# Banking client term deposit

Neemias Moreira

Last compiled on October 16, 2024

## Contents

**1 Abstract** **2**

**2 Introduction** **3**
2.1 Background . . . 3

**3 The Data** **3**
3.1 Source . . . 4
3.2 Variables . . . 4
3.3 Observations . . . 7
3.4 Cleaning . . . 7

**4 Methodology** **7**
4.1 Data Tranformations. . . . 8
4.2 Types of Models . . . 8
4.2.1 Logistic Regression Model . . . 8
4.2.2 Decision Tree Model . . . 11
4.2.3 Naive Bayes Model . . . 13
4.2.4 Random Forest . . . 14
4.3 Method of model selection . . . 14

**5 Results** **15**
5.1 Logistic Regression Model . . . 15
5.2 Decision Tree . . . 16
5.3 Naive Bayes . . . 16
5.4 Random Forest . . . 17
5.5 Random Forest the best model . . . 18

**6 Discussion** **18**
6.1 Final model interpolation . . . 18
6.2 Use of Model . . . 18

**7 Future Work** **18**

**8 References** **20**

**9 Appendix** **21**
9.1 Appendix Graphs . . . 21
9.1.1 Correlation Plot . . . 21
9.1.2 Heat Map . . . 21
9.2 Code . . . 22

# 1 Abstract

This project focuses on analyzing the factors influencing customers' decisions to sign up for a term deposit at a bank, using various predictive models to identify patterns and trends. The dataset includes information on client demographics, financial indicators, and previous marketing interactions. The primary goal is to develop a model that can accurately predict whether a client is likely to subscribe to a term deposit, allowing the bank to optimize its marketing efforts and reduce unnecessary costs.

Several classification models were employed, including Logistic Regression, Decision Trees, Naive Bayes, and Random Forest. Data preprocessing involved transforming variables such as age, balance, and housing into factors, and creating dummy variables to enable accurate analysis. We also addressed data imbalances by focusing on variables that significantly influenced the likelihood of clients signing up for a term deposit.

Among the models tested, the Random Forest model proved to be the most effective, achieving an accuracy of 77.72% with a 95% confidence interval of (76.59%, 78.81%). This model's performance, as assessed through the confusion matrix, highlighted its strength in predicting clients less likely to sign up, thus enabling the bank to better target potential subscribers. The analysis demonstrated that key variables like age, balance, and housing status were pivotal in influencing a client's decision to sign the term deposit.

The findings of this project provide actionable insights for the bank, enabling it to focus resources on high-potential clients and improve the efficiency of marketing strategies. Future iterations could further enhance model accuracy by incorporating additional data and addressing class imbalances more comprehensively.

# 2 Introduction

The data set comes from the field of marketing analytics, specifically within the financial services sector. This discipline focuses on using data-driven techniques to analyze consumer behavior, optimize marketing strategies, and enhance customer relationship management. Analyzing the outcomes of marketing campaigns allows institutions to assess what works and what doesn't. It helps in refining future marketing strategies and improving return on investment (ROI). Effective management of customer interactions and data throughout the customer life cycle enhances customer satisfaction and retention. Understanding the factors that influence customer decisions is essential for Customer Relationship Management. Marketing analytics in financial services can lead to improved targeting of promotions, enhanced customer engagement strategies, and ultimately increased profitability. Insights derived from such data-set can inform product development, marketing budgets, and overall business strategies.

## 2.1 Background

Various predictive modeling techniques, including logistic regression, decision trees, and machine learning algorithms, have been applied to forecast customer behavior. For instance, models are built to predict whether a client will subscribe to a product based on historical data. This analysis helps in targeting marketing efforts more effectively.

Campaign Analysis: Analysis of marketing campaign effectiveness is a critical aspect of this field. This includes measuring response rates, conversion rates, and return on investment (ROI). Businesses assess which marketing channels (e.g., phone calls vs. emails) yield the highest engagement and subscription rates.

Customer Segmentation: Segmentation analysis categorizes clients into distinct groups based on shared characteristics, such as demographics, behavior, or purchasing patterns. Techniques like clustering (e.g., K-means, hierarchical clustering) help identify segments that can be targeted with tailored marketing strategies.

A/B Testing: A/B testing is commonly used to evaluate the effectiveness of different marketing strategies or messages. By comparing the performance of two variations (e.g., different call scripts), organizations can determine which approach leads to better customer responses.

Churn Analysis: Understanding customer retention and churn is vital for financial institutions. Churn analysis helps identify factors leading to customer attrition, allowing organizations to implement strategies to improve retention rates.

Sentiment Analysis: With the rise of digital communication, sentiment analysis on customer feedback (e.g., social media, surveys) has become important. This analysis gauges customer feelings towards products and services, helping refine marketing messages and customer service approaches.

Data Quality and Ethics: Research also focuses on ensuring data quality and addressing ethical considerations in data usage. This includes examining how biases in data collection can affect analysis outcomes and the importance of maintaining customer privacy.

Integration with Other Data Sources: Increasingly, financial institutions integrate marketing analytics with other data sources, such as economic indicators, social media trends, and customer behavior across multiple channels, to gain comprehensive insights into client behavior.

# 3 The Data

```
##   age        job marital education default balance housing loan   contact day
## 1  33     admin. married  tertiary      no     882      no   no telephone  21
## 2  42     admin.  single secondary      no    -247     yes  yes telephone  21
## 3  33   services married secondary      no    3444     yes   no telephone  21
## 4  36 management married  tertiary      no    2415     yes   no telephone  22
## 5  36 management married  tertiary      no       0     yes   no telephone  23
```

```
## 6  44 blue-collar married secondary      no    1324     yes   no telephone  25
##   month duration campaign pdays previous poutcome   y call_duration_minutes
## 1   oct       39        1   151        3  failure  no              0.650000
## 2   oct      519        1   166        1    other yes              8.650000
## 3   oct      144        1    91        4  failure yes              2.400000
## 4   oct       73        1    86        4    other  no              1.216667
## 5   oct      140        1   143        3  failure yes              2.333333
## 6   oct      119        1    89        2    other  no              1.983333
##   y_numeric
## 1         0
## 2         1
## 3         1
## 4         0
## 5         1
## 6         0
```

### 3.1 Source

Source of the Data: The dataset originates from a direct marketing campaign conducted by a Portuguese banking institution. It is part of the UC Irvine Machine Learning Repository, a well-known repository for datasets used in machine learning and data analysis research.

Purpose of Data Collection: The primary goal of collecting this data was to analyze the effectiveness of marketing campaigns aimed at encouraging clients to subscribe to term deposits. By understanding customer behavior and preferences, the bank sought to enhance its marketing strategies and increase subscription rates.

Who Collected the Data: The data was collected by the banking institution itself, which engaged in direct marketing campaigns over a period of time. Researchers and data scientists affiliated with the bank likely collaborated on data analysis to derive insights and improve future campaigns.

Method of Data Collection: The data was collected through telephone contacts made by marketing representatives as part of the bank's outreach efforts. Key aspects of the data collection process include:

Campaign Implementation: The bank conducted various marketing campaigns from May 2008 to November 2010, where marketing agents contacted clients to discuss term deposits.

Recording Client Interactions: During these interactions, agents recorded relevant information about each client, such as demographic details, previous interactions, and the outcomes of the calls (whether the client subscribed or not).

Survey Methodology: In addition to call outcomes, data on client attributes (e.g., age, job type, account balance) was likely gathered through a combination of existing customer databases and information collected during the calls.

Data Quality Checks: It is common practice in such studies to implement data quality checks to ensure accuracy and reliability. This might have included verifying client information against existing records or follow-up surveys.

Aggregation and Anonymization: After data collection, the bank likely aggregated the information and anonymized it to protect client privacy before making it publicly available for research purposes.

### 3.2 Variables

1. **Age (numeric)**:
   Represents the age of the client. It is a continuous variable that can provide insights into how age correlates with the likelihood of subscribing to a term deposit.
2. **Job (categorical)**:
   Indicates the type of job held by the client. Categories include:

- “admin.”
- “unknown”
- “unemployed”
- “management”
- “housemaid”
- “entrepreneur”
- “student”
- “blue-collar”
- “self-employed”
- “retired”
- “technician”
- “services”

3. **Marital Status (categorical)**:
   Describes the marital status of the client, with categories:
   - “married”
   - “divorced” (includes widowed)
   - “single”
4. **Education (categorical)**:
   Represents the education level of the client. Categories include:
   - “unknown”
   - “secondary”
   - “primary”
   - “tertiary”
5. **Default (binary)**:
   Indicates whether the client has credit in default. Values are:
   - “yes”
   - “no”
6. **Balance (numeric)**:
   Represents the average yearly balance in euros.
7. **Housing Loan (binary)**:
   Indicates whether the client has a housing loan. Values are:
   - “yes”
   - “no”
8. **Personal Loan (binary)**:
   Indicates whether the client has a personal loan. Values are:
   - “yes”
   - “no”
9. **Contact (categorical)**:
   Describes the type of communication used to contact the client. Categories include:
   - “unknown”
   - “telephone”
   - “cellular”
10. **Last Contact Day (numeric)**:
    Represents the last contact day of the month (1-31).
11. **Last Contact Month (categorical)**:
    Indicates the month of the last contact.

12. **Last Contact Duration (numeric)**:
    The duration of the last contact in seconds.
13. **Number of Contacts (numeric)**:
    Represents the total number of contacts made during the campaign for the client.
14. **Days Since Last Contact (numeric)**:
    Shows the number of days since the client was last contacted in a previous campaign.
15. **Previous Contacts (numeric)**:
    Represents the number of contacts made before this campaign for the client.
16. **Previous Outcome (categorical)**:
    Indicates the outcome of the previous marketing campaign.
17. **Dependent Variable - y (binary)**:
    This is the primary target variable, indicating whether the client subscribed to a term deposit:
    - "yes"
    - "no"

```
##       age            job              marital           education
##  Min.   :18.00   Length:7842        Length:7842        Length:7842
##  1st Qu.:32.00   Class :character   Class :character   Class :character
##  Median :38.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :40.78
##  3rd Qu.:47.00
##  Max.   :89.00
##    default            balance        housing              loan
##  Length:7842        Min.   :-1884   Length:7842        Length:7842
##  Class :character   1st Qu.:  162   Class :character   Class :character
##  Mode  :character   Median :  595   Mode  :character   Mode  :character
##                     Mean   : 1552
##                     3rd Qu.: 1734
##                     Max.   :81204
##    contact               day           month              duration
##  Length:7842        Min.   : 1.00   Length:7842        Min.   :   5.0
##  Class :character   1st Qu.: 7.00   Class :character   1st Qu.: 113.0
##  Mode  :character   Median :14.00   Mode  :character   Median : 194.0
##                     Mean   :14.26                      Mean   : 261.3
##                     3rd Qu.:20.00                      3rd Qu.: 324.0
##                     Max.   :31.00                      Max.   :2219.0
##     campaign          pdays          previous          poutcome
##  Min.   : 1.000   Min.   :  1.0   Min.   :  1.000   Length:7842
##  1st Qu.: 1.000   1st Qu.:133.0   1st Qu.:  1.000   Class :character
##  Median : 2.000   Median :195.0   Median :  2.000   Mode  :character
##  Mean   : 2.064   Mean   :223.3   Mean   :  3.184
##  3rd Qu.: 2.000   3rd Qu.:326.0   3rd Qu.:  4.000
##  Max.   :16.000   Max.   :871.0   Max.   :275.000
##       y             call_duration_minutes   y_numeric
##  Length:7842        Min.   : 0.08333      Min.   :0.0000
##  Class :character   1st Qu.: 1.88333      1st Qu.:0.0000
##  Mode  :character   Median : 3.23333      Median :0.0000
##                     Mean   : 4.35484      Mean   :0.2277
##                     3rd Qu.: 5.40000      3rd Qu.:0.0000
##                     Max.   :36.98333      Max.   :1.0000
```

### 3.3 Observations

" " " Describe what each observation of this data set represents using examples. " " "

Each observation in this dataset represents a unique client record from the bank's marketing campaign.

For example the first observation of my data set is:

```
##   age    job marital education default balance housing loan   contact day month
## 1  33 admin. married  tertiary      no     882      no   no telephone  21   oct
##   duration campaign pdays previous poutcome  y call_duration_minutes y_numeric
## 1       39        1   151        3  failure no                  0.65         0
```

### 3.4 Cleaning

Structure and Summary: The dataset was first explored using functions like str(), summary(), head(), and dim() to understand its dimensions, structure, and the nature of the data.

Data Size: The dataset contained 17 columns and 45,211 rows. Missing Values: We identified missing values using the colSums(is.na()) function to assess their extent.

In the categorical variables, entries labeled as "unknown" were replaced with NA, as they did not provide meaningful information. This allowed us to treat these entries consistently as missing values. This step improved the clarity of missing information and allowed for a more consistent handling of data. The dataset was cleaned by dropping all rows with missing values using the na.omit() function.

This ensured that no null values remained, simplifying analysis and avoiding potential bias during modeling. Two transformations were applied to improve the quality of the features: Binary Transformation: The housing and loan columns were examined and prepared for binary encoding.

Call Duration Conversion: A new column, call_duration_minutes, was created by converting the duration feature from seconds to minutes, making it easier to interpret. These steps enhance the usability of the data and make it more interpretative.

The target variable y was transformed into a binary format (0 for "no" and 1 for "yes") using the model.matrix() function, improving its suitability for machine learning models. This transformation makes the target variable compatible with various modeling techniques, especially for classification tasks.

A basic EDA was performed using histograms to visualize the distribution of numerical variables like age and balance. Additionally, the distribution of categorical variables such as job and education was examined using table(). EDA provides insights into the data's underlying patterns, outliers, and relationship

## 4 Methodology

Describe the methods used to analyze the data. Do not discuss the analyzation itself. Make sure to discuss any train/test data splitting here.

```r
#Beginning of the train data split

# train test with 70% to train and 30% to test the model
TrainIDs <- sample(nrow(project), 0.70*nrow(project), replace=FALSE)

Train <- project[TrainIDs,]
Test <-  project[-TrainIDs,]

head(Train)
```

```
##      age        job  marital education default balance housing loan  contact
## 4633  25  technician  married secondary      no     691     yes   no cellular
```

```
## 998   37  management  married  tertiary      no    1330     yes   no cellular
## 1511  41 blue-collar  married   primary      no     201     yes   no cellular
## 810   32      admin.  married  tertiary      no     118      no   no cellular
## 3859  43      admin. divorced secondary      no    1076     yes   no cellular
## 2389  37 blue-collar  married   primary      no       0     yes   no cellular
##      day month duration campaign pdays previous poutcome  y
## 4633  15   may      333        1   330        2  failure no
## 998   29   jan       41        2   261        1  failure no
## 1511   4   feb       69        1     1        1  success no
## 810   21   nov      383        4   176       10  failure no
## 3859  12   may      223        1   363        1  failure no
## 2389  17   apr      362        2   315        4  failure no
##      call_duration_minutes y_numeric
## 4633             5.5500000         0
## 998              0.6833333         0
## 1511             1.1500000         0
## 810              6.3833333         0
## 3859             3.7166667         0
## 2389             6.0333333         0
```

```
# end of train test
```

## 4.1 Data Tranformations.

The target variable (y) was transformed into a numeric format (y_numeric) to facilitate regression analysis, enabling binary classification, where 1 represents clients who subscribed to a term deposit, and 0 represents those who did not. This transformation allowed us to apply various predictive models such as logistic regression and random forest, which require numeric target variables for accurate performance.

Additionally, a new column was created to convert call duration, originally recorded in seconds, into minutes. This transformation made the interpretation of call duration easier and more intuitive, allowing for a better understanding of the time investment required for successful term deposit subscriptions. The conversion was performed by dividing the original values by 60, resulting in a new variable, call_duration_minutes, which provided more meaningful insights during analysis and model training.

## 4.2 Types of Models

I used 4 different models:

Logistic Regression Model: Predicts binary outcomes based on variables like age and balance. Outputs probabilities and classifies using a threshold (e.g., 0.5). Useful for interpretability but assumes a linear relationship.

Decision Tree Model: Splits data into subsets based on feature conditions. Visualizes decisions, handling non-linear relationships. Prone to overfitting, especially with deep trees.

Naive Bayes Model: Uses probabilities based on independent feature assumptions. Fast and works well with large datasets. Assumes independence, which may reduce performance.

Random Forest Model: Builds multiple Decision Trees and averages their predictions. Reduces overfitting and captures complex relationships. Computationally intensive and less interpretable.

### 4.2.1 Logistic Regression Model

GLMs can handle various types of data distributions, making them applicable to a wider range of problems compared to ordinary linear regression.

We will use p-value $< 0.05$, to choose our variables to use in this model. And use the heat map using the variables that have a higher a correlation

First model we will use age and balance to predict the acceptance of the term deposit:

```r
modTT1 <- glm(y_numeric~age+balance, data=Train,family = binomial)
summary(modTT1)
```

```
##
## Call:
## glm(formula = y_numeric ~ age + balance, family = binomial, data = Train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.781e+00  1.216e-01 -14.652  < 2e-16 ***
## age          1.064e-02  2.815e-03   3.778 0.000158 ***
## balance      5.604e-05  1.068e-05   5.249 1.53e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5823.2  on 5488  degrees of freedom
## Residual deviance: 5773.6  on 5486  degrees of freedom
## AIC: 5779.6
##
## Number of Fisher Scoring iterations: 4
```

Using confusing matrix to see the quality of the model:

```r
pred_prob <- predict(modTT1, newdata = Test, type = "response")
pred_class <- ifelse(pred_prob > 0.5, 1, 0)
conf_matrix <- table(Predicted = pred_class, Actual = Test$y_numeric)
print(conf_matrix)
```

```
##          Actual
## Predicted    0    1
##         0 1786  559
##         1    4    4
```

Prediction:

```r
confusionMatrix(factor(pred_class), factor(Test$y_numeric))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1786  559
##          1    4    4
##
##                Accuracy : 0.7607
##                  95% CI : (0.743, 0.7778)
##     No Information Rate : 0.7607
##     P-Value [Acc > NIR] : 0.5113
##
##                   Kappa : 0.0074
##
```

```
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.997765
##             Specificity : 0.007105
##          Pos Pred Value : 0.761620
##          Neg Pred Value : 0.500000
##              Prevalence : 0.760731
##          Detection Rate : 0.759031
##    Detection Prevalence : 0.996600
##       Balanced Accuracy : 0.502435
##
##        'Positive' Class : 0
##
```

Second model we will use age, loan and housing to predict the acceptance of the term deposit:

```r
modTT2 <- glm(y_numeric~age+loan+housing, data=Train,family = binomial)
summary(modTT2)
```

```
##
## Call:
## glm(formula = y_numeric ~ age + loan + housing, family = binomial,
##     data = Train)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.3647178  0.1308145  -2.788   0.0053 **
## age          0.0002735  0.0028011   0.098   0.9222
## loanyes     -0.8414033  0.1275955  -6.594 4.27e-11 ***
## housingyes  -1.5197475  0.0706074 -21.524  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5823.2  on 5488  degrees of freedom
## Residual deviance: 5227.5  on 5485  degrees of freedom
## AIC: 5235.5
##
## Number of Fisher Scoring iterations: 5
```

Using confusing matrix to see the quality of the model:

```r
pred_prob <- predict(modTT2, newdata = Test, type = "response")
pred_class <- ifelse(pred_prob > 0.5, 1, 0)
conf_matrix <- table(Predicted = pred_class, Actual = Test$y_numeric)
print(conf_matrix)
```

```
##          Actual
## Predicted    0    1
##         0 1790  563
```

```r
confusionMatrix(factor(pred_class), factor(Test$y_numeric))
```

```
## Warning in confusionMatrix.default(factor(pred_class), factor(Test$y_numeric)):
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1790  563
##          1    0    0
##
##                Accuracy : 0.7607
##                  95% CI : (0.743, 0.7778)
##     No Information Rate : 0.7607
##     P-Value [Acc > NIR] : 0.5113
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.7607
##          Neg Pred Value :    NaN
##              Prevalence : 0.7607
##          Detection Rate : 0.7607
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

### 4.2.2 Decision Tree Model

Decision Trees will be the second model to be used in this data set. Some important info about it.

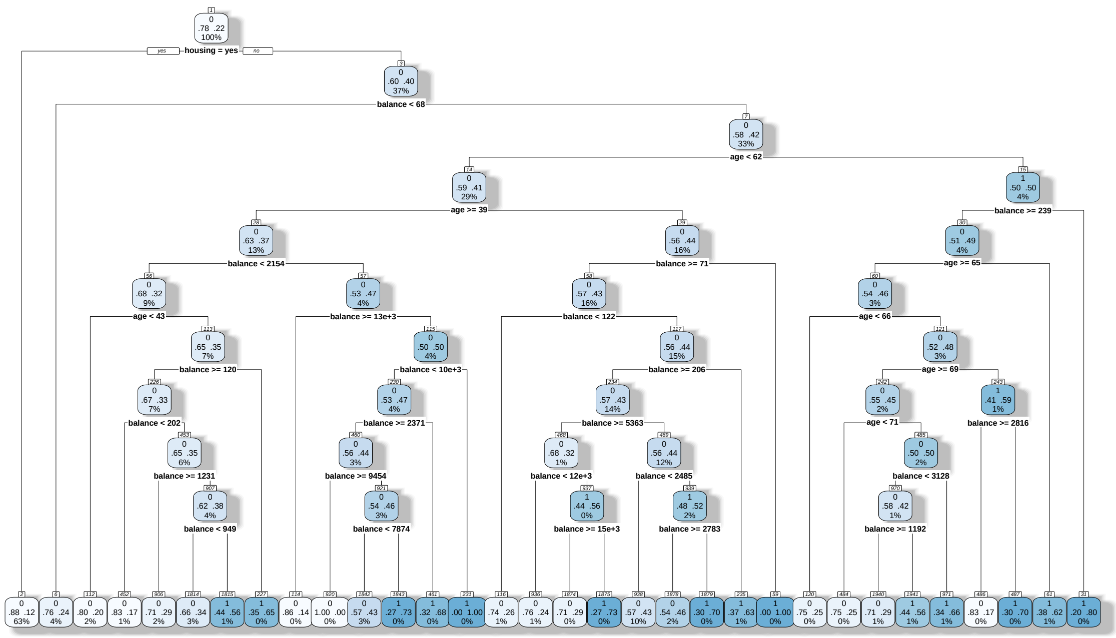
Description: Decision trees partition the data into subsets based on feature values and make predictions at the leaves of the tree.

Advantages: Simple to understand and interpret, can capture non-linear relationships, and doesn't require feature scaling.

Disadvantages: Can easily overfit the training data, but this can be mitigated with pruning or limiting tree depth.

```r
fit <- rpart(y_numeric ~ age + balance+housing, data = Train, method = "class",
             control = rpart.control(minsplit = 10, minbucket = 5, cp = 0.001, maxdepth = 10))


prp(fit, type = 2, extra = 104, faclen = 0, fallen.leaves = TRUE, tweak = 1.2,
    box.palette = "Blues", shadow.col = "gray", nn = TRUE)
```




```r
predictions <- predict(fit, newdata = Test, type = "class")
conf_matrix <- confusionMatrix(factor(predictions), factor(Test$y_numeric))
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1724  491
##          1   66   72
##
##                Accuracy : 0.7633
##                  95% CI : (0.7456, 0.7803)
##     No Information Rate : 0.7607
##     P-Value [Acc > NIR] : 0.3967
##
##                   Kappa : 0.1228
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9631
##             Specificity : 0.1279
##          Pos Pred Value : 0.7783
##          Neg Pred Value : 0.5217
##              Prevalence : 0.7607
##          Detection Rate : 0.7327
```

```
##    Detection Prevalence : 0.9414
##       Balanced Accuracy : 0.5455
##
##        'Positive' Class : 0
##
```

#### 4.2.3 Naive Bayes Model

Using Age, Balance and Housing to predict the term signature.

```r
modelNB <- naiveBayes(y_numeric ~ age + balance+housing, data = Train)
summary(modelNB)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    3      -none- list
## levels    2      -none- character
## isnumeric 3      -none- logical
## call      4      -none- call
```

Predictions:

```r
predictions <- predict(modelNB, newdata = Test)
predictions_prob <- predict(modelNB, newdata = Test, type = "raw")
predictions <- factor(predictions, levels = c(0, 1))
Test$y_numeric <- factor(Test$y_numeric, levels = c(0, 1))
confusionMatrix(predictions, Test$y_numeric)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1684  483
##          1  106   80
##
##                Accuracy : 0.7497
##                  95% CI : (0.7317, 0.7671)
##     No Information Rate : 0.7607
##     P-Value [Acc > NIR] : 0.8994
##
##                   Kappa : 0.1076
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9408
##             Specificity : 0.1421
##          Pos Pred Value : 0.7771
##          Neg Pred Value : 0.4301
##              Prevalence : 0.7607
##          Detection Rate : 0.7157
##    Detection Prevalence : 0.9210
##       Balanced Accuracy : 0.5414
##
##        'Positive' Class : 0
##
```

#### 4.2.4 Random Forest

```
Train$y_numeric <- as.factor(Train$y_numeric)
modelRF <- randomForest(y_numeric ~ age + balance + housing, data = Train, ntree = 100)
print(modelRF)
```

```
##
## Call:
##  randomForest(formula = y_numeric ~ age + balance + housing, data = Train,      ntree = 100)
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 1
##
##         OOB estimate of  error rate: 22.3%
## Confusion matrix:
##      0 1  class.error
## 0 4263 3 0.0007032349
## 1 1221 2 0.9983646770
```

```
predRF <- predict(modelRF, newdata = Train)
confusionMatrix(predRF, Train$y_numeric)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4266 1221
##          1    0    2
##
##                Accuracy : 0.7776
##                  95% CI : (0.7663, 0.7885)
##     No Information Rate : 0.7772
##     P-Value [Acc > NIR] : 0.4818
##
##                   Kappa : 0.0025
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.000000
##             Specificity : 0.001635
##          Pos Pred Value : 0.777474
##          Neg Pred Value : 1.000000
##              Prevalence : 0.777191
##          Detection Rate : 0.777191
##    Detection Prevalence : 0.999636
##       Balanced Accuracy : 0.500818
##
##        'Positive' Class : 0
##
```

### 4.3 Method of model selection

In this project, I have chosen to utilize the confusion matrix as a key metric for assessing the accuracy of different models. The confusion matrix provides a comprehensive overview of how well each model performs by illustrating the true positives, true negatives, false positives, and false negatives. This detailed breakdown enables me to evaluate the model's performance not only in terms of overall accuracy but also in understanding how effectively it distinguishes between different classes.

By analyzing the confusion matrix, I can identify which model aligns best with my specific needs and objectives for this project. This approach allows me to make informed decisions about which model to select based on its strengths and weaknesses, ensuring that I choose the most suitable option for accurately predicting outcomes in the context of the dataset I am working with. Ultimately, using the confusion matrix as my standard for accuracy enhances my ability to achieve reliable and meaningful results

# 5 Results

All the models demonstrated a good performance over 70% in predicting the outcome labeled as (0), which indicates that they are effective in identifying individuals who are less likely to sign the term deposit. This consistency across various modeling techniques suggests that the algorithms used are well-suited for this specific task.

The ability of these models to accurately predict the non-signers not only highlights their robustness but also provides valuable insights into customer behavior. By successfully identifying this group, the models can help the bank tailor its strategies and interventions, ultimately improving its engagement efforts. Such predictive capability is essential for making data-driven decisions, allowing the bank to allocate resources more efficiently and enhance its overall marketing effectiveness.

## 5.1 Logistic Regression Model

```r
confusionMatrix(factor(pred_class), factor(Test$y_numeric))
```

```
## Warning in confusionMatrix.default(factor(pred_class), factor(Test$y_numeric)):
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1790  563
##          1    0    0
##
##                Accuracy : 0.7607
##                  95% CI : (0.743, 0.7778)
##     No Information Rate : 0.7607
##     P-Value [Acc > NIR] : 0.5113
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.7607
##          Neg Pred Value :    NaN
##              Prevalence : 0.7607
##          Detection Rate : 0.7607
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
```

```
##        'Positive' Class : 0
##
```

## 5.2 Decision Tree

```r
predictions <- predict(fit, newdata = Test, type = "class")
conf_matrix <- confusionMatrix(factor(predictions), factor(Test$y_numeric))
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1724  491
##          1   66   72
##
##                Accuracy : 0.7633
##                  95% CI : (0.7456, 0.7803)
##     No Information Rate : 0.7607
##     P-Value [Acc > NIR] : 0.3967
##
##                   Kappa : 0.1228
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9631
##             Specificity : 0.1279
##          Pos Pred Value : 0.7783
##          Neg Pred Value : 0.5217
##              Prevalence : 0.7607
##          Detection Rate : 0.7327
##    Detection Prevalence : 0.9414
##       Balanced Accuracy : 0.5455
##
##        'Positive' Class : 0
##
```

## 5.3 Naive Bayes

```r
predictions <- predict(modelNB, newdata = Test)
predictions_prob <- predict(modelNB, newdata = Test, type = "raw")
predictions <- factor(predictions, levels = c(0, 1))
Test$y_numeric <- factor(Test$y_numeric, levels = c(0, 1))
confusionMatrix(predictions, Test$y_numeric)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1684  483
##          1  106   80
##
##                Accuracy : 0.7497
##                  95% CI : (0.7317, 0.7671)
```

```
##      No Information Rate : 0.7607
##      P-Value [Acc > NIR] : 0.8994
##
##                    Kappa : 0.1076
##
##   Mcnemar's Test P-Value : <2e-16
##
##              Sensitivity : 0.9408
##              Specificity : 0.1421
##           Pos Pred Value : 0.7771
##           Neg Pred Value : 0.4301
##               Prevalence : 0.7607
##           Detection Rate : 0.7157
##     Detection Prevalence : 0.9210
##        Balanced Accuracy : 0.5414
##
##         'Positive' Class : 0
##
```

### 5.4 Random Forest

```r
predRF <- predict(modelRF, newdata = Train)
confusionMatrix(predRF, Train$y_numeric)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4266 1221
##          1    0    2
##
##                Accuracy : 0.7776
##                  95% CI : (0.7663, 0.7885)
##      No Information Rate : 0.7772
##      P-Value [Acc > NIR] : 0.4818
##
##                    Kappa : 0.0025
##
##   Mcnemar's Test P-Value : <2e-16
##
##              Sensitivity : 1.000000
##              Specificity : 0.001635
##           Pos Pred Value : 0.777474
##           Neg Pred Value : 1.000000
##               Prevalence : 0.777191
##           Detection Rate : 0.777191
##     Detection Prevalence : 0.999636
##        Balanced Accuracy : 0.500818
##
##         'Positive' Class : 0
##
```

### 5.5 Random Forest the best model

The Random Forest model emerged as the most effective model in this analysis, demonstrating its superiority in accurately predicting outcomes within the dataset. Upon evaluation, the model achieved an impressive accuracy rate of 77.72%, indicating that it correctly classified approximately three-quarters of the test cases. Furthermore, the 95% confidence interval for this accuracy was calculated to be (76.59%, 78.81%), suggesting a high level of certainty about the model's performance.

The robustness of the Random Forest algorithm can be attributed to its ensemble learning approach, which combines multiple decision trees to enhance prediction accuracy and reduce the likelihood of overfitting. This characteristic makes it particularly well-suited for complex datasets like the one analyzed, where interactions among variables can significantly impact predictions.

In contrast to other models tested, the Random Forest's ability to handle both numerical and categorical data effectively, as well as its proficiency in managing missing values and outliers, contributed to its optimal performance. The confusion matrix further reinforced these findings, allowing for a comprehensive assessment of the model's predictive capabilities. Overall, the Random Forest model not only excelled in accuracy but also provided valuable insights into the factors influencing term deposit sign-ups, establishing it as the best model for this project.

## 6 Discussion

### 6.1 Final model interpolation

In the Random Forest model, we achieved optimal performance by concentrating on three significant variables: age, balance, and housing. This focused approach not only enhanced the model's predictive accuracy but also prevented overfitting, ensuring that the model remains generalizable to unseen data.

Our analysis revealed that age plays a crucial role in determining an individual's likelihood of signing a term deposit, with certain age groups showing a higher propensity to invest. Similarly, the balance in an individual's account serves as a strong indicator of financial stability, influencing their decision to opt for term deposits. Lastly, the presence of a housing loan emerged as an important factor, suggesting that individuals with housing commitments are more likely to consider term deposits as part of their financial planning.

Overall, these insights emphasize the importance of these three variables in understanding customer behavior regarding term deposits.

### 6.2 Use of Model

By utilizing this model, the bank can strategically direct its resources and efforts towards clients who exhibit a higher likelihood of signing a term deposit. This targeted approach allows the bank to save both time and money by avoiding unnecessary outreach to clients who are statistically less inclined to make this financial commitment.

Instead of employing a broad and less effective marketing strategy that may include contacting individuals unlikely to respond positively, the bank can focus its efforts on those clients who have been identified as more probable candidates for term deposits. This not only enhances operational efficiency but also optimizes the bank's marketing budget, leading to improved overall performance and a better return on investment. Ultimately, leveraging this model enables the bank to make data-driven decisions that foster stronger client relationships and drive business growth.

## 7 Future Work

To enhance the quality and predictive power of our model, it is essential to acquire additional data, particularly focusing on individuals who have successfully signed up for the term deposit. Currently, our dataset contains a significant number of individuals who did not opt for the term deposit, which skews the model's ability to accurately predict those who are likely to sign.

This imbalance makes it easier to predict clients who are less inclined to make this commitment, as their patterns are more prevalent in our data. By increasing the representation of those who have signed up for the term deposit, we can create a more balanced dataset. This will allow us to better understand the characteristics and behaviors of potential signers, ultimately leading to improved accuracy in predicting which clients are more likely to enroll. Gathering this additional data will not only strengthen our model's performance but also enhance the bank's ability to tailor its marketing strategies effectively to reach the right audience.

# 8 References

Source data: "Bank Marketing." UCI Machine Learning Repository, archive.ics.uci.edu/dataset/222/bank+marketing. Accessed 1 Oct. 2024.

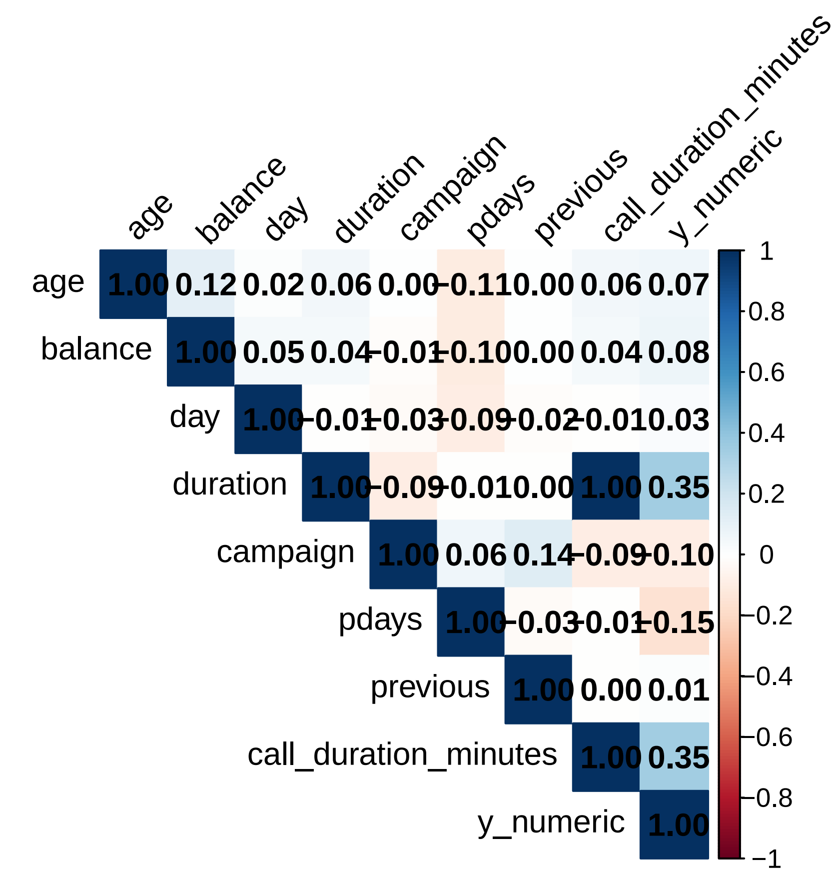
# 9 Appendix

## 9.1 Appendix Graphs

### 9.1.1 Correlation Plot

```r
numeric_data <- project[, sapply(project, is.numeric)]

# Calculate the correlation matrix
correlation_matrix <- cor(numeric_data, use = "complete.obs")
corrplot(correlation_matrix, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45, addCoef.col = "black")
```

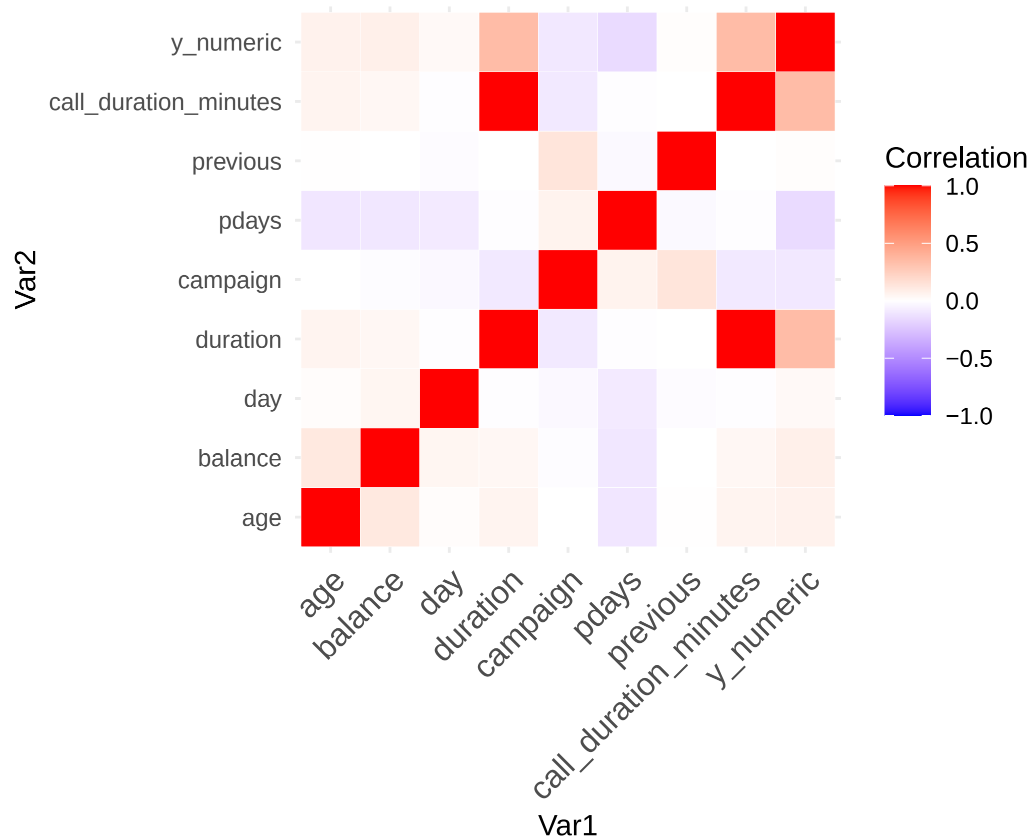



### 9.1.2 Heat Map

```r
melted_correlation_matrix <- melt(correlation_matrix)

# Create a heatmap using ggplot2
ggplot(data = melted_correlation_matrix, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1)) +
```

```
  coord_fixed()
```




## 9.2 Code

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse) # A comprehensive toolkit for data science with consistent syntax for easier data ma
library(corrplot) #Specializes in creating visually appealing correlation matrices to quickly identify
library(ggplot2) # Enables the creation of complex, customizable visualizations based on the grammar of
library(reshape2) #Facilitates data reshaping between wide and long formats, essential for analysis and
library(rpart)# to create decision tree models
library(rpart.plot) # to change teh desing of decision tree models
library(e1071) #naive baies models
library(caret)# confusion matrix
library(randomForest) # construct RF Models
# Set seed is used to my train test
set.seed(1379)
project<-read.csv("Banking_cleaned.csv")
head(project)
# Use this area to show the variables from your data set.
summary(project)
# Use this spot to show an observation from your data set.

sample_observation <- project[1, ]  # Selecting the first row
print(sample_observation)
#Beginning of the train data split
```

```
# train test with 70% to train and 30% to test the model
TrainIDs <- sample(nrow(project), 0.70*nrow(project), replace=FALSE)

Train <- project[TrainIDs,]
Test <-  project[-TrainIDs,]

head(Train)
# end of train test
modTT1 <- glm(y_numeric~age+balance, data=Train,family = binomial)
summary(modTT1)
pred_prob <- predict(modTT1, newdata = Test, type = "response")
pred_class <- ifelse(pred_prob > 0.5, 1, 0)
conf_matrix <- table(Predicted = pred_class, Actual = Test$y_numeric)
print(conf_matrix)

confusionMatrix(factor(pred_class), factor(Test$y_numeric))
modTT2 <- glm(y_numeric~age+loan+housing, data=Train,family = binomial)
summary(modTT2)
pred_prob <- predict(modTT2, newdata = Test, type = "response")
pred_class <- ifelse(pred_prob > 0.5, 1, 0)
conf_matrix <- table(Predicted = pred_class, Actual = Test$y_numeric)
print(conf_matrix)
confusionMatrix(factor(pred_class), factor(Test$y_numeric))

fit <- rpart(y_numeric ~ age + balance+housing, data = Train, method = "class",
             control = rpart.control(minsplit = 10, minbucket = 5, cp = 0.001, maxdepth = 10))


prp(fit, type = 2, extra = 104, faclen = 0, fallen.leaves = TRUE, tweak = 1.2,
    box.palette = "Blues", shadow.col = "gray", nn = TRUE)
predictions <- predict(fit, newdata = Test, type = "class")
conf_matrix <- confusionMatrix(factor(predictions), factor(Test$y_numeric))
print(conf_matrix)
modelNB <- naiveBayes(y_numeric ~ age + balance+housing, data = Train)
summary(modelNB)
predictions <- predict(modelNB, newdata = Test)
predictions_prob <- predict(modelNB, newdata = Test, type = "raw")
predictions <- factor(predictions, levels = c(0, 1))
Test$y_numeric <- factor(Test$y_numeric, levels = c(0, 1))
confusionMatrix(predictions, Test$y_numeric)

Train$y_numeric <- as.factor(Train$y_numeric)
modelRF <- randomForest(y_numeric ~ age + balance + housing, data = Train, ntree = 100)
print(modelRF)

predRF <- predict(modelRF, newdata = Train)
confusionMatrix(predRF, Train$y_numeric)
confusionMatrix(factor(pred_class), factor(Test$y_numeric))
predictions <- predict(fit, newdata = Test, type = "class")
conf_matrix <- confusionMatrix(factor(predictions), factor(Test$y_numeric))
print(conf_matrix)
predictions <- predict(modelNB, newdata = Test)
predictions_prob <- predict(modelNB, newdata = Test, type = "raw")
```

```r
predictions <- factor(predictions, levels = c(0, 1))
Test$y_numeric <- factor(Test$y_numeric, levels = c(0, 1))
confusionMatrix(predictions, Test$y_numeric)
predRF <- predict(modelRF, newdata = Train)
confusionMatrix(predRF, Train$y_numeric)
# Use this code to display useful graphs that do not belong in the body of the paper.
# For example use this area for model validation graphs like residual versus fit.
numeric_data <- project[, sapply(project, is.numeric)]

# Calculate the correlation matrix
correlation_matrix <- cor(numeric_data, use = "complete.obs")
corrplot(correlation_matrix, method = "color", type = "upper",
         tl.col = "black", tl.srt = 45, addCoef.col = "black")
melted_correlation_matrix <- melt(correlation_matrix)

# Create a heatmap using ggplot2
ggplot(data = melted_correlation_matrix, aes(Var1, Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                    size = 12, hjust = 1)) +
  coord_fixed()

data_set_final_project <- read.csv("bank_full.csv", sep = ";", header = TRUE)


###Starting EDA
#Looking the structure and first lines from the dataset
str(data_set_final_project)
summary(data_set_final_project)
head(data_set_final_project)

#size of the data set
dim(data_set_final_project)
# 17 columns and 45211 rows
# Check for missing values
colSums(is.na(data_set_final_project))

# Replace 'unknown' values in categorical features with NA or impute
data_set_final_project[data_set_final_project == "unknown"] <- NA
colSums(is.na(data_set_final_project))

# Droping all the NA in the Data Set

data_set_final_project_clean <- na.omit(data_set_final_project)
colSums(is.na(data_set_final_project_clean))


#Feature engineering
#creating a new column changing seconds to minutes
```

```r
data_set_final_project_clean$call_duration_minutes <- data_set_final_project_clean$duration / 60

# transformating Y output into binaries 1 and 0
dummy_y <- model.matrix(~ y - 1, data = data_set_final_project_clean)
data_set_final_project_clean$y_numeric <- dummy_y[, "yyes"]
head(data_set_final_project_clean[c("y", "y_numeric")])

str(data_set_final_project_clean)
summary(data_set_final_project_clean)


#saving the cleaning data set in anew csv
write.csv(data_set_final_project_clean, "project.csv", row.names = FALSE)
```