Predicting Customer Response in A Portuguese Bank Marketing Campaigns

Daniela Mañozca Cruz (30262558) Luisa Alejandra Sierra Guerra (30261956)
Ruby Nouri Kermani (30261323)

Contents

Introduction

In today's highly competitive marketplace, companies face relentless pressure to stand out and achieve measurable results. As competition intensifies, organizations are investing more than ever in marketing campaigns to promote new products and services. However, it is essential to assess whether these campaigns effectively meet their intended goals.

Within the banking industry, term deposits represent a critical component of the service portfolio, directly impacting financial stability and long-term planning. To promote this product, banks often rely on direct marketing strategies, particularly through telephone-based outreach. The primary objective is to convert contacts into term deposit subscriptions.

The dataset under analysis provides a comprehensive overview of a direct marketing campaign conducted by a Portuguese banking institution. It includes detailed information on client demographics, macroeconomic and social indicators. Beyond individual profiles, the dataset also captures strategic elements of the campaign (such as timing, prior contact history). Crucially, the dataset contains the final outcome variable indicating whether or not the client subscribed to a term deposit.

Using the information from the dataset, the objective of this analysis is to develop a predictive model that estimates the likelihood of subscription. Understanding these drivers is not only a matter of campaign performance evaluation, it is a strategic necessity. By leveraging insights, financial institutions can refine their targeting strategies, enhance resource allocation, and ultimately increase the efficiency and effectiveness of their marketing operations.

Data Sourcing and Justification

The dataset used for this project is licensed under a Creative Commons Attribution 4.0 International (CC By 4.0) license. The dataset can be found on this website: https://www.kaggle.com/datasets/henriqueyamahata/bank-marketing

It is related to tele-marketing campaigns of a Portuguese banking institution. The classification goal is to predict if the client will subscribe to a term deposit (binary target variable). The dataset contains more than 41000 records of 21 variables.

Table 1. *Description of Variables in the Dataset*

Variable_Name	Description	Type
age	Client age in years	Numeric
job	Client main job	Categorical
marital	Client marital status	Categorical
education	Client education level	Categorical
default	Has the client credit in default?	Categorical
housing	Does the client have a housing loan?	Categorical
loan	Does the client have a personal loan?	Categorical
contact	Contact communication type	Categorical
month	Last contact month	Categorical
day_of_week	Last contact day of the week	Categorical
duration	Last contact duration in seconds	Numeric
campaign	Contacts performed during this campaign	Numeric
pdays	Days since last contact in previous campaign	Numeric
previous	Contacts performed before this campaign	Numeric
poutcome	Outcome of the previous campaign	Categorical
emp.var.rate	Employment variation rate	Numeric

cons.price.idx	Consumer price index	Numeric
cons.conf.idx	Consumer confidence index	Numeric
euribor3m	3 month Euro Interbank Offered Rate	Numeric
nr.employed	Number of employees	Numeric
y	Client subscribed to a term deposit?	Categorical (Binary)

Objective and guiding questions

Based on the dataset, the objective of this project is to develop and evaluate predictive models to determine whether a client will subscribe to a term deposit based on various economic and campaign-related attributes. The goal is to identify key factors that influence customer decisions and improve the efficiency of future marketing campaigns.

Guiding questions

1. How do demographic attributes influence customer subscription behavior?

2. What is the relationship between economic indicators (e.g., employment variation rate, number of employees, EURIBOR) and customer subscription response?

3. How do ongoing campaign variables influence customers' willingness to subscribe?

4. How do previous campaign outcomes (**'poutcome'** column) influence the likelihood of subscription to a term deposit?

```
data <- read.csv("bank-full.csv")
```

Data Cleaning

The dataset was initially explored to identify the variable names, examine its dimensions, and assess the proportion of observations based on whether or not the client subscribed to a term deposit.

Table 2. *Overview of the dataset structure, only 5 columns*

age	job	marital	education	default
56	housemaid	married	basic.4y	no
57	services	married	high.school	unknown
37	services	married	high.school	no
40	admin.	married	basic.6y	no
56	services	married	high.school	no
45	services	married	basic.9y	unknown

Table 3. *Dimension of the dataset*

Measure	Value
Number of Rows	41188
Number of Columns	21

Missing Values

An inspection was performed to know the number of missing values per column:

```
missing_table <- data %>%
  summarise(across(everything(), ~ mean(is.na(.)) * 100)) %>%
  pivot_longer(cols = everything(),
               names_to = "Variable",
               values_to = "Missing_Percentage") %>%
  arrange(desc(Missing_Percentage))
```

Table 4. *Missing values in the dataset*

Variable	Missing (%)
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
y	0

The dataset does not contain missing values in the traditional sense; however, it was observed that some variables used the category "unknown" to indicate the absence of information. Therefore, these "unknown" entries are treated as missing values. This issue was causing problems during modeling, especially because some variables were binary (yes/no) while others, such as job categories, included these "unknown" values. Consequently, the decision was made to remove any row containing "unknown" in any variable.

```
# Code to delete rows containing "unknown" in any column
data_clean <- data[!apply(data == "unknown", 1, any), ]
data <- data_clean
```

Outliers

Outliers were examined using the Mahalanobis distance method because it effectively identifies multivariate outliers by considering the combined variability and correlation among predictor variables.

```r
numeric_cols <- sapply(data, is.numeric)
x_data <- data[, numeric_cols]
x_data <- na.omit(x_data)

mahal_dist <- mahalanobis(x_data,
                          center = colMeans(x_data),
                          cov = cov(x_data))

threshold <- qchisq(0.999, df = ncol(x_data))

mahal_table <- tibble(Observation = 1:nrow(x_data),
                      Mahalanobis_Distance = mahal_dist,
                      Outlier = mahal_dist > threshold)

total_obs <- nrow(x_data)
num_outliers <- sum(mahal_table$Outlier)
```

Table 5. *Outliers*

Description	Count
Total observations	30488
Number of Mahalanobis outliers	2188

The 2188 observations identified as outliers represent around 7% of the dataset; therefore, it was decided to remove these observations.

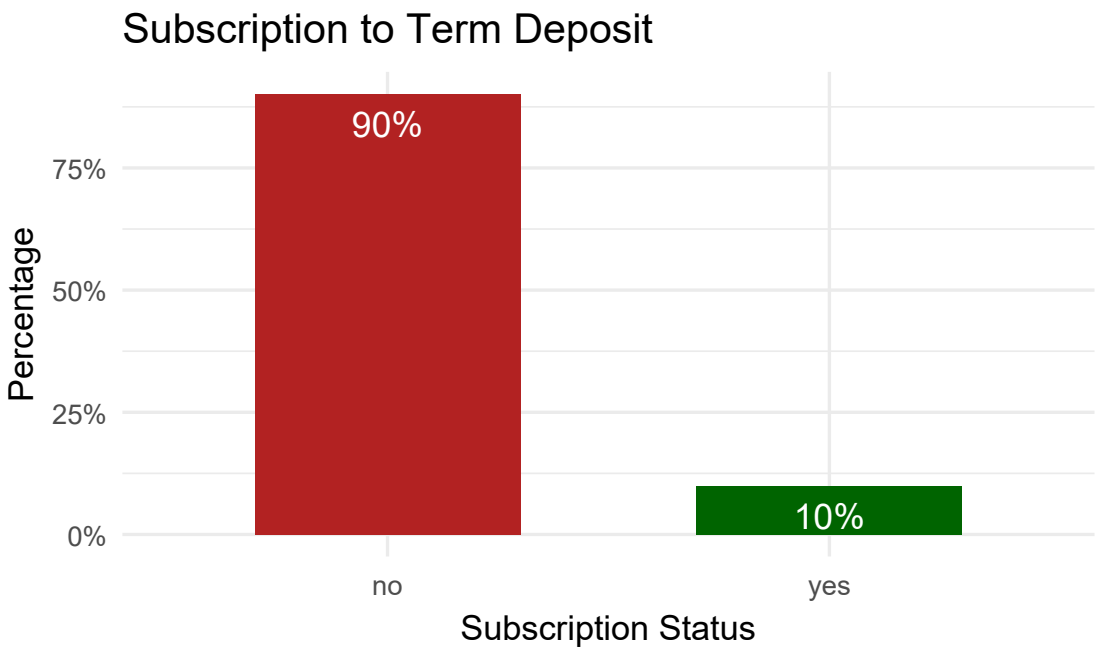
```r
# Code to eliminate Outliers
numeric_data <- data %>%
  filter(complete.cases(dplyr::select(., where(is.numeric))))

x_data <- dplyr::select(numeric_data, where(is.numeric))

x_data <- x_data[, apply(x_data, 2, function(col) var(col) != 0)]

mahal_dist <- mahalanobis(x_data, colMeans(x_data), cov(x_data))
threshold <- qchisq(0.999, df = ncol(x_data))
outliers <- mahal_dist > threshold

clean_data <- numeric_data[!outliers, ]

# Removing pdays
clean_data <- clean_data %>%
  dplyr::select(-pdays)

data <- clean_data
```

When this elimination was performed, one variable became constant, so it was necessary to eliminate this column **(pdays)**.

Exploratory Data Analysis

Marketing Campaign Performance Overview

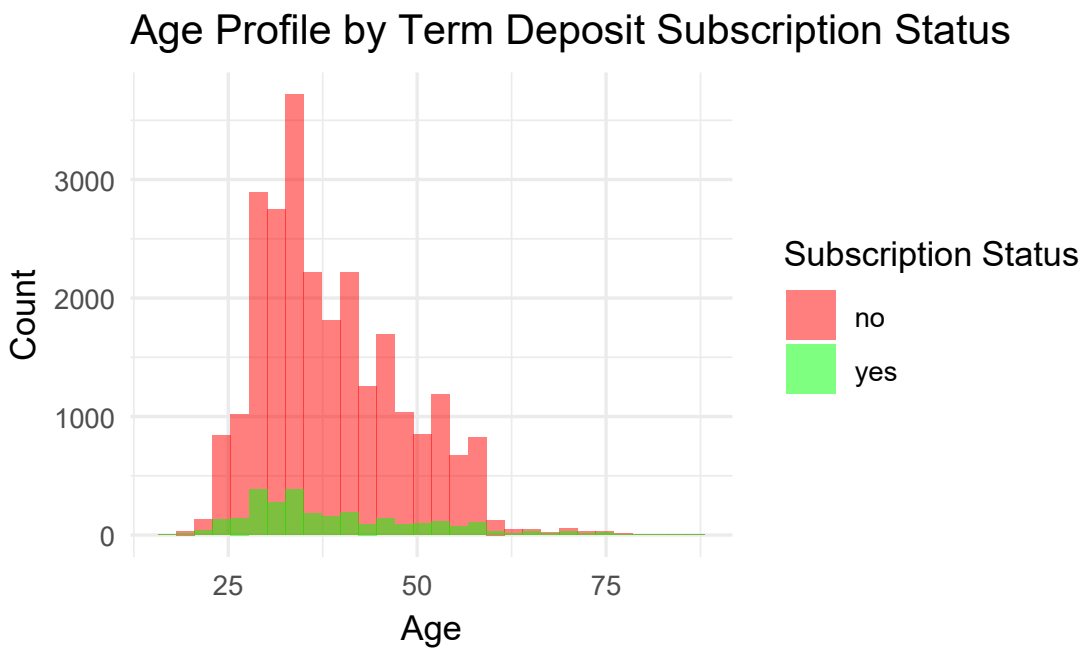
Figure 1. *Distribution of Subscription Outcomes in the Dataset*



The success rate of this campaign was notably low, with 90% of calls resulting in a rejection. This outcome is quite typical in telemarketing campaigns, where the probability of closing a sale through a single cold call is inherently low. This is primarily because cold calling involves reaching out to prospects without prior engagement or interest, which often leads to a higher rate of negative responses.

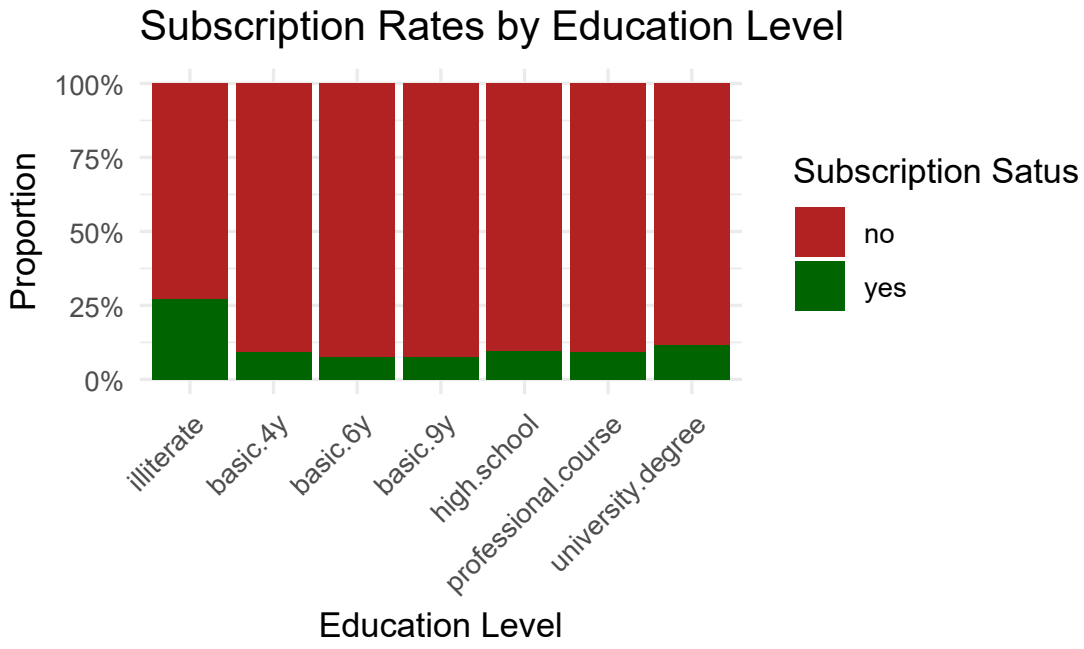
Targeting Optimization (Demographic Attributes)

Figure 2. *Age Profile by Term Deposit Subscription Status*



The chart reveals a consistent level of subscriptions across individuals aged 20 to 50+, with a clear concentration of subscriber activity peaking between ages 28 and 35, suggesting a key demographic window for engagement.

Figure 3. *Subscription Rates by Education Level*

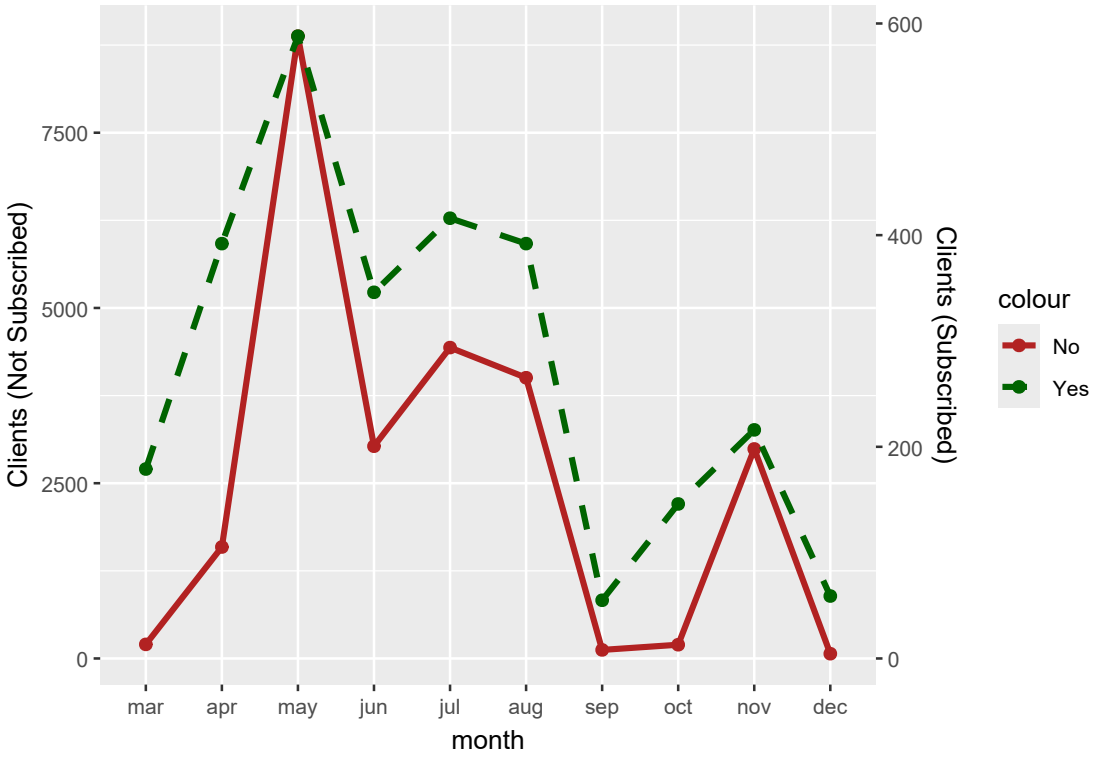


The graph illustrates subscription distribution by education level, revealing that illiterate individuals have the highest acceptance rate at nearly 25%. This may seem surprising but is understandable, as verbal communication likely makes the service more accessible to them compared to other channels they might struggle to navigate. Conversely, those with a university degree follow as the second-highest group, exceeding 20% adoption.

On the other end, the segment with the lowest subscription rate consists of individuals who left school before 9th grade, reaching only 13%. This suggests that while basic education may increase self-sufficiency in accessing services, it doesn't necessarily correlate with higher subscription rates compared to more advanced education levels or purely oral-dependent demographics.

Influence of Campaign Variables

Figure 4. *Temporal Distribution of Campaign Contacts by Subscription Status*



This dual-axis line chart presents the monthly distribution of campaign contacts by subscription outcome. The left y-axis displays the number of clients who did not subscribe, while the right y-axis shows those who did, with the latter re-scaled to allow for direct visual comparison. This approach was selected to account for the over 500% difference between both groups, enabling a more nuanced interpretation of the campaign's success rate across time.

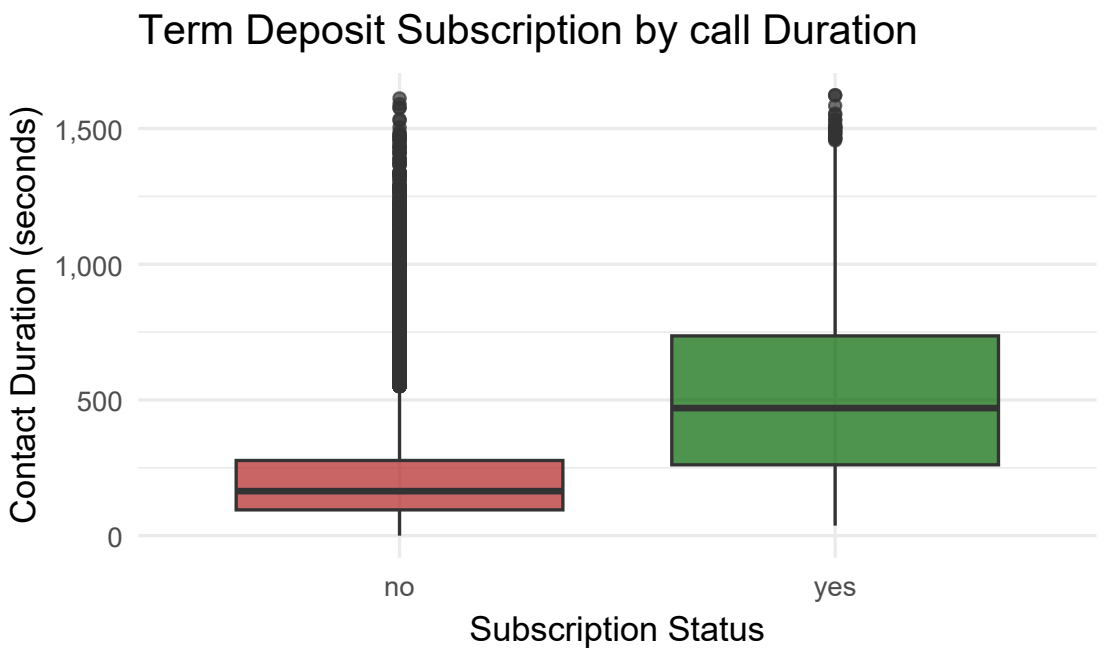
The data reveal that May stands out as the month with the highest campaign activity, surpassing 15000 client contacts. This spike likely reflects the execution of major biannual marketing efforts. Notably, May also recorded the highest absolute number of subscriptions, despite a corresponding peak in rejections.

A secondary peak is observed in July and August, with over 5000 non-subscribers each month. However, these months also achieved the second-highest subscription levels, suggesting a disproportionately high conversion rate relative to contact volume. This pattern indicates potentially more receptive audiences or improved

campaign efficiency during mid-year periods, warranting further investigation into seasonal or contextual factors influencing consumer decisions.

Finally, November marked a minor increase in contact activity, with fewer than 5000 rejections. Although subscription levels during this month were moderate, they contribute to a recurring pattern of campaign intensity tapering off toward the end of the year.

Figure 5. *Effect of Contact Duration on Term Deposit Subscription*



The graph demonstrates how engagement improves with longer call duration, revealing that calls lasting over 500 seconds are critical for securing a **"YES"** to the service subscription. In contrast, rejected calls (**"NO"**) are significantly shorter, averaging under 250 seconds. This trend is understandable given the nature of financial services — customers tend to be more cautious with banking-related offerings due to the perceived risks and need for clarity. As a result, call duration plays a vital role in building engagement, as it allows sales agents to establish trust, address concerns, and effectively communicate the value of the service, ultimately increasing acceptance rates.

The data underscores the importance of investing in longer, quality interactions rather than rushed pitches. A well-structured conversation that educates and reassures the customer leads to higher conversion rates, while shorter calls often fail to overcome initial skepticism. This insight suggests optimizing sales strategies to prioritize meaningful dialogue over call volume efficiency.

Logistic Regression

Variance Inflation Factor

The categorical variables were first coded and converted into factors.

```
# The response variable 'y' was converted into a factor with specified levels
data$y <- factor(data$y, levels = c("no", "yes"))
```

```r
# All categorical variables were converted to factors
categorical_vars <- c("job", "marital", "education", "default", "housing", "loan",
                      "contact", "month", "day_of_week", "poutcome")

data[categorical_vars] <- lapply(data[categorical_vars], as.factor)
```

The Variance Inflation Factor (VIF) was evaluated using a model that excluded the **"loan"** variable, as its inclusion led to a perfect multicollinearity error.

Note:

The following lines are provided as an illustrative example only. When attempting to include all predictor variables simultaneously in the logistic regression model, a perfect multicollinearity error occurs due to redundant or highly correlated predictors. Although the model runs and produces output in the R environment, it fails during the knit process when rendering to PDF format. Therefore, these lines have been commented out to avoid compilation errors:

```r
#LogModelFull <- glm(y~ ., data=data, family = binomial)
#vif(LogModelFull)
```

The results show that the variables **poutcome, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m, and nr.employed** exhibit VIF values exceeding the commonly accepted threshold of 5. Consequently, variable elimination was necessary to mitigate multicollinearity in the model.

The VIF results are presented in the following table. Due to the presence of perfect multicollinearity among several predictors, the values were extracted and compiled manually to ensure compatibility with the document rendering process.

Table 6. *Variance Inflation Factor (VIF)*

Variable	GVIF	Df	GVIF^(1/(2*Df))
age	2.312897	1	1.520821
job	5.832002	11	1.083453
marital	1.465762	3	1.065804
education	3.197933	7	1.086581
default	1.142854	2	1.033946
housing	1.014135	2	1.003515
contact	2.318299	1	1.522596
month	63.049855	9	1.258875
day_of_week	1.066399	4	1.008068
duration	1.243535	1	1.115139
campaign	1.052431	1	1.025881
previous	4.474373	1	2.115271
poutcome	24.228747	2	2.218619
emp.var.rate	142.232401	1	11.926123
cons.price.idx	68.108673	1	8.252798
cons.conf.idx	5.333698	1	2.309480
euribor3m	135.037594	1	11.620568
nr.employed	172.009860	1	13.115253

Based on the results, it was necessary to exclude the variables with a Variance Inflation Factor (VIF) greater than 5. As a result, only the following variables were retained in the model:

Table 7. *Variance Inflation Factor (VIF) Final*

Variable	GVIF	Df	GVIF^(1/(2*Df))
age	2.054999	1	1.433527
job	5.190247	10	1.085824
marital	1.395131	2	1.086810
education	3.072495	6	1.098055
default	1.000001	1	1.000000
housing	1.010704	1	1.005338
contact	1.706823	1	1.306454
month	6.895378	9	1.113234
day_of_week	1.061853	4	1.007530
duration	1.267266	1	1.125729
campaign	1.051676	1	1.025513
previous	1.091259	1	1.044633
emp.var.rate	2.273321	1	1.507754
cons.conf.idx	3.513584	1	1.874456

Likelihood - Ratio Test

To assess variable significance, we first applied the Wald Z-test through a logistic regression model including a comprehensive set of predictors:

```
LogModelFull<- glm(y ~ age + job +  marital + education + default + housing +
                   contact + month + day_of_week + duration + campaign + previous +
                   emp.var.rate + cons.conf.idx, data = data, family = binomial)
summary(LogModelFull)
```

```
##
## Call:
## glm(formula = y ~ age + job + marital + education + default +
##     housing + contact + month + day_of_week + duration + campaign +
##     previous + emp.var.rate + cons.conf.idx, family = binomial,
##     data = data)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -3.381e+00  4.426e-01  -7.638 2.21e-14 ***
## age                          -1.932e-03  3.106e-03  -0.622  0.53388
## jobblue-collar               -2.597e-01  9.938e-02  -2.613  0.00897 **
## jobentrepreneur              -1.541e-01  1.489e-01  -1.035  0.30085
## jobhousemaid                  9.761e-02  1.849e-01   0.528  0.59758
## jobmanagement                -1.108e-01  1.021e-01  -1.086  0.27745
## jobretired                    4.212e-01  1.378e-01   3.057  0.00223 **
## jobself-employed             -1.011e-01  1.352e-01  -0.747  0.45485
## jobservices                  -2.419e-01  1.046e-01  -2.314  0.02069 *
## jobstudent                    3.424e-01  1.437e-01   2.384  0.01714 *
## jobtechnician                -4.796e-02  8.424e-02  -0.569  0.56916
## jobunemployed                 1.507e-01  1.527e-01   0.987  0.32374
## maritalmarried                5.413e-03  8.407e-02   0.064  0.94866
## maritalsingle                 9.691e-02  9.363e-02   1.035  0.30063
## educationbasic.6y             1.427e-02  1.610e-01   0.089  0.92937
## educationbasic.9y             2.305e-02  1.231e-01   0.187  0.85148
## educationhigh.school          8.212e-02  1.199e-01   0.685  0.49323
```

```
## educationilliterate           1.487e+00  9.097e-01   1.634  0.10226
## educationprofessional.course  1.501e-01  1.305e-01   1.150  0.25017
## educationuniversity.degree    2.465e-01  1.205e-01   2.046  0.04078 *
## defaultyes                   -7.302e+00  1.135e+02  -0.064  0.94869
## housingyes                   -1.556e-02  4.984e-02  -0.312  0.75488
## contacttelephone             -1.465e-01  8.196e-02  -1.788  0.07377 .
## monthaug                      9.673e-02  1.399e-01   0.691  0.48933
## monthdec                      2.880e-01  2.353e-01   1.224  0.22094
## monthjul                      3.268e-01  1.158e-01   2.821  0.00478 **
## monthjun                      3.209e-01  1.060e-01   3.027  0.00247 **
## monthmar                      1.804e+00  1.320e-01  13.673  < 2e-16 ***
## monthmay                     -8.221e-01  8.838e-02  -9.302  < 2e-16 ***
## monthnov                     -7.181e-01  1.252e-01  -5.735 9.76e-09 ***
## monthoct                      2.960e-01  1.964e-01   1.507  0.13173
## monthsep                     -5.919e-01  2.294e-01  -2.581  0.00986 **
## day_of_weekmon               -1.804e-03  8.078e-02  -0.022  0.98218
## day_of_weekthu                8.074e-02  7.905e-02   1.021  0.30705
## day_of_weektue                1.525e-01  8.069e-02   1.890  0.05881 .
## day_of_weekwed                2.223e-01  8.042e-02   2.764  0.00572 **
## duration                      5.308e-03  9.636e-05  55.085  < 2e-16 ***
## campaign                     -3.844e-02  1.548e-02  -2.483  0.01304 *
## previous                      2.125e-02  6.046e-02   0.352  0.72521
## emp.var.rate                 -6.642e-01  2.166e-02 -30.665  < 2e-16 ***
## cons.conf.idx                 2.218e-02  7.738e-03   2.867  0.00414 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 18219  on 28299  degrees of freedom
## Residual deviance: 11531  on 28259  degrees of freedom
## AIC: 11613
##
## Number of Fisher Scoring iterations: 10
```

Table 8. *Summary of Variable Significance Based on Wald z-test*

Variable	Significant
age	No
job	Yes
marital	No
education	Yes
default	No
housing	No
contact	No
month	Yes
day_of_week	Yes
duration	Yes
campaign	Yes
previous	No
emp.var.rate	Yes
cons.conf.idx	Yes

Since several variables were found to be non-significant, we employed a stepwise selection procedure using both forward and backward directions to retain only significant predictors:

```
##
## Call:
## glm(formula = y ~ job + education + contact + month + day_of_week +
##     duration + campaign + emp.var.rate + cons.conf.idx, family = binomial,
##     data = data)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -3.477e+00  4.032e-01  -8.623  < 2e-16 ***
## jobblue-collar               -2.590e-01  9.915e-02  -2.612  0.00900 **
## jobentrepreneur              -1.787e-01  1.483e-01  -1.205  0.22815
## jobhousemaid                  7.496e-02  1.839e-01   0.408  0.68357
## jobmanagement                -1.389e-01  1.008e-01  -1.377  0.16838
## jobretired                    3.494e-01  1.197e-01   2.920  0.00350 **
## jobself-employed             -1.071e-01  1.352e-01  -0.792  0.42824
## jobservices                  -2.400e-01  1.044e-01  -2.299  0.02153 *
## jobstudent                    4.198e-01  1.374e-01   3.055  0.00225 **
## jobtechnician                -4.397e-02  8.421e-02  -0.522  0.60158
## jobunemployed                 1.450e-01  1.527e-01   0.950  0.34217
## educationbasic.6y             1.702e-02  1.605e-01   0.106  0.91557
## educationbasic.9y             3.508e-02  1.224e-01   0.287  0.77447
## educationhigh.school          1.077e-01  1.185e-01   0.908  0.36370
## educationilliterate           1.502e+00  9.036e-01   1.662  0.09643 .
## educationprofessional.course  1.671e-01  1.297e-01   1.288  0.19770
## educationuniversity.degree    2.804e-01  1.187e-01   2.362  0.01816 *
## contacttelephone             -1.492e-01  8.181e-02  -1.823  0.06829 .
## monthaug                      9.947e-02  1.398e-01   0.711  0.47690
## monthdec                      2.798e-01  2.347e-01   1.192  0.23323
## monthjul                      3.402e-01  1.156e-01   2.944  0.00324 **
## monthjun                      3.311e-01  1.056e-01   3.134  0.00172 **
## monthmar                      1.817e+00  1.317e-01  13.799  < 2e-16 ***
## monthmay                     -8.137e-01  8.815e-02  -9.230  < 2e-16 ***
## monthnov                     -7.192e-01  1.252e-01  -5.745 9.19e-09 ***
## monthoct                      3.051e-01  1.960e-01   1.556  0.11961
## monthsep                     -5.980e-01  2.291e-01  -2.610  0.00906 **
## day_of_weekmon               -4.596e-03  8.072e-02  -0.057  0.95460
## day_of_weekthu                8.058e-02  7.900e-02   1.020  0.30775
## day_of_weektue                1.518e-01  8.064e-02   1.883  0.05971 .
## day_of_weekwed                2.234e-01  8.036e-02   2.780  0.00544 **
## duration                      5.306e-03  9.627e-05  55.110  < 2e-16 ***
## campaign                     -3.866e-02  1.548e-02  -2.498  0.01248 *
## emp.var.rate                 -6.670e-01  2.134e-02 -31.248  < 2e-16 ***
## cons.conf.idx                 2.151e-02  7.717e-03   2.788  0.00531 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 18219  on 28299  degrees of freedom
## Residual deviance: 11535  on 28265  degrees of freedom
## AIC: 11605
```

```
##
## Number of Fisher Scoring iterations: 6
```

As a result, five variables were excluded from the model: age, marital, default, housing, and contact. We now have two models:

- Full Model: includes 14 predictors
- Reduced Model: includes only 9 predictors

To evaluate whether the reduced model provides a statistically comparable fit to the full model, we conducted a Likelihood Ratio Test. This test compares the log-likelihoods of the two models. While removing predictors usually reduces model fit, the test allows us to determine if the decrease is statistically significant.

Hypotheses:

$$H_0 : \beta_{r+1} = \beta_{r+2} ... = \beta_p = 0 (\text{ reduced model is true})$$
$$H_1 : \text{at least one } \beta_i \neq 0 (\text{ larger model is true})$$

The likelihood ratio statistic is

$$\triangle G^2 = -2logL_{reduced} - (-2logL_{larger})$$

The p-value is $p(\chi^2 > \triangle G^2)$

```
anova(LogModelFull, reduced_model, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ age + job + marital + education + default + housing + contact +
##     month + day_of_week + duration + campaign + +previous + emp.var.rate +
##     cons.conf.idx
## Model 2: y ~ job + education + contact + month + day_of_week + duration +
##     campaign + emp.var.rate + cons.conf.idx
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1     28259      11531
## 2     28265      11535 -6  -4.3385    0.631
```

The results indicate that the p-value exceeds 0.05, suggesting there is insufficient evidence to reject the null hypothesis. Therefore, the reduced model should be preferred, as it offers a more parsimonious representation without a significant loss in explanatory power. Additionally, by excluding non-contributing predictors, the reduced model minimizes complexity and mitigates the risk of over-fitting.

A final model comparison will be performed using a validation approach to confirm these findings and assess predictive performance.

Validation Approach

As observed in the exploratory data analysis (EDA), the subscription success rate (i.e., instances where the individual subscribed) was only 11%. Given the limited number of "yes" responses, a manual stratified subsetting approach was implemented. Specifically, 80% of the "yes" responses and 80% of the "no" responses were assigned to the test dataset, while the remaining 20% of each group were allocated to the training dataset. This ensured that both datasets preserved the original class distribution.

```
#Code to create train and test dataset
set.seed(2024)
data$id <- 1:nrow(data)

# Separate dataset in "yes" or "no"
yes_data <- data[data$y == "yes", ]
no_data  <- data[data$y == "no",  ]

# Select the same proportion on each category
train_yes_idx <- sample(1:nrow(yes_data), 0.8 * nrow(yes_data))
train_no_idx  <- sample(1:nrow(no_data),  0.8 * nrow(no_data))

train_data <- rbind(yes_data[train_yes_idx, ], no_data[train_no_idx, ])
test_data  <- anti_join(data, train_data, by = "id")

train_data <- train_data %>%
  dplyr::select(-id)

test_data <- test_data %>%
  dplyr::select(-id)
```

Table 9. *Records in Train and Test Datasets*

Dataset	Records
Train	22639
Test	5661

Final Model GLM

To compare the predictive performance of the full and reduced models, both were trained on the training dataset and evaluated on the test dataset using a classification threshold of 0.5.

Full Model

```
LogModel <- glm(y ~ age + job +  marital + education + default + housing +
                  contact + month + day_of_week + duration + campaign + previous +
                  emp.var.rate + cons.conf.idx, data = train_data, family = binomial)


Prob.predict <- predict(LogModel, test_data, type="response")
Predict <- rep("no", dim(test_data)[1])
Predict[Prob.predict>=0.5]="yes"
Actual <- test_data$y
```

Table 10. *Confusion Matrix for Logistic Regression - Full Model*

	no	yes
no	4969	399
yes	134	159

```
conf_matrix <- table(Predict, Actual)
misclassification_rate <- sum(Predict != Actual) / length(Actual)
print(paste("Misclassification Rate:", round(misclassification_rate, 4)))
```

```
## [1] "Misclassification Rate: 0.0942"
```

The full model correctly identified 4969 true negatives and 159 true positives were correctly predicted. However, there were 134 false negatives and 399 false positives. The overall misclassification rate is 9.42%, indicating that the model accurately classifies approximately 90.58% of the cases.

Reduced Model:

```
Prob.predict <- predict(reduced_model, test_data, type="response")
Predict <- rep("no", dim(test_data)[1])
Predict[Prob.predict>=0.5]="yes"
Actual <- test_data$y

conf_matrix <- table(Predict, Actual)
misclassification_rate <- sum(Predict != Actual) / length(Actual)
```

Table 11. *Confusion Matrix for Logistic Regression - Reduced Model*

	no	yes
no	4972	397
yes	131	161

```
## [1] "Misclassification Rate: 0.0933"
```

The reduced model correctly predicted 4972 true negatives and 161 true positives were correctly predicted. However, there were 131 false negatives and 397 false positives. The overall misclassification rate is 9.33%, indicating that the model accurately classifies approximately 90.67% of the cases.

Although the performance improvement in terms of misclassification is marginal, the reduced model is preferred due to its:

- Lower complexity, using fewer predictors
- Improved interpretability
- Reduced risk of over-fitting

Therefore, the reduced model is selected as the final model for this logistic regression analysis.

```
reduced_model <- glm(y~ job + education + contact +
                       month + day_of_week + duration+
                       campaign + emp.var.rate+ cons.conf.idx,
                     data = data, family = binomial)
#summary(reduced_model)
```

- (Dispersion parameter for binomial family taken to be 1)
- Null deviance: 18219 on 28299 degrees of freedom
- Residual deviance: 11535 on 28265 degrees of freedom
- AIC: 11605
- Number of Fisher Scoring iterations: 6

Table 12. *Summary for Logistic Regression Final Model*

Variable	Estimate	Std_Error	z_value	p_value
(Intercept)	-1.6600	0.3993	-4.158	0.0000
jobblue-collar	-0.2590	0.0992	-2.612	0.0090
jobentrepreneur	-0.1787	0.1483	-1.205	0.2282
jobhousemaid	0.0750	0.1839	0.408	0.6836
jobmanagement	-0.1389	0.1008	-1.377	0.1684
jobretired	0.3494	0.1197	2.920	0.0035
jobself-employed	-0.1071	0.1352	-0.792	0.4282
jobservices	-0.2400	0.1044	-2.299	0.0215
jobstudent	0.4198	0.1374	3.055	0.0022
jobtechnician	-0.0440	0.0842	-0.522	0.6016
jobunemployed	0.1450	0.1527	0.950	0.3422
educationbasic.6y	0.0170	0.1605	0.106	0.9156
educationbasic.9y	0.0351	0.1224	0.287	0.7745
educationhigh.school	0.1077	0.1185	0.908	0.3637
educationilliterate	1.5020	0.9036	1.662	0.0964
educationprofessional.course	0.1671	0.1297	1.288	0.1977
educationuniversity.degree	0.2804	0.1187	2.362	0.0182
contacttelephone	-0.1492	0.0818	-1.823	0.0683
monthapr	-1.8170	0.1317	-13.799	0.0000
monthmay	-2.6310	0.1252	-21.021	0.0000
monthjun	-1.4860	0.1345	-11.049	0.0000
monthjul	-1.4770	0.1416	-10.428	0.0000
monthaug	-1.7180	0.1564	-10.981	0.0000
monthsep	-2.4150	0.2394	-10.087	0.0000
monthoct	-1.5120	0.2065	-7.323	0.0000
monthnov	-2.5360	0.1495	-16.962	0.0000
monthdec	-1.5370	0.2459	-6.252	0.0000
day_of_weekmon	-0.0046	0.0807	-0.057	0.9546
day_of_weekthu	0.0806	0.0790	1.020	0.3078
day_of_weektue	0.1518	0.0806	1.883	0.0597
day_of_weekwed	0.2234	0.0804	2.780	0.0054
duration	0.0053	0.0001	55.110	0.0000
campaign	-0.0387	0.0155	-2.498	0.0125
emp.var.rate	-0.6670	0.0213	-31.248	0.0000

Variable	Estimate	Std_Error	z_value	p_value
cons.conf.idx	0.0215	0.0077	2.788	0.0053

Linear Discriminant Analysis (LDA)

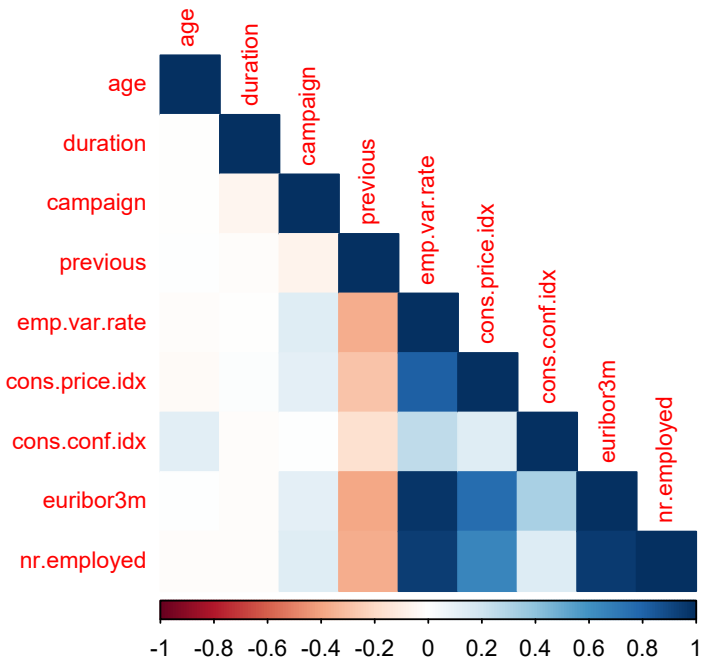
LDA Assumptions Validation

The criteria for LDA is the following:

High correlation validation

Highly correlated predictors can cause problems because LDA relies on the inverse of the covariance matrix. (Correlation matrix can be checked)

Figure 6. *Correlation Matrix*



The variables that show very strong correlations are:

- **euribor3m** and **nr.employed**
- **emp.var.rate** and **euribor3m**
- **emp.var.rate** and **nr.employed**

The variable **id** also exhibit a high correlation with multiple variables, but since is an identifier, it will not be consider in the modeling. Based on the results the variables that will be excluded is **"nr.employed"** and **"euribor3m"**.

Normal distribution within classes

LDA assumes the explanatory variables are normally distributed within each class of the response variable.We are going to use Mardia's Test, which determines whether or not a group of variables follows a multivariate normal distribution. The null and alternative hypotheses for the test are as follows:

H0 (null hypothesis): The variables follow a multivariate normal distribution.

Ha (alternative hypothesis): The variables do not follow a multivariate normal distribution.

```
#The dependent variable has 2 classes: Yes and No.
#The explanatory variables must be numerical
# The variable "nr.employed" and "euribor3m" are excluded

predictors<-c("age", "duration","campaign", "previous", "emp.var.rate" ,
              "cons.price.idx", "cons.conf.idx")
yes_data_numerical<-subset(yes_data, select = predictors)
no_data_numerical<-subset(no_data, select = predictors)
```

```
# Mardia's multivariate normality test for Yes class
mult.norm(yes_data_numerical)$mult.test
```

```
##          Beta-hat      kappa p-val
## Skewness 19.15357 8903.22011     0
## Kurtosis 71.41659   19.79908     0
```

Given p-value for Skewness and Kurtosis are extremely small, less than 0.05, we reject the null hypothesis for both test, meaning the Multivariate normality assumption is not met for the **"Yes"** class. LDA assumes that the predictor variables are multivariate normally distributed within each class, since this assumption is violated the LDA boundary for classifying could be less reliable.

For performance issues, the Normality test for the **No Class** will be run only over the 10% of the data.

```
set.seed(10)
sample_size <- 0.1*nrow(no_data_numerical)
idx <- sample(1:nrow(no_data_numerical), size = sample_size)
no_data_numerical_10 <- no_data_numerical[idx, ]
```

```
# Mardia's multivariate normality test for No class
mult.norm(no_data_numerical_10)$mult.test
```

```
##          Beta-hat      kappa p-val
## Skewness 28.78310 12237.6155     0
## Kurtosis 92.07956    65.4226     0
```

Given p-value for Skewness and Kurtosis are less than 0.05, we reject the null hypothesis for both test, meaning the Multivariate normality assumption is not met for the **"No"** class.

If the second assumption (equal covariance between classes) is not met either, we can try QDA modeling or non parametric modeling as classification tree (this does not assume normality).

Equal variance (Homoscedasticity) for predictor variables within each class

LDA relies on Equal variance (Homoscedasticity) for predictor variables within each class. For Equality of Variances test we are going to use Levene's test given that our data set is not meeting th normality distribution assumption.

H0 (null hypothesis): The variances are equal between groups ("Yes" vs "No")

Ha (alternative hypothesis): The variances are not equal between groups ("Yes" vs "No")

```
library(car)

predictors_lda<-c("age", "duration","campaign", "previous", "emp.var.rate" ,
                  "cons.price.idx", "cons.conf.idx", "y")
data_lda<-subset(data, select = predictors_lda)

#Test for age
leveneTest(age ~ y, data=data_lda)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  195.87 < 2.2e-16 ***
##       28298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Test for duration
leveneTest(duration ~ y, data=data_lda)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  2100.8 < 2.2e-16 ***
##       28298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since p-values are very small (< 0.05) for both variables, we reject the null hypothesis. This means the assumption of equal variances is violated for both age and duration.

```
#Test for previous
leveneTest(previous ~ y, data=data_lda)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  113.49 < 2.2e-16 ***
##       28298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Test for emp.var.rate
leveneTest(emp.var.rate ~ y, data=data_lda)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value   Pr(>F)
## group     1   7.484 0.006229 **
##       28298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For the variables previous and **emp.var.rate**, the assumption of equal variances within classes is not met.

P-values are very small (< 0.05) for both variables.

```
#Test for cons.price.idx
leveneTest(cons.price.idx ~ y, data=data_lda)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value Pr(>F)
## group     1  1.8642 0.1722
##       28298
```

```
#Test for cons.conf.idx
leveneTest(cons.conf.idx ~ y, data=data_lda)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##          Df F value    Pr(>F)
## group     1  525.44 < 2.2e-16 ***
##       28298
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For the variable **"cons.price.idx"** variances are not significantly different between the classes, so we fail to reject the Null Hypothesis, meaning the Homoscedasticity assumption is met for this variable. On the contrary, **"cons.conf.idx"** violates the assumption since the P-value in the test is very small (close to zero).

Although the assumptions are not fully met, we will proceed with modeling the dependent variable using LDA. This decision is based on LDA's practical robustness, and our goal is to compare its accuracy with other models such as QDA, which does not require equal covariance assumptions—and classification tree, which is non-parametric.

Final Model LDA

Based on the previously explained rationale, the following LDA model was implemented:

```
#Model without nr.employed, euriborm3m
set.seed(2024)
predictors_lda<-c("age", "duration","campaign", "previous", "emp.var.rate" ,
                  "cons.price.idx", "cons.conf.idx", "y")

lda.fit3<-lda(y~age+duration+campaign+previous+emp.var.rate+cons.price.idx+cons.conf.idx,
              data = train_data)
lda.fit3
```

```
## Call:
## lda(y ~ age + duration + campaign + previous + emp.var.rate +
##     cons.price.idx + cons.conf.idx, data = train_data)
##
## Prior probabilities of groups:
##         no        yes
## 0.90145324 0.09854676
##
## Group means:
##          age duration campaign  previous emp.var.rate cons.price.idx
## no  38.69012 216.3286 2.432526 0.1091239    0.1699628       93.55115
## yes 39.31242 535.4836 2.061407 0.1887046   -1.1883012       93.24461
##     cons.conf.idx
## no      -40.75616
## yes     -40.28875
##
## Coefficients of linear discriminants:
##                            LD1
## age              0.0007575316
## duration         0.0042301488
## campaign         0.0119316439
## previous        -0.0999638333
## emp.var.rate    -0.6033524455
## cons.price.idx   0.6219596165
## cons.conf.idx    0.0577687803
```

Applying the fitted model to the test set:

```
actual = test_data$y
#Confusion matrix
y.pred = predict(lda.fit3, test_data)$class
```

Table 13. *Confusion Matrix for LDA*

	no	yes
no	4899	329
yes	204	229

The model correctly classified 4899 clients as not subscribing to the term deposit (True Negatives) and 229 clients as subscribing (True Positives). However, it also misclassified 329 clients who did not subscribe as subscribers (False Positives), and 204 clients who actually subscribed were missed by the model (False Negatives).

```
# Misclassification Rate
conf_mat_lda <- table(Predicted = y.pred, Actual = actual)
incorrect <- sum(conf_mat_lda) - sum(diag(conf_mat_lda))
total <- sum(conf_mat_lda)
misclassification_rate <- incorrect / total
paste("Misclassification Rate:", round(misclassification_rate*100, 4), "%")
```

```
## [1] "Misclassification Rate: 9.4153 %"
```

The misclassification rate for the final LDA model is 9.41%.

These results suggest that the model performs quite well in identifying clients who are unlikely to subscribe, but it struggles to accurately predict those who will subscribe. This is likely due to class imbalance, where non-subscribers dominate the dataset, even when the data split was balanced before generating the dataset split. As a result of the imbalanced data, the model is biased toward the majority class, and its ability to capture positive cases (subscribers) is limited.

```
train_pplot<-train_data[,!names(train_data) %in%
                          c("job","marital","education","default",
                            "housing","loan","contact",
                            "month","day_of_week", "poutcome" )]
##partimat(y~., data=train_pplot, method="lda")
```

Figure 7. *Partition Boundaries of Top Two Variable Pairs Using LDA*

```
library(klaR)
library(dplyr)
library(combinat)  # for combn
```

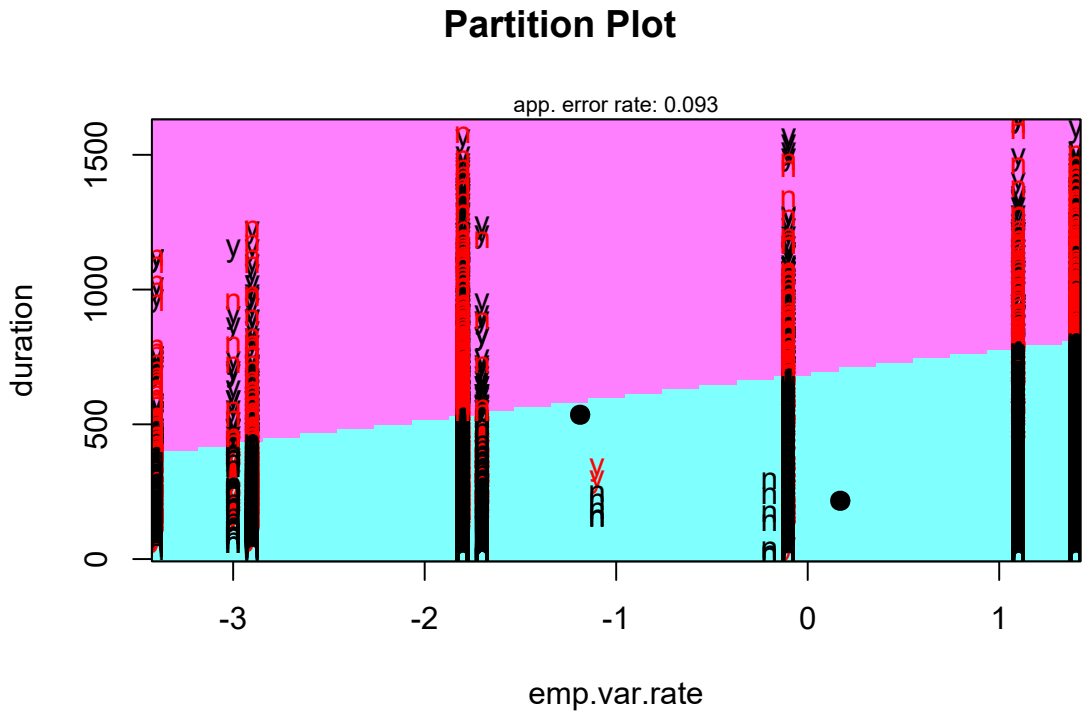
```
##
## Attaching package: 'combinat'
```

```
## The following object is masked from 'package:utils':
##
##     combn
```

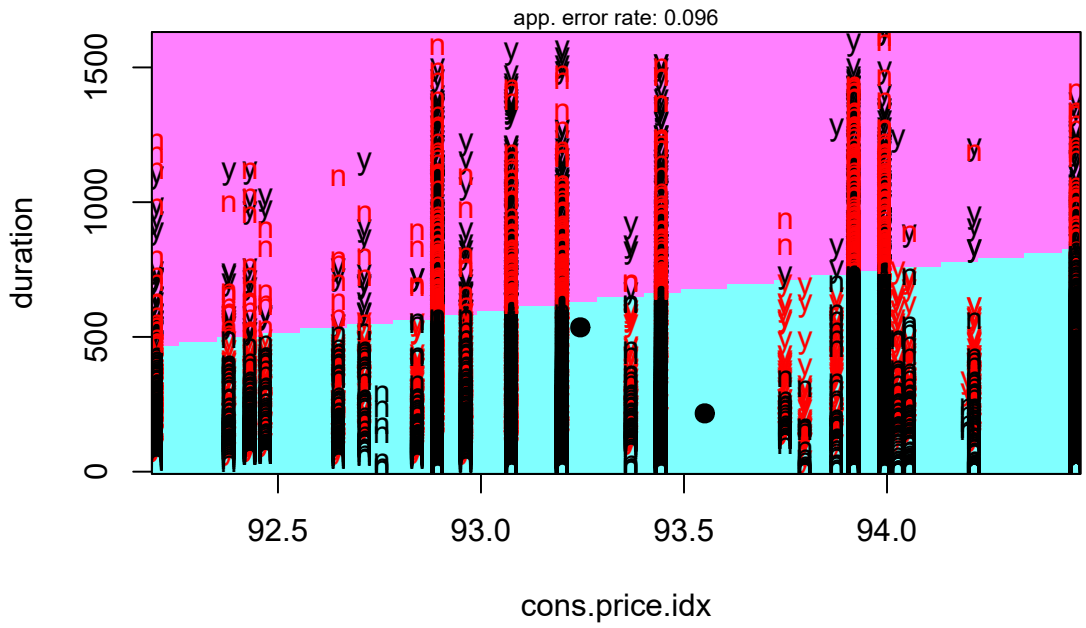
```
# Subset numeric predictors only
vars <- c("age", "duration", "campaign",
          "previous", "emp.var.rate",
          "cons.price.idx", "cons.conf.idx")
train_pp <- train_data[, c(vars, "y")]

# Store error rates
results <- data.frame(var1 = character(), var2 = character(), error = numeric())

# Loop over all 2-variable combinations
combinations <- combn(vars, 2)

for (i in 1:ncol(combinations)) {
  pair <- combinations[, i]
  formula <- as.formula(paste("y ~", paste(pair, collapse = " + ")))

  # Fit model and compute apparent error
  model <- lda(formula, data = train_pp)
  pred <- predict(model)$class
  error <- mean(pred != train_pp$y)

  results <- rbind(results, data.frame(var1 = pair[1], var2 = pair[2], error = error))
}

# Get top 2 lowest error pairs
top2 <- results %>% arrange(error) %>% head(2)
```

```r
# Plot them
for (i in 1:2) {
  v1 <- top2$var1[i]
  v2 <- top2$var2[i]
  cat(paste("Plotting: ", v1, " vs. ", v2, " | error rate = ", round(top2$error[i], 4), "\n"))
  print(partimat(as.formula(paste("y ~", v1, "+", v2)), data = train_pp, method = "lda"))
}
```

```
## Plotting:  duration  vs.  emp.var.rate  | error rate =  0.093
```




```
## NULL
## Plotting:  duration  vs.  cons.price.idx  | error rate =  0.0964
```




```
## NULL
```

Quadratic Discriminant Analysis (QDA)

Running QDA on all numerical variables:

```
set.seed(2024)
qda_fit <- qda(y~ age + duration + campaign + previous +
                 emp.var.rate + cons.price.idx +
                 cons.conf.idx + euribor3m + nr.employed,
               data = train_data)
qda_fit
```

```
## Call:
## qda(y ~ age + duration + campaign + previous + emp.var.rate +
##     cons.price.idx + cons.conf.idx + euribor3m + nr.employed,
##     data = train_data)
##
## Prior probabilities of groups:
##         no        yes
## 0.90145324 0.09854676
##
## Group means:
##          age duration campaign  previous emp.var.rate cons.price.idx
## no  38.69012 216.3286 2.432526 0.1091239    0.1699628       93.55115
## yes 39.31242 535.4836 2.061407 0.1887046   -1.1883012       93.24461
##     cons.conf.idx euribor3m nr.employed
```

```
## no      -40.75616  3.746647    5175.581
## yes     -40.28875  2.292769    5112.468
```

Table 14. *Confusion Matrix for QDA on all numerical variables*

	no	yes
no	4648	212
yes	455	346

The model correctly classified 4648 clients as not subscribing to the term deposit (True Negatives) and 346 clients as subscribing (True Positives). However, it also misclassified 212 clients who did not subscribe as subscribers (False Positives), and 455 clients who actually subscribed were missed by the model (False Negatives).

```
# Misclassification rate
conf_mat_qda <- table(Predicted = qda_pred, Actual = actual)
incorrect <- sum(conf_mat_qda) - sum(diag(conf_mat_qda))
total <- sum(conf_mat_qda)
misclassification_rate <- incorrect / total
paste("Misclassification Rate:", round(misclassification_rate*100, 4), "%")
```

```
## [1] "Misclassification Rate: 11.7824 %"
```

Misclassification rate for QDA in all numeric values is 11.78%.

There's a chance some features are redundant, and removing them might improve generalization.

```
train_data_numeric <- train_data %>%
  dplyr::select(where(is.numeric), y)
```

```
cor_matrix <- cor(dplyr::select(train_data_numeric, -y))
high_corr <- findCorrelation(cor_matrix, cutoff = 0.95)
names(train_data_numeric)[high_corr]
```

```
## [1] "euribor3m"
```

We will drop **'euribor3m'** because of it's high correlation and we will run the QDA model on all variables except for this column.

```
qda_reduced <- qda(y~ age + duration + campaign + previous +
                      emp.var.rate + cons.price.idx +
                      cons.conf.idx + nr.employed,
                    data = train_data)
qda_reduced
```

```
## Call:
## qda(y ~ age + duration + campaign + previous + emp.var.rate +
##     cons.price.idx + cons.conf.idx + nr.employed, data = train_data)
##
## Prior probabilities of groups:
```

```
##         no        yes
## 0.90145324 0.09854676
##
## Group means:
##          age duration campaign  previous emp.var.rate cons.price.idx
## no  38.69012 216.3286 2.432526 0.1091239    0.1699628       93.55115
## yes 39.31242 535.4836 2.061407 0.1887046   -1.1883012       93.24461
##     cons.conf.idx nr.employed
## no      -40.75616    5175.581
## yes     -40.28875    5112.468
```

```
actual = test_data$y
#Confusion matrix
qda_pred1 = predict(qda_reduced, test_data)$class
```

Table 15. *Confusion Matrix for QDA variables except for euribor3m*

	no	yes
no	4644	218
yes	459	340

The model correctly classified 4644 clients as not subscribing to the term deposit (True Negatives) and 340 clients as subscribing (True Positives). However, it also misclassified 218 clients who did not subscribe as subscribers (False Positives), and 459 clients who actually subscribed were missed by the model (False Negatives).

```
# Misclassification rate
conf_mat_qda1 <- table(Predicted = qda_pred1, Actual = actual)
incorrect <- sum(conf_mat_qda1) - sum(diag(conf_mat_qda1))
total <- sum(conf_mat_qda1)
misclassification_rate1 <- incorrect / total
paste("Misclassification Rate:", round(misclassification_rate1*100, 4), "%")
```

```
## [1] "Misclassification Rate: 11.959 %"
```

The misclassification rate didn't change much (11.95%), so we will drop **nr.employed** as well.

Final Model QDA

```
qda_reduced2 <- qda(y~ age + duration + campaign + previous
                    + emp.var.rate + cons.price.idx + cons.conf.idx,
                    data = train_data)
qda_reduced2
```

```
## Call:
## qda(y ~ age + duration + campaign + previous + emp.var.rate +
##     cons.price.idx + cons.conf.idx, data = train_data)
##
## Prior probabilities of groups:
```

```
##         no        yes
## 0.90145324 0.09854676
##
## Group means:
##          age duration campaign  previous emp.var.rate cons.price.idx
## no  38.69012 216.3286 2.432526 0.1091239    0.1699628       93.55115
## yes 39.31242 535.4836 2.061407 0.1887046   -1.1883012       93.24461
##     cons.conf.idx
## no      -40.75616
## yes     -40.28875
```

```
actual = test_data$y
#Confusion matrix
qda_pred2 = predict(qda_reduced2, test_data)$class
```

Table 16. *Confusion Matrix for QDA variables except for euribor3m and nr.employed*

	no	yes
no	4695	242
yes	408	316

The model correctly classified 4695 clients as not subscribing to the term deposit (True Negatives) and 316 clients as subscribing (True Positives). However, it also misclassified 242 clients who did not subscribe as subscribers (False Positives), and 408 clients who actually subscribed were missed by the model (False Negatives).

```
# Misclassification rate
conf_mat_qda2 <- table(Predicted = qda_pred2, Actual = actual)
incorrect <- sum(conf_mat_qda2) - sum(diag(conf_mat_qda2))
total <- sum(conf_mat_qda2)
misclassification_rate2 <- incorrect / total
paste("Misclassification Rate:", round(misclassification_rate2*100, 4), "%")
```

```
## [1] "Misclassification Rate: 11.4821 %"
```

By dropping **euribor3m** and **nr.employed**, the misclassification rate reduced very little to 11.48%.

Since **emp.var.rate** was also a high correlated column, I will drop that too to see if the model accuracy gets better or not.

```
qda_reduced3 <- qda(y~age+duration+campaign+previous+cons.price.idx+cons.conf.idx,
                    data = train_data)
qda_reduced3
```

```
## Call:
## qda(y ~ age + duration + campaign + previous + cons.price.idx +
##     cons.conf.idx, data = train_data)
##
## Prior probabilities of groups:
##         no        yes
## 0.90145324 0.09854676
```

```
##
## Group means:
##          age duration campaign  previous cons.price.idx cons.conf.idx
## no  38.69012 216.3286 2.432526 0.1091239       93.55115     -40.75616
## yes 39.31242 535.4836 2.061407 0.1887046       93.24461     -40.28875
```

```
actual = test_data$y
#Confusion matrix
qda_pred3 = predict(qda_reduced3, test_data)$class
```

Table 17. *Confusion Matrix for QDA variables except for euribor3m and nr.employed and emp.var.rate*

	no	yes
no	4764	291
yes	339	267

The model correctly classified 4764 clients as not subscribing to the term deposit (True Negatives) and 267 clients as subscribing (True Positives). However, it also misclassified 291 clients who did not subscribe as subscribers (False Positives), and 339 clients who actually subscribed were missed by the model (False Negatives).

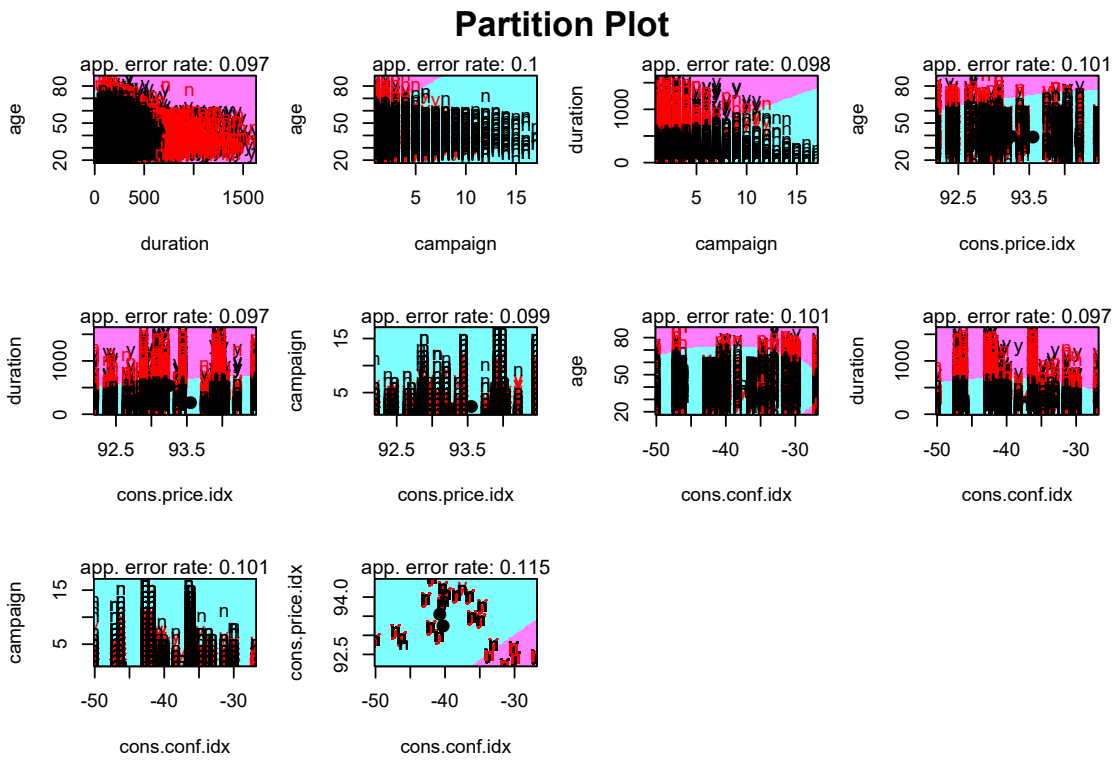
```
# Misclassification rate
conf_mat_qda3 <- table(Predicted = qda_pred3, Actual = actual)
incorrect <- sum(conf_mat_qda3) - sum(diag(conf_mat_qda3))
total <- sum(conf_mat_qda3)
misclassification_rate3 <- incorrect / total
paste("Misclassification Rate:", round(misclassification_rate3*100, 4), "%")
```

```
## [1] "Misclassification Rate: 11.1288 %"
```

The misclassification rate reduced to 11.12%.

This means that **euribor3m** and **nr.employed**and **emp.var.rate** are highly correlated and by removing them, we made our model quite a bit more stable.

Figure 8. *QDA Partition Plot*



This reduction suggests that eliminating redundant features can enhance QDA performance by reducing instability in the covariance matrices without sacrificing predictive power.

Classification Tree Model

As seen in the previous models (LDA and QDA), the assumptions of normality and equal variances were not met. We will now proceed to model a classification tree, which is a non-parametric method and does not rely on these two assumptions. Although classification trees require the observations to be independent, this condition is satisfied in our dataset, as we are working with individual records from different bank clients, and no relationships between clients are indicated.

Additionally, classification trees do not assume linear relationships between the dependent and independent variables, which means they are good for capturing non-linear patterns, as the model is based on recursive decision rules. While trees are generally robust to multicollinearity, highly correlated predictors can lead to instability. Therefore, we excluded the two variables with the highest observed correlation in the dataset: **"nr.employed"** and **"euribor3m"** (as we did in the previous models).

The tree will be evaluated using cross-validation to determine whether pruning is appropriate. This will help mitigate the risk of over-fitting and improve the model's generalization performance.

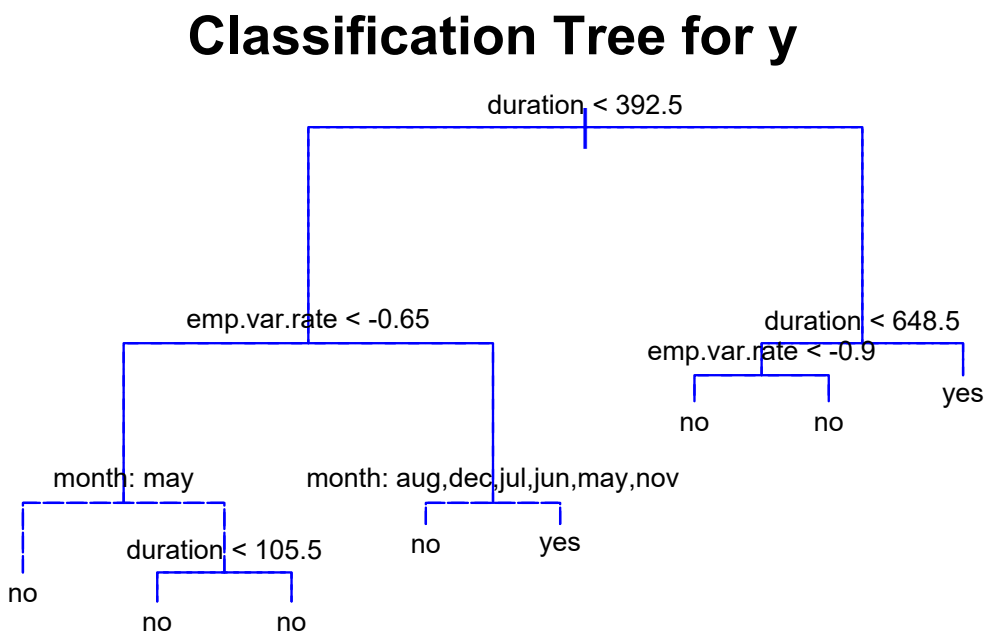
```
#Making sure the 2 variables with high correlation are dropped "nr.employed" and "euribor3m"
train_data <- train_data %>%
  select(-nr.employed, -euribor3m)

test_data<-test_data %>%
  select(-nr.employed, -euribor3m)
```

```r
#Making sure the dataset is complete withou splitting for the k fold
new_data <- rbind(train_data, test_data)
```

```r
#Apply a classification tree to the train part to establish relation between "y" and other variables.
tree_bank<-tree(factor(y)~., data=train_data)
summary(tree_bank)
```

```
##
## Classification tree:
## tree(formula = factor(y) ~ ., data = train_data)
## Variables actually used in tree construction:
## [1] "duration"     "emp.var.rate" "month"
## Number of terminal nodes:  8
## Residual mean deviance:  0.3971 = 8987 / 22630
## Misclassification error rate: 0.09735 = 2204 / 22639
```

Figure 9. *Full Classification Tree*



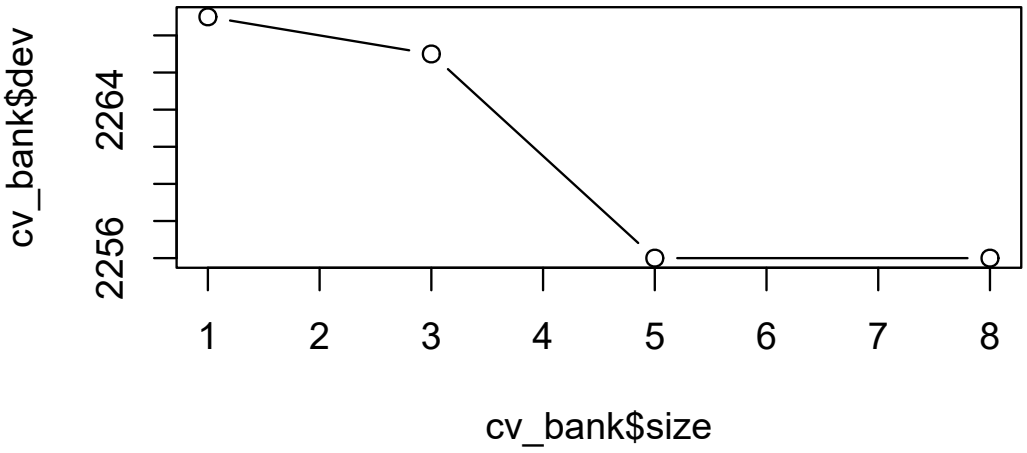

```r
tree_bank_pred<-predict(tree_bank,test_data,type = "class")
```

Table 18. *Confusion Matrix for Classification Tree Model*

	no	yes
no	4901	369
yes	202	189

Perform cross-validation:

Figure 10. *Classification Error by Tree Size from Cross-Validation Pruning*

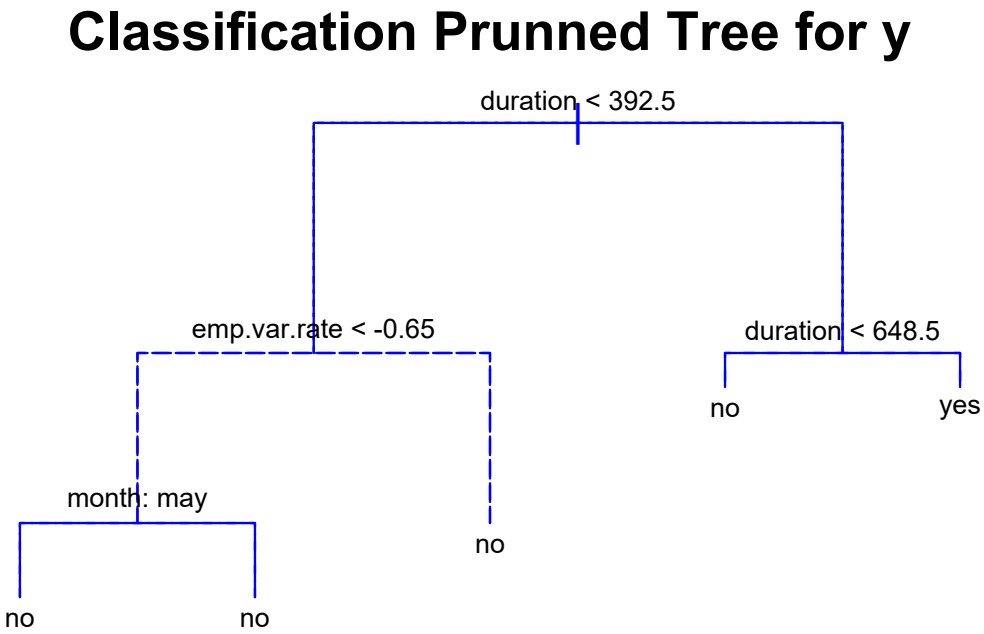


Final Model Classification Tree Model

Based on cross-validation pruning, the tree will be pruned to 5 terminal nodes for the final model

```
prune_bank=prune.tree(tree_bank,best=5)
```

Figure 11. *Pruned Classification Tree Based on Cross-Validation*



re-calculate the misclassification rate:

```
#Apply the prune tree to the test set
prune_bank_pred<-predict(prune_bank,test_data,type="class")
```

Table 19. *Confusion Matrix for the Final Classification Tree Model*

```
kable(table(Predicted=prune_bank_pred,Actual=test_data$y))
```

	no	yes
no	4903	378
yes	200	180

```
# Misclassification Rate
confusion_prune=table(Predicted=prune_bank_pred,Actual=test_data$y)
incorrect <- sum(confusion_prune) - sum(diag(confusion_prune))
total <- sum(confusion_prune)
misclassification_rate <- incorrect / total
paste("Misclassification Rate:", round(misclassification_rate*100, 4), "%")
```

```
## [1] "Misclassification Rate: 10.2102 %"
```

The pruned classification tree model correctly identified 4,903 negative cases (clients who did not subscribe) and 180 positive cases (clients who did subscribe). However, it also produced 378 false positives (predicted as subscribers, but they were not) and 200 false negatives (predicted as non-subscribers, but they actually subscribed).From this confusion matrix, we can estimate the following performance metrics:

- Accuracy: 89.8%. This represents the proportion of correctly classified observations out of all predictions.
- Precision for yes: 32.2%. This is the proportion of correct positive predictions among all predicted positives (the true and false ones).
- Sensitivity for yes: 47.4%. This indicates the model's ability to correctly identify actual subscribers $(180/(180 + 200))$.
- Specificity is: 92.8%. This is the true negative rate and reflects the model's ability to correctly classify non-subscribers.(TN/Total Neg$= 4903/(4903 + 378)$).
- Misclassification rate of 10.2%. This is estimated by all the incorrectly classified over the total predicted and it is the complement of the accuracy.

Overall, the model performs well in terms of general accuracy and specificity, which is expected given the dataset's imbalance toward non-subscribers. Although class balancing was applied before splitting the data into training and test sets, the prediction of the minority class **('yes')** remains challenging.

Only 47.4% of actual subscribers were correctly identified, which indicates poor sensitivity. This highlights the model's limited ability to detect clients who will subscribe, which is a crucial insight in applications where identifying positive cases is a priority.

K-fold Cross-validation.

A 10-fold cross-validation procedure was used to compare the performance of the Linear Discriminant Analysis and Classification Tree models.

```
#10 folds partition
folds<-createFolds(factor(new_data$y), k=10)
```

Table 20 and 21. *Distribution of Class Labels in Fold 1 and Fold 10*

Var1	Freq
no	2551
yes	279

Var1	Freq
no	2551
yes	279

Misclassification for GLM with K folds

```
misclassification <- function(idx) {
  Train <- new_data[-idx, ]
  Test  <- new_data[idx, ]
  fit <- glm(y~job+education+contact+month+day_of_week+duration+campaign+emp.var.rate+cons.conf.idx,
             family = binomial,
             data = Train)
  prob <- predict(fit, newdata = Test, type = "response")
  pred <- rep("no", length(prob))
  pred[prob >= 0.5] <- "yes"
  return(1 - mean(pred == Test$y))
}

mis_rate=lapply(folds,misclassification)

cv_error_glm=mean(as.numeric(mis_rate))
paste("Misclassification Rate for glm:", round(cv_error_glm*100, 2), "%")
```

```
## [1] "Misclassification Rate for glm: 9 %"
```

Misclassification for LDA with K folds

```
misclassification<-function(idx){
  Train<-new_data[-idx,]
  Test<-new_data[idx,]
  fit<-lda(y~ age + duration + campaign + previous + emp.var.rate +
    cons.price.idx + cons.conf.idx, data=train_data)
```

```
  pred<-predict(fit,Test)
  return(1-mean(pred$class==Test$y))
}

#Passing the function in the folds
mis_rate=lapply(folds,misclassification)

#Average of the missclasification
cv_error_lda=mean(as.numeric(mis_rate))
paste("Misclassification Rate for lda:", round(cv_error_lda*100, 2), "%")
```

```
## [1] "Misclassification Rate for lda: 9.1 %"
```

Misclassification for QDA with K folds:

```
## [1] "Misclassification Rate for qda: 11.15 %"
```

Misclassification for Classification Tree with K folds:

```
misclassification<-function(idx){
  Train <- new_data[-idx, ]
  Test<- new_data[idx, ]
  fit <-prune.tree(tree_bank,best=7)
  pred  <- predict(fit, Test, type = "class")
  return(1 - mean(pred == Test$y))
}

#Passing the function in the folds
mis_rate=lapply(folds,misclassification)


#Average of the missclasification
cv_error_prunnedT=mean(as.numeric(mis_rate))
paste("Misclassification Rate for prunned Tree:", round(cv_error_prunnedT*100, 2), "%")
```

```
## [1] "Misclassification Rate for prunned Tree: 9.86 %"
```

Table 22. *Summary of Classification Models: Misclassification Rates and Included Predictors*

Model	Misclassification.Rate	Number.of.Variables...Nodes
Logistic Regression	9.00%	9
Linear Discriminant Analysis	9.1%	7
Quadratic Discriminant Analysis	11.15%	6
Classification Tree	9.86%	7 nodes

Based on the 10-fold cross-validation results, Logistic Regression (GLM) emerged as the model with the lowest misclassification rate (9.00%), followed closely by Linear Discriminant Analysis (LDA) with 9.1%. These findings suggest that both models offer strong predictive performance, with GLM slightly outperforming in terms of classification accuracy.

However, model evaluation should not rely solely on misclassification rates. Other performance metrics—such as precision, sensitivity, and specificity—provide a more nuanced understanding of each model's strengths and limitations, particularly in the context of marketing campaigns where the cost of false positives and false negatives can differ substantially.

The second table summarizes the classification metrics for each model before applying cross-validation:

Table 23. *Classification Performance Metrics Before Cross-Validation*

Model	Assumptions	Accuracy	MR	Precision..Yes.	Sensitivity..Yes.	Specificity..No.
GLM	Met	90.7%	9.3%	55.1%	28.9%	97.4%
LDA	Not met	90.6%	9.4%	41.0%	52.9%	93.7%
QDA	Not met	88.9%	11.1%	44.0%	47.9%	93.4%
Classification Tree	Met	89.8%	10.2%	47.4%	32.2%	92.8%

GLM is recommended if the objective is to minimize false positives and improve overall prediction accuracy. In contrast, LDA may be more suitable when the marketing strategy emphasizes maximizing subscriber detection, accepting a higher false positive rate as a trade-off.

Ultimately, the optimal model selection depends on the specific goals and resource constraints of the campaign. These performance results provide a robust basis for aligning statistical accuracy with operational effectiveness.

Conclusion

Demographic attributes showed limited influence on subscription behavior. While only education was significant in the logistic regression model, both age and education were included in the LDA and QDA models, suggesting these variables may contribute under certain modeling assumptions, but their overall impact appears modest.

Among economic indicators, the employment variation rate was the only variable retained, and even then, only in one model. This suggests that macroeconomic fluctuations have limited direct impact on individual subscription decisions within the context of this campaign.

In contrast, campaign-related variables such as the month of contact and call duration were consistently identified as important across all models, GLM, LDA, and the classification tree. This consistency underscores the operational importance of timing and customer engagement length as key drivers of campaign effectiveness.

The outcome of previous campaigns ('poutcome') was only considered in the LDA model and excluded from the others, indicating that past campaign results have limited predictive value in this context and may not be a key factor influencing current customer decisions.

Ultimately, the choice of the "best" model depends on the marketing campaign's strategic priorities. While the GLM model achieved the lowest misclassification rate (9.00%), followed closely by LDA (9.10%). Other performance metrics, such as sensitivity, precision, and specificity must also be considered. For instance, LDA exhibited higher sensitivity (52.9%) compared to GLM (28.9%), making it more suitable in scenarios where correctly identifying potential subscribers (true positives) is a priority.

Therefore, model selection should be aligned with the specific objectives of the marketing strategy. If minimizing false positives is more critical, a model with higher specificity (such as GLM) may be preferred. Conversely, if capturing more true positives is essential, LDA may be the more appropriate choice despite a slightly higher misclassification rate.

References

Yamahata, H. (n.d.). Bank Marketing. Kaggle. https://www.kaggle.com/datasets/henriqueyamahata/bank-marketing