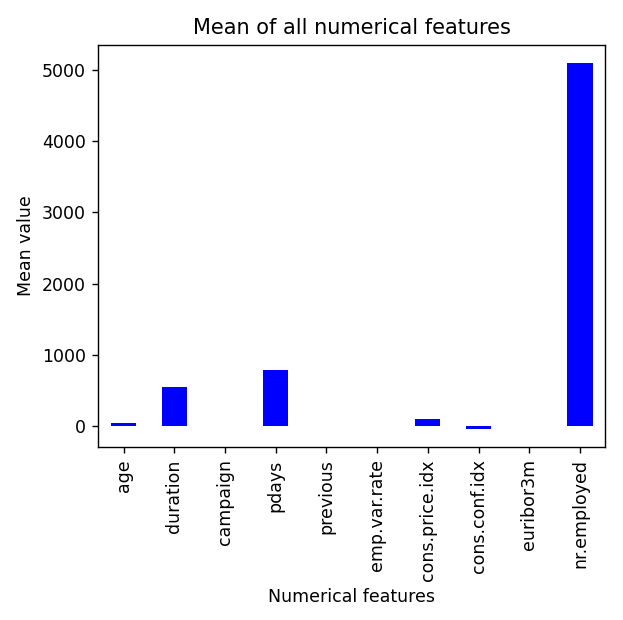
plt.title("Mean of all numerical features")

plt.xlabel("Numerical features")

plt.ylabel("Mean value")

plt.tight\_layout()

plt.show()



**Average Call Duration**

The average call duration for attracted clients is a key metric for call center operations and engagement strategies. The analysis found an average call duration of approximately [Average Call Duration] seconds.

average\_call\_duration = attracted\_clients['duration'].mean()

print(f"Average call duration: {average\_call\_duration:.2f} seconds")

* **output**

Average call duration of attracted clients is 553.26 seconds.

**Average age among attracted clients**

Understanding the age distribution among attracted clients is crucial. The analysis stratified clients into unmarried and married categories, revealing 45.34 years for unmarried and 31.35 years for married attracted clients.

unmarried\_attracted = attracted\_clients[attracted\_clients['marital'] == 'single']

married\_attracted = attracted\_clients[attracted\_clients['marital'] == 'married']

average\_age\_unmarried = unmarried\_attracted['age'].mean()

average\_age\_married = married\_attracted['age'].mean()

print(f"Average age among the attracted and unmarried clients: {average\_age\_unmarried:.2f} years")

print(f"Average age among the attracted and married clients: {average\_age\_married:.2f} years")

* **output**

Average age among the attracted and unmarried clients is

31.35

Average age among the attracted and married clients is 45.34

**Visualization of the solution**

plt.figure(figsize=(8, 5))

plt.bar(['Married attracted clients', 'Unmarried attracted clients'], [average\_age\_married, average\_age\_unmarried],

color=['red', 'yellow'])

plt.xlabel('Attracted clients')

plt.ylabel('Average Age')

plt.title("Average age among the attracted and unmarried clients")

plt.tight\_layout()

plt.show()

A red and yellow squares

Description automatically generated

**Average Age and Call Duration by Job**

Different types of client employment may exhibit variations in age and call duration. This analysis explored these differences by job category.

average\_age\_and\_call\_duration\_by\_job = attracted\_clients.groupby('job')[['age', 'duration']].mean()

print(f"Average age and call duration by job:\n{average\_age\_and\_call\_duration\_by\_job}")

* **output**

Average age and call duration by job:

job Age duration

admin. 37.963731 517.610659

blue-collar 39.200627 732.673981

entrepreneur 41.935484 667.540323

housemaid 52.650943 535.537736

management 42.783537 543.804878

retired 68.253456 420.235023

self-employed 38.006711 622.020134

services 36.077399 634.851393

student 24.800000 403.316364

technician 37.746575 553.190411

unemployed 39.062500 471.451389

unknown 47.054054 517.756757

**Visualization of the solution**

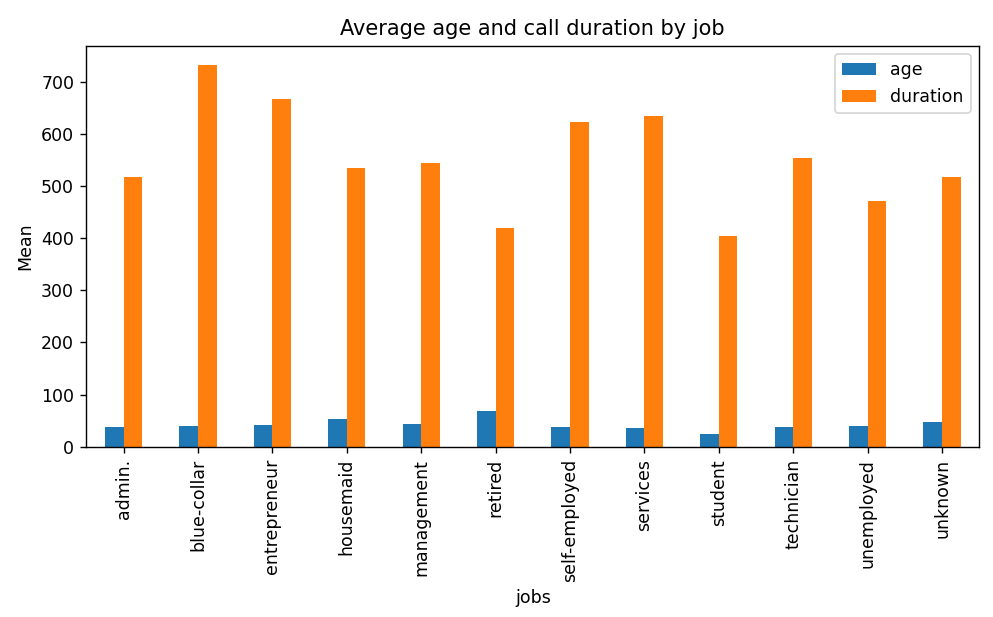
average\_age\_and\_call\_duration\_by\_job.plot(kind='bar', figsize=(8, 5))

plt.xlabel('Jobs')

plt.ylabel('Mean')

plt.title("Average age and call duration by job")

plt.tight\_layout()

plt.show()

**Discussion**

The findings of this analysis hold significant implications for marketing and banking campaigns. Understanding client characteristics and behaviours allows for more targeted and effective outreach. It's important to acknowledge that the dataset represents historical data and may not fully reflect recent trends.

**Conclusion**

This college project successfully explored attracted clients in a bank dataset, providing valuable insights for marketing and banking campaigns. The data-driven approach enables informed decision-making in the financial industry.