Banking Dataset
Marketing Targets Prediction

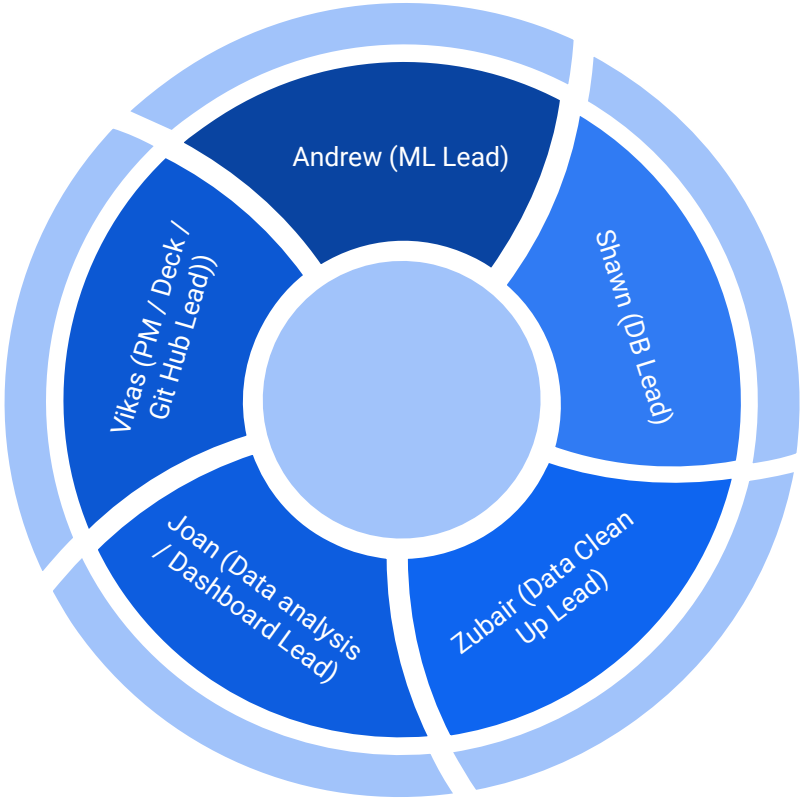
Customer conversion prediction for term deposits

Group 8

Topic

- Every business has a limited marketing budget. Therefore it's vital that each dollar is spent in the most efficient way possible.
- Prediction based on existing customer base really helps in developing focussed marketing initiative and better customer upselling
- Product: Term deposits are similar to GICs, where a client will give the bank money in exchange for the money plus interest after a fixed period of time. During this time, the client is unable to withdraw their money.
- In a bank's case, it's necessary to determine which of their clients will be receptive to phone marketing campaigns regarding the bank's financial services, specifically term deposits.

Team Members



Rationale

- The motivation behind the topic is to determine if marketing campaigns through phone calls is an effective use of marketing spend by companies such as a bank or large institution.
- This could also shed light on why so many people receive fraudulent phone calls of people claiming to be from IRS/CRA demanding money. If phone campaigns are truly effective, then one would expect to continue receiving fraudulent calls.

Data Source

- Dataset source: Kaggle (https://www.kaggle.com/prakharrathi25/banking-dataset-marketing-targets)
- The data was pulled from the UCI Machine learning repository. The data was gathered from marketing campaigns a Portuguese banking institution implemented through phone calls.
- Source: [Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014

Objectives

(Questions to be answered)

Primary Goal:

- To determine whether or not a bank client would be interested in a term deposit subscription based on their profile and past history with the bank. This will enable the bank to better target their phone based marketing efforts towards clients who would be open to a term deposit subscription.

Secondary Goal(s)

- Determine if there is an upper limit on the amount of marketing campaigns a client can receive before terminating communication
- Determine if a relationship between a client's profile and their usage of financial services exists

Communication Protocols

- Group communication will be located on a Slack group that each member will join.
- We also created whatsapp group for quick assembly / communication
- Any updates or changes throughout the project will be posted in this group chat.
- Additionally members will be able to direct message any other member of the group in order to ask them questions or make comments about the project, the data or the work.
- Microsoft Teams is being used for weekly project update meetings

Technology stack

- **Database Storage**
  - Postgres is the database we intend to use since the data for the topic is well structued with a specific schema. We will import and export data using SQL queries and the Python "Pandas" library.
- **Machine Learning**
  - SciKitLearn is the ML library we'll be using to create, train, and test a model. The order of creating a model will be the following
  - Preprocessing data
  - Model selection
  - Model training
  - Model testing and results
  - Output metrics
- **Dashboard**
  - In addition to using a Flask template, we will also integrate D3.js and Plotly for a fully functioning and interactive dashboard. It will be hosted on Github Pages.
  - Tableau

Data exploration

- Extraction:
  - Data extracted from the CSV file using Pandas read me
  - Analysis of features to be used for analysis and machine learning model
- Transformation
  - Removed null values
  - Checked the data types, converted string to boolean whenever needed for code
  - Analysis to check properties about the features such as
    - Distribution
    - missing values
    - extreme values

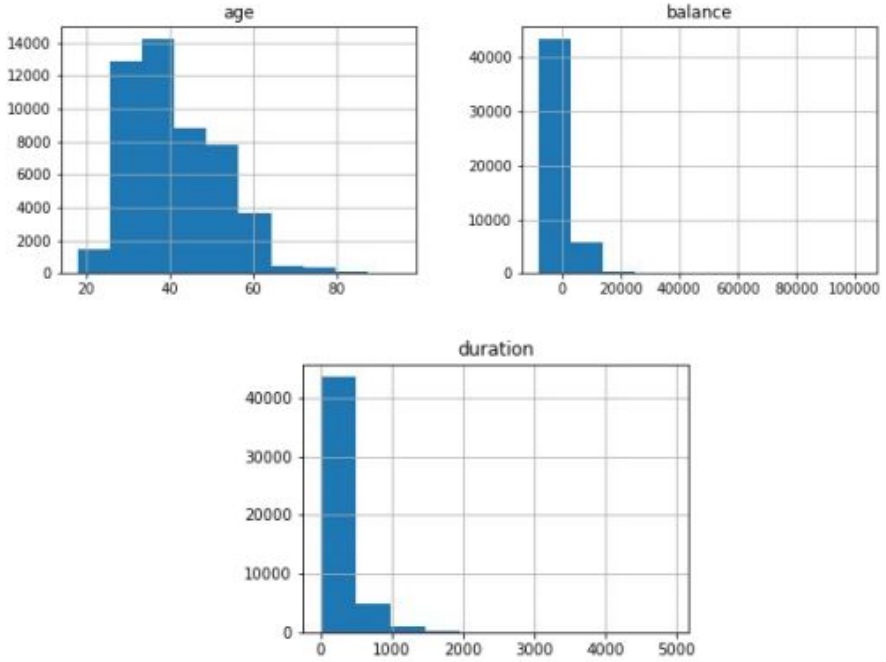
Data exploration

	age	balance	day	duration	campaign	pdays	previous
count	49732.000000	49732.000000	49732.000000	49732.000000	49732.000000	49732.000000	49732.000000
mean	40.957472	1367.761562	15.816315	258.690179	2.766549	40.158630	0.576892
std	10.615008	3041.608766	8.315680	257.743149	3.099075	100.127123	2.254838
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1431.000000	21.000000	320.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

- Balance: Outliers on negative and positive end
- Pdays (past campaign contact): Majority or participants were not part of previous campaign

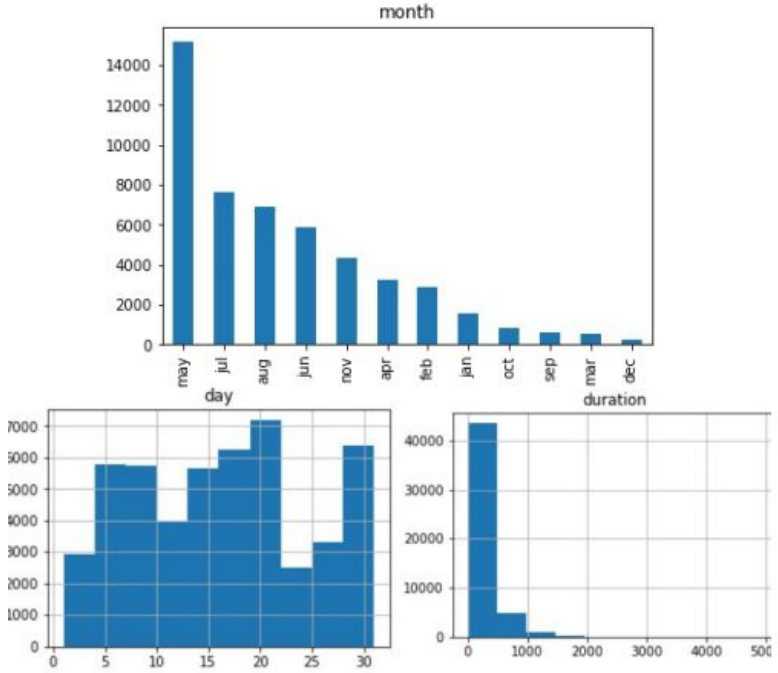
Analysis phase - Feature Distributions

- Age: Age of participant - Most participants are between 20-50 years old
- Balance: Bank account balance - Most people have less than $1500 in their bank account
- Duration: Seconds before end of day when contacted during campaign - Majority of contacts were around the same time of day



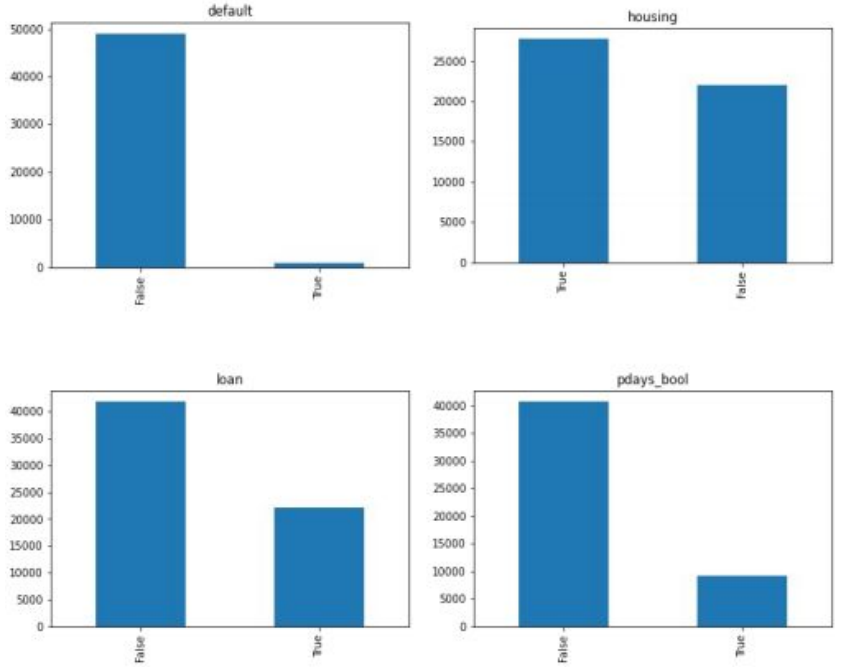
Analysis phase - Feature Distributions

- Month: Month when contacted during campaign - Campaign was held primarily during the summer
- Day: Day of month when contacted during campaign - No specific day was favored
- Duration: Seconds before end of day when contacted during campaign - Majority of contacts were around the same time of day



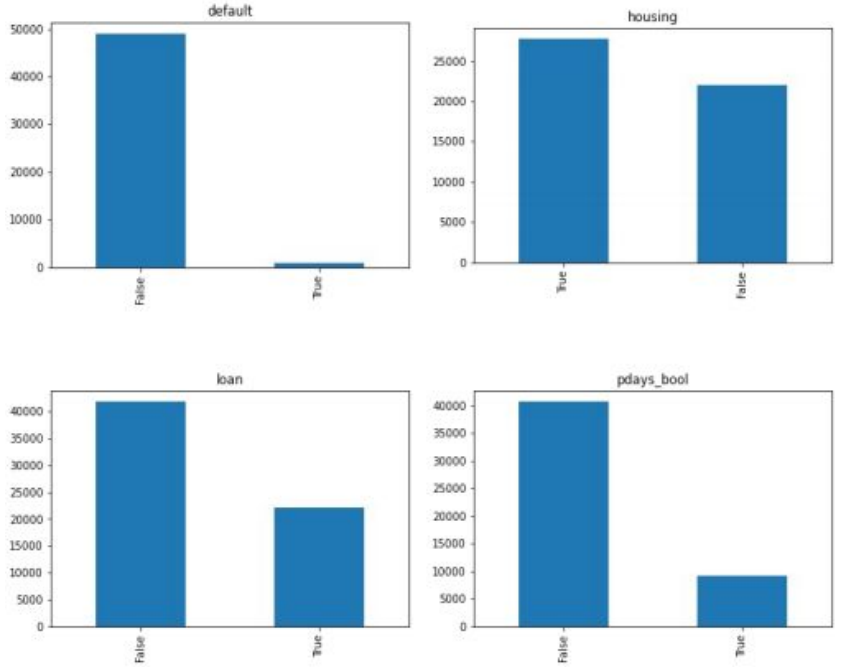
Analysis phase - Feature Distributions

- Default: User has credit in default - Almost no one has defaulted
- Housing: Has a housing loan - Little more than half of the participants have a housing loan
- Loan: Has a personal loan - 20% of participants have a personal loan
  pdays_bool: Whether the participant was contacted in a past campaign - Most people were not contacted in a past campaign



Analysis phase - Feature Distributions

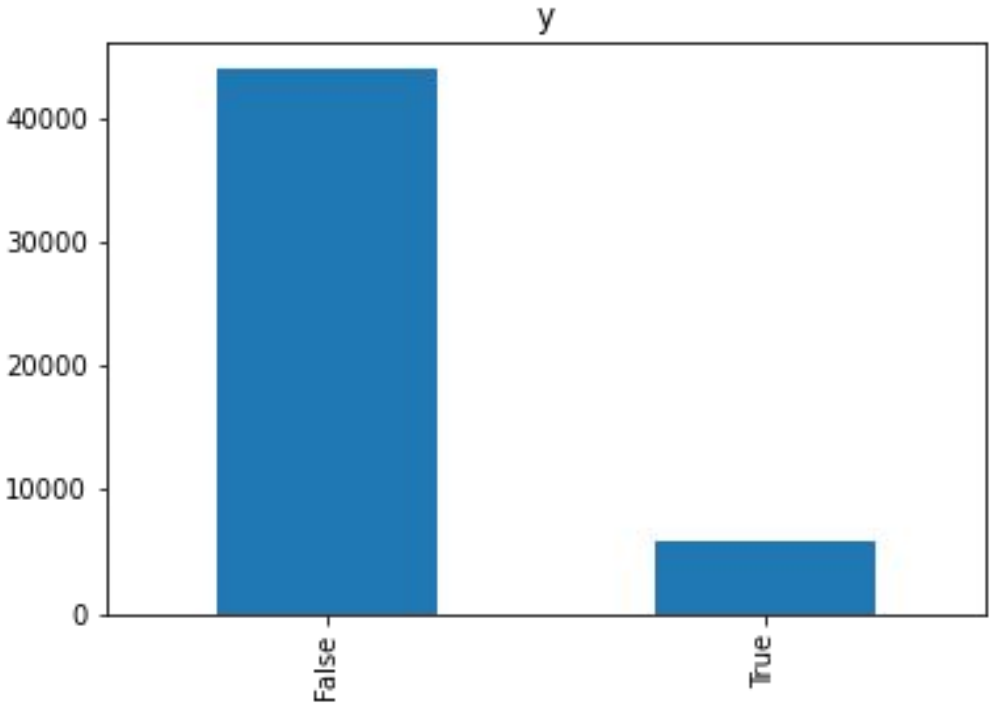
- Default: User has credit in default - Almost no one has defaulted
- Housing: Has a housing loan - Little more than half of the participants have a housing loan
- Loan: Has a personal loan - 20% of participants have a personal loan
  pdays_bool: Whether the participant was contacted in a past campaign - Most people were not contacted in a past campaign



Analysis phase

Frequency of outcome:

- Y: client has subscribed to term deposit
- A class imbalance between True and False is present

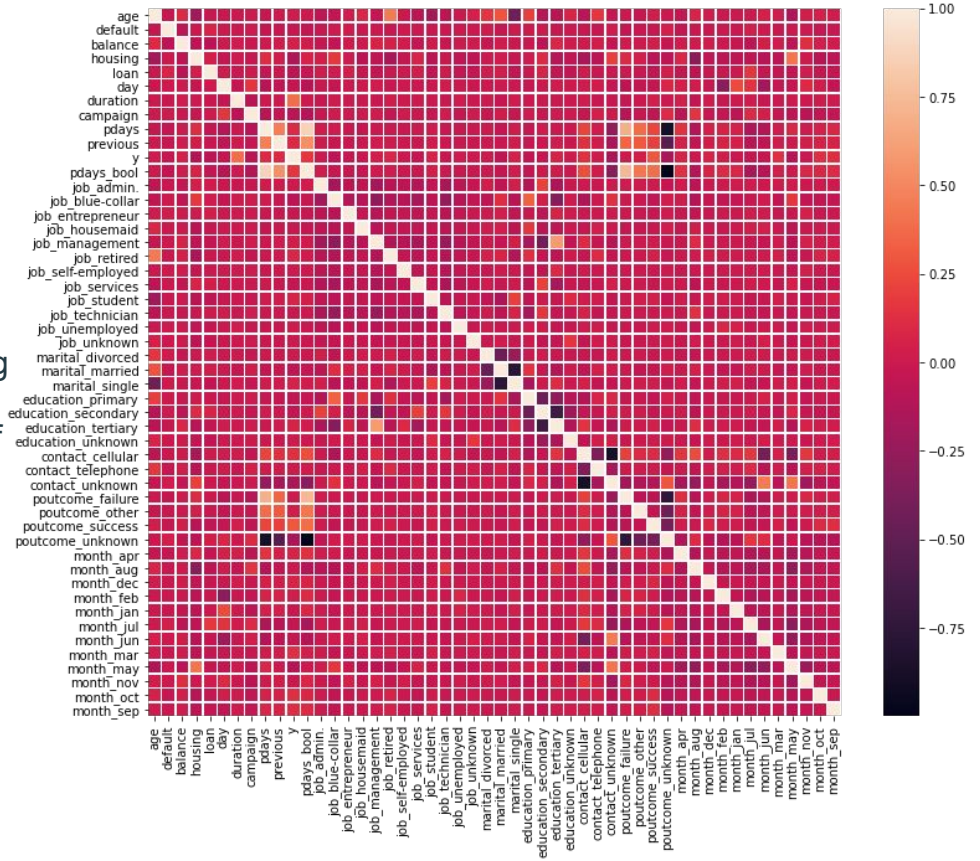


Analysis phase

Find one to one relationships between features and outcome

y: Whether the participant has subscribed to a term deposit

- Y is not correlated strongly with any particular variable. The strongest correlation is with housing age
- Age of participant - Age is correlated with each of the follow features: housing, being married and being a student education
- Education of the participant - Correlated with job



Machine learning

- Preprocessing / Feature engineering:
  - Drop features related to last contact of current campaign (day, month, duration)
    - When conducting a future marketing campaign, these would be chosen by the campaign so shouldn't be a factor in determining who to contact about a term deposit
  - One hot encode categorical features (job, marital, education, contact, poutcome, month)
  - Converted category variables into multiple binary variables for modelling to work
- Train/test set
  - Created data splits into a training and test set using stratification
  - More data for training compared to the default 75/25 split to make up for dataset size - Stratified to handle class imbalance
  - Used sampling on the training set to accommodate the class imbalance of the outcome variable.
- Models
  - Upsample with SMOTE
  - Trained models using the preprocessed training set

Database Integration

- Used Sqlalchemy to connect to database
- Data exported to excel for analysis
- Tables created to hold required data
- Join created to combine the data

```
from sqlalchemy import create_engine
db_password="####################"
engine = create_engine(f"postgresql://postgres:{db_password}@localhost:5432/groupproject")
```

```
# Database: Export to SQL
bank_df.drop(["contact", "day", "month", "duration"], axis=1).to_sql('bank', con=engine, if_exists='replace')
bank_df[["contact", "day", "month", "duration"]].to_sql("contact", con=engine, if_exists='replace')
```

```
# Database: Import to SQL
imported_bank_df = pd.read_sql('bank', con=engine)
imported_contact_df = pd.read_sql('contact', con=engine)
imported_full_df = pd.read_sql("SELECT * FROM bank JOIN contact ON contact.index = bank.index;",
                               con=engine).drop(["index"], axis=1)
print(imported_bank_df.dtypes)
print(imported_contact_df.dtypes)
print(imported_full_df)
```