Predicting Customer Subscription to Long-Term Bank Deposits: An Exploratory and Statistical Analysis

Name: Valentina Mercieca

Student ID: 12696474

Module: Visualising Data

Due: 24th March, 2025

Table of Contents

List of Tables

List of Figures

1. Introduction

Marketing campaigns are essential for customer acquisition and business growth. Banks use direct telemarketing to promote financial products like long-term deposits—a strategy that can increase acquisition by 28%, retention by 12%, and revenue by 16% (Basha, 2024). By using data-driven models, banks can predict the likelihood of customer subscriptions, allowing for more precise targeting, greater efficiency, and reduced costs.

This report analyses a Portuguese bank dataset from the UCI Machine Learning Repository (Moro, Rita & Cortez, 2014), containing client demographics, telemarketing details, macroeconomic indicators, and a binary target variable indicating subscription outcomes (Table 1).

	Variable	Type	Description
Customer Profile	age	Numeric	Client's age in years
	job	Categorical	Client's occupation
	marital	Categorical	Marital status
	k	Categorical	Education level
	default	Categorical	Has credit in default?
	housing	Categorical	Has housing loan?
	loan	Categorical	Has personal loan?
Campaign Engagement	contact	Categorical	Contact method
	month	Categorical	Last contact month
	day of week	Categorical	Last contact day
	campaign	Numeric	Number of contacts in this campaign
	duration	Numeric	Duration of call in seconds
	pdays	Numeric	Days since last contact from prior campaign
	previous	Numeric	Number of prior campaign contacts
	poutcome	Categorical	Previous campaign result
Economic Indicators	emp.var.rate	Numeric	Quarterly employment change rate
	cons.price.idx	Numeric	Monthly consumer price index
	cons.conf.idx	Numeric	Monthly consumer confidence index
	euribor3m	Numeric	Daily 3-month Euribor rate
	nr.employed	Numeric	Quarterly number of employees
Target	y	Categorical	Binary response variable indicating whether a customer opened a long-term deposit account or not

Table 1 Variables in Dataset

2. Exploratory Data Analysis (EDA)

2.1 Work Process

EDA, pioneered by Tukey (1977), begins with a clear understanding of the problem at hand. Before engaging with the data, it is essential to define the objective – whether to uncover patterns, detect anomalies, or prepare for predictive modelling (IBM, no date).

As shown in Figure 1, the process starts with importing and inspecting the dataset, uncovering issues like missing values, outliers, and unexpected distributions. These are addressed to ensure data integrity. Once cleaned, univariate, bivariate, and correlation analyses reveal distributions, relationships, and dependencies using visualisation techniques. The dataset is then refined by removing irrelevant attributes in feature selection.

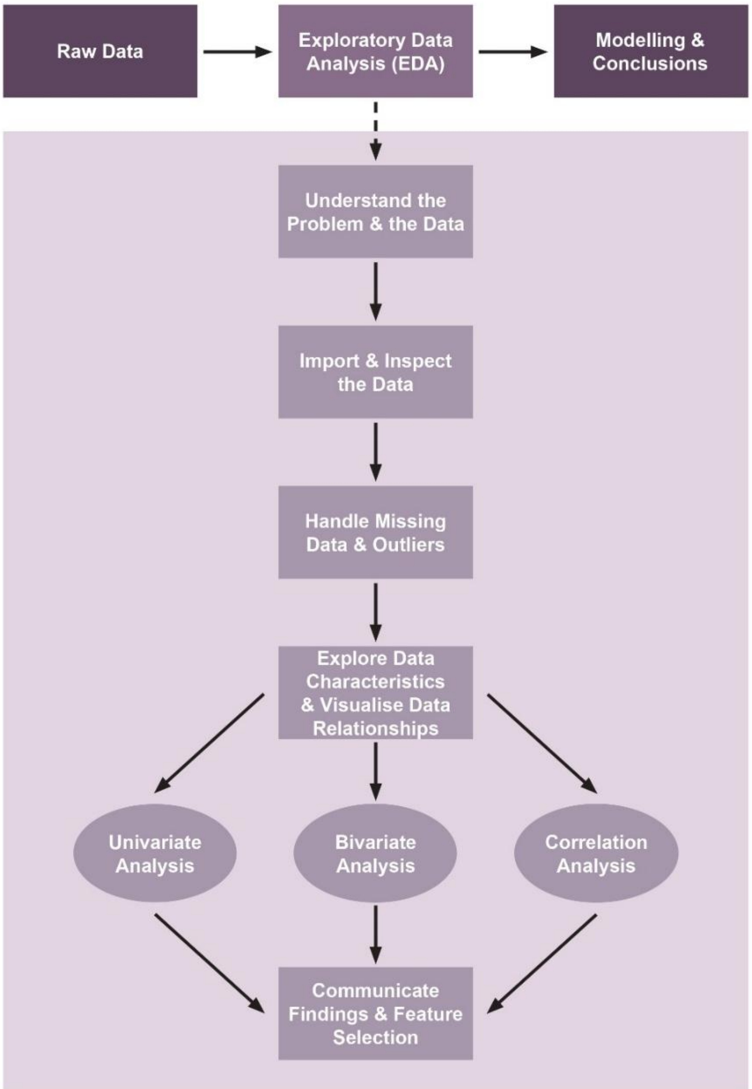

Figure 1 EDA Diagram

2.2 Understanding the Dataset

A preliminary review shows that the dataset contains 4,100 observations and 21 variables, comprising numerical and categorical data types. The dependent variable, *y*, is binary, indicating whether a client subscribed ("yes") or not ("no"). Independent variables cover client demographics, financial background, marketing interactions, and economic indicators (Figure 2).

Figure 3 confirms the dataset is complete, with no null or duplicate values. However, some variables require attention. The *pdays* variable is predominantly set to 999, suggesting most clients were not previously contacted – indicating it may be better treated as categorical. The *campaign* variable ranges from 1 to 35 contacts with a median of 2, warranting further analysis. Likewise, *duration* varies from 0 to over 3,600 seconds, requiring assessment to determine if it signals client interest or noise. For clarity, the *k* variable was renamed *education*.

```
> str(original_data)
'data.frame':	4100 obs. of  21 variables:
 $ age           : int  30 39 25 38 47 32 32 41 31 35 ...
 $ job           : chr  "blue-collar" "services" "services" "services" ...
 $ marital       : chr  "married" "single" "married" "married" ...
 $ k             : chr  "basic.9y" "high.school" "high.school" "basic.9y" ...
 $ default       : chr  "no" "no" "no" "no" ...
 $ housing       : chr  "yes" "no" "yes" "unknown" ...
 $ loan          : chr  "no" "no" "no" "unknown" ...
 $ contact       : chr  "cellular" "telephone" "telephone" "telephone" ...
 $ month         : chr  "may" "may" "jun" "jun" ...
 $ day_of_week   : chr  "fri" "fri" "wed" "fri" ...
 $ duration      : int  487 346 227 17 58 128 290 44 68 170 ...
 $ campaign      : int  2 4 1 3 1 3 4 2 1 1 ...
 $ pdays         : int  999 999 999 999 999 999 999 999 999 999 ...
 $ previous      : int  0 0 0 0 0 2 0 0 1 0 ...
 $ poutcome      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...
 $ emp.var.rate  : num  -1.8 1.1 1.4 1.4 -0.1 -1.1 -1.1 -0.1 -0.1 1.1 ...
 $ cons.price.idx: num  92.9 94 94.5 94.5 93.2 ...
 $ cons.conf.idx : num  -46.2 -36.4 -41.8 -41.8 -42 -37.5 -37.5 -42 -42 -36.4 ...
 $ euribor3m     : num  1.31 4.86 4.96 4.96 4.19 ...
 $ nr.employed   : num  5099 5191 5228 5228 5196 ...
 $ y             : chr  "no" "no" "no" "no" ...
```

Figure 2 Inspecting the Structure

```
> summary(original_data)
      age             job              marital               k               default
 Min.   :18.00   Length:4100        Length:4100        Length:4100        Length:4100
 1st Qu.:32.00   Class :character   Class :character   Class :character   Class :character
 Median :38.00   Mode  :character   Mode  :character   Mode  :character   Mode  :character
 Mean   :40.12
 3rd Qu.:47.00
 Max.   :88.00
   housing              loan             contact             month           day_of_week
 Length:4100        Length:4100        Length:4100        Length:4100        Length:4100
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


    duration          campaign          pdays          previous          poutcome
 Min.   :   0.0   Min.   : 1.000   Min.   :  0.0   Min.   :0.0000   Length:4100
 1st Qu.: 103.0   1st Qu.: 1.000   1st Qu.:999.0   1st Qu.:0.0000   Class :character
 Median : 181.0   Median : 2.000   Median :999.0   Median :0.0000   Mode  :character
 Mean   : 256.8   Mean   : 2.539   Mean   :960.2   Mean   :0.1907
 3rd Qu.: 317.0   3rd Qu.: 3.000   3rd Qu.:999.0   3rd Qu.:0.0000
 Max.   :3643.0   Max.   :35.000   Max.   :999.0   Max.   :6.0000
  emp.var.rate      cons.price.idx  cons.conf.idx     euribor3m      nr.employed         y
 Min.   :-3.40000   Min.   :92.20   Min.   :-50.8   Min.   :0.635   Min.   :4964   Length:4100
 1st Qu.:-1.80000   1st Qu.:93.08   1st Qu.:-42.7   1st Qu.:1.334   1st Qu.:5099   Class :character
 Median : 1.10000   Median :93.75   Median :-41.8   Median :4.857   Median :5191   Mode  :character
 Mean   : 0.08517   Mean   :93.58   Mean   :-40.5   Mean   :3.621   Mean   :5166
 3rd Qu.: 1.40000   3rd Qu.:93.99   3rd Qu.:-36.4   3rd Qu.:4.961   3rd Qu.:5228
 Max.   : 1.40000   Max.   :94.77   Max.   :-26.9   Max.   :5.045   Max.   :5228
>
> #Checking missing values (NAs)
> colSums(is.na(original_data))
           age            job        marital              k        default        housing
             0              0              0              0              0              0
          loan        contact          month    day_of_week       duration       campaign
             0              0              0              0              0              0
         pdays       previous       poutcome   emp.var.rate cons.price.idx  cons.conf.idx
             0              0              0              0              0              0
     euribor3m    nr.employed              y
             0              0              0
>
> #Checking for duplicate rows
> original_data[duplicated(original_data), ]
 [1] age            job            marital        k              default        housing
 [7] loan           contact        month          day_of_week    duration       campaign
[13] pdays          previous       poutcome       emp.var.rate   cons.price.idx cons.conf.idx
[19] euribor3m      nr.employed    y
<0 rows> (or 0-length row.names)
> sum(duplicated(original_data))
[1] 0
```

Figure 3 Summary Statistics and Checks

2.3 Handling Missing Data and Outliers

To prepare the dataset for analysis, initial preprocessing was conducted to address "unknown" values in categorical variables (Figure 4). Since removing rows could lead to data loss and selection bias, a strategic approach was taken to handle missing values while preserving valuable information.

```
> #Creating copy of original_data
> updated_data <- original_data
> #Calculate count and percentage of "unknown" values for each categorical variable
> unknown_summary <- updated_data %>%
+   summarise(across(where(is.character), ~ sum(. == "unknown"))) %>%
+   pivot_longer(cols = everything(), names_to = "Variable", values_to = "Unknown_Count") %>%
+   mutate(Percentage = round((Unknown_Count / nrow(updated_data)) * 100, 2)) %>%
+   arrange(desc(Unknown_Count))
> unknown_summary
# A tibble: 11 × 3
   Variable    Unknown_Count Percentage
   <chr>               <int>      <dbl>
 1 default               799      19.5 
 2 education             167       4.07
 3 housing               104       2.54
 4 loan                  104       2.54
 5 job                    39       0.95
 6 marital                11       0.27
 7 contact                 0       0   
 8 month                   0       0   
 9 day_of_week             0       0   
10 poutcome                0       0   
11 y                       0       0   
```

Figure 4 Calculating the Proportion of "unknown" Values

For *housing*, *loan*, *job*, and *marital*, the mode was imputed due to the low proportion of "unknown" values, minimising distribution impact (Figure 5). However, *default* had 19.5% unknowns, with only one "yes" response. Given its lack of meaningful variation, it was excluded from the study.

```
> #Replacing "unknown" values with mode for housing, loan, job, marital
> #Reason: lower levels of "unknown", hence imputation will not greatly affect distribution
> unknown_with_mode <- function(column) {
+ 
+   #Finding mode
+   mode_value <- names(sort(table(column), decreasing = TRUE))[1]
+ 
+   #Replacing "unknown" with mode
+   column <- ifelse(column == "unknown", mode_value, column)
+ 
+   #Converting to factor
+   return(as.factor(column))
+ }
> #Applying function to housing, loan, job, marital
> #To get housing_updated, loan_updated, job_updated, marital_updated
> updated_data <- updated_data %>%
+   mutate(housing_updated = unknown_with_mode(housing),
+          loan_updated = unknown_with_mode(loan),
+          job_updated = unknown_with_mode(job),
+          marital_updated = unknown_with_mode(marital)
+   )
```

Figure 5 Mode Imputation

The “unknown” category in *education* was retained to avoid bias and preserve data integrity. To simplify analysis, all basic education levels (“basic.4y”, “basic.6y”, “basic.9y”) and the single “illiterate” case were merged into a "Basic" category (Figure 6).

```
> #Grouping basic.6y, basic.9y, and basic.4y as "Basic Education" for simplification
> #Also including 1 case "illiterate" under basic for model stability
> updated_data <- updated_data %>%
+   mutate(education_updated = case_when(
+     education == "basic.6y" ~ "basic",
+     education == "basic.9y" ~ "basic",
+     education == "basic.4y" ~ "basic",
+     education == "illiterate" ~ "basic",
+     TRUE ~ education
+   ))
```

Figure 6 Grouping Education

Analysing outliers was a crucial step in ensuring the dataset remained reliable for further study. Boxplots (Figure 7) revealed extreme values in *age*, *campaign*, *duration*, *previous*, and *pdays*.

The *age* variable was right-skewed, with some clients reaching 88 years old. Although these extremes may not represent the broader population, they are still valid. Hence, instead of removal, they were retained for further analysis. Call *duration* varied widely, peaking at 3,643 seconds, far above the 181-second median. To reduce distortion, values above the 99th percentile (1,221.1s) were capped. Similarly, *campaign* was capped above the 99th percentile (13.01 contacts) to limit undue influence. Figure 8 confirms these adjustments preserved overall distributions.

The *previous* variable was right-skewed, mostly clustered at zero but reaching six, requiring careful consideration in modelling. Meanwhile, *pdays*, dominated by 999 values, was converted into a categorical feature to differentiate prior contact levels (Figure 9).

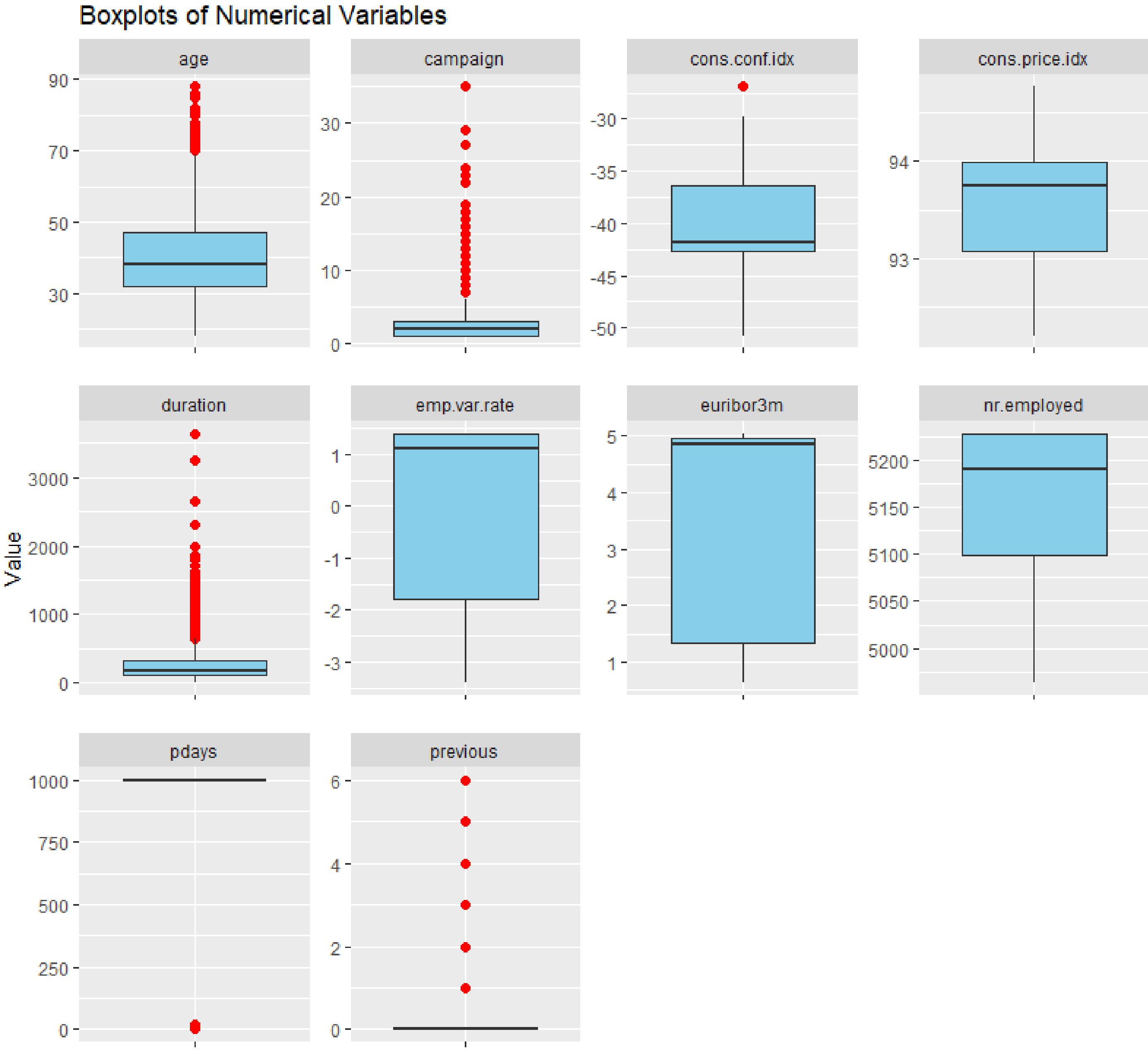

Figure 7 Boxplots to Investigate Noise

```
> #Function to cap extreme values at the 99th percentile
> cap_outliers <- function(x) {
+ 
+   #Extracting the 99th percentile
+   upper_limit <- quantile(x, 0.99)
+ 
+   #Capping values above the upper_limit
+   x <- ifelse(x > upper_limit, upper_limit, x)
+ 
+   return(x)
+ }
> updated_data <- updated_data %>%
+   mutate(campaign_capped = cap_outliers(campaign),
+          duration_capped = cap_outliers(duration)
+   )
> #Checking differences in distributions
> summary(updated_data$campaign)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   1.000   2.000   2.539   3.000  35.000 
> summary(updated_data$campaign_capped)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   1.000   2.000   2.485   3.000  13.010 
> summary(updated_data$duration)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    0.0   103.0   181.0   256.8   317.0  3643.0 
> summary(updated_data$duration_capped)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    0.0   103.0   181.0   252.8   317.0  1221.1 
```

Figure 8 Function to Cap Outliers

```
> #Converting pdays into a categorical variable
> updated_data <- updated_data %>%
+   mutate(pdays_category = case_when(
+     pdays == 999 ~ "Not Previously Contacted",
+     pdays < 7 ~ "Recently Contacted",
+     pdays >= 7 & pdays < 30 ~ "Contacted over a Week Ago",
+     pdays >= 30 & pdays < 999 ~ "Contacted over a Month Ago"
+   ))
> table(updated_data$pdays_category)

Contacted over a Week Ago  Not Previously Contacted        Recently Contacted 
                       39                      3940                       121 
```

Figure 9 Grouping pdays

2.4 Univariate Analysis

Understanding the distribution of the dependent variable *y* is essential before proceeding with thorough analysis. The dataset is imbalanced, with 89% of clients not subscribing (Figure 10). To address this, subsequent analyses compare subscribed and non-subscribed clients separately.

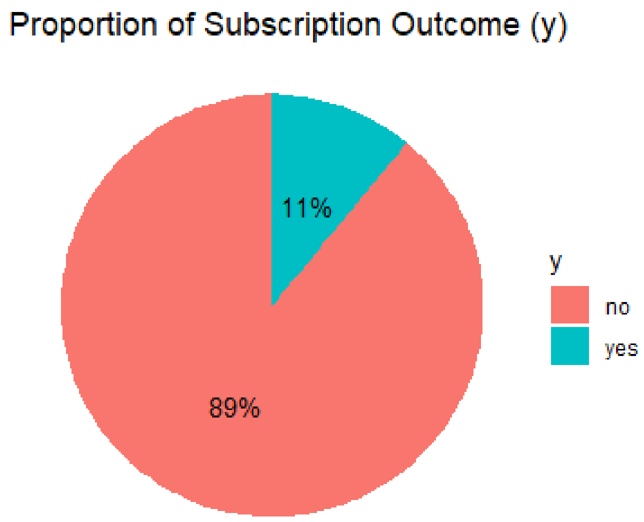

Figure 10 Pie Chart of Target Variable

Density plots (Figure 11) show relationships between numerical variables and the likelihood of subscription. The *age* distribution is similar across both groups, suggesting it is not a strong predictor. However, *duration_capped* shows that longer calls are associated with a higher proportion of subscriptions, making it a critical variable for modelling.

An inverse trend is exhibited in *campaign_capped*, where increased calls reduce subscription likelihood possibly due to customer fatigue. Similarly, *previous* indicates that clients with multiple past interactions showed a decline in subscriptions, possibly due to over-marketing effects.

Wilcoxon tests (Table 2) confirm these findings at the 0.05 significance level – *duration_capped*, *campaign_capped*, and *previous* are significantly associated with subscription, while *age* is not.

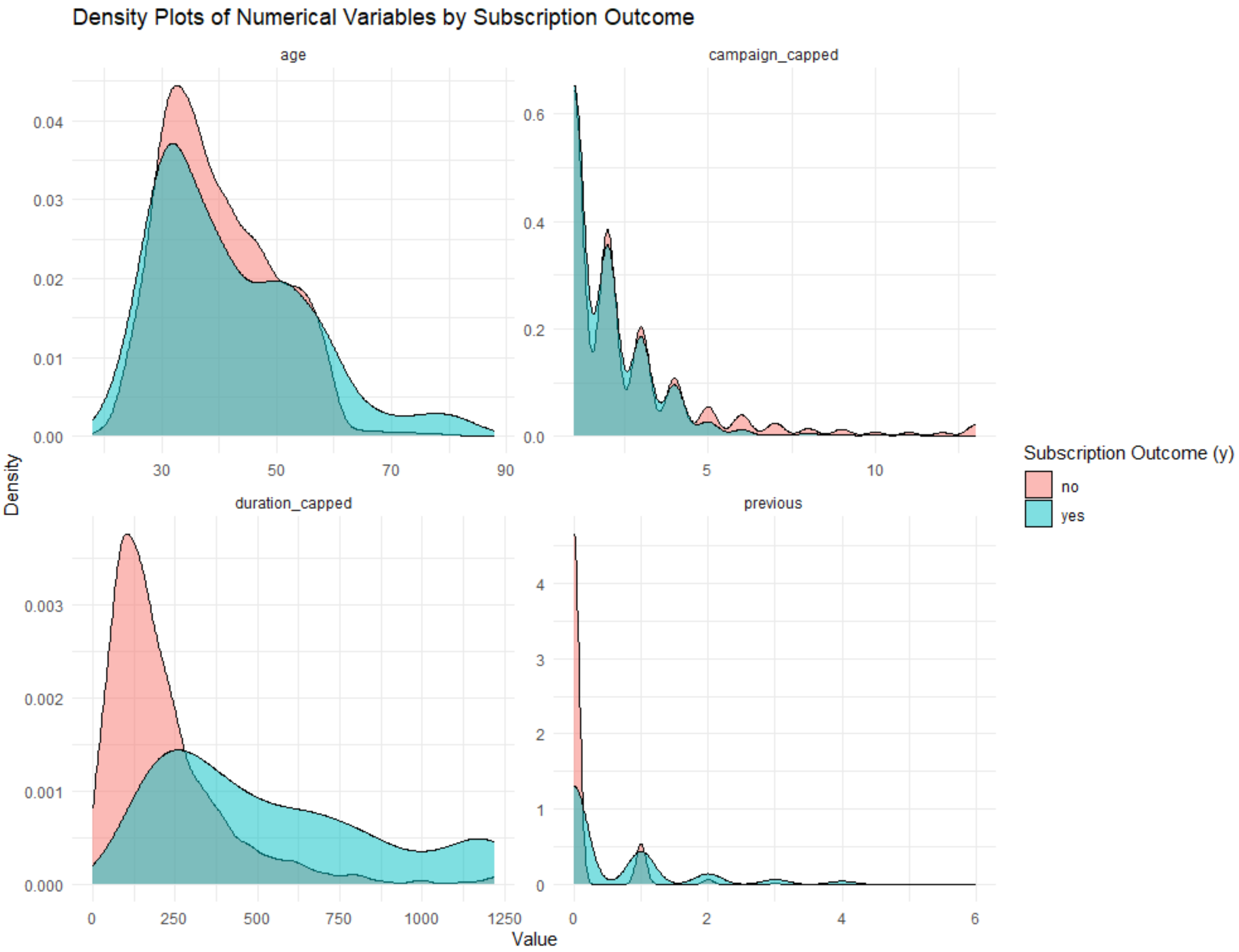

Figure 11 Density Plots of Numerical Variables

Variable	W Statistic	p-value	Significance
age	788209	0.1439	Not Significant
duration_capped	293345	< 2.2e-16	Significant
campaign_capped	911677	8.091e-05	Significant
previous	622215	< 2.2e-16	Significant

Table 2 Wilcoxon Tests

Stacked bar charts (Figure 12) show that *day_of_week* has no impact on subscription rates, with consistent "yes" proportions across all days. Likewise, *housing_updated* and *loan_updated* exhibit similar subscription rates, indicating weak predictive value. Given their minimal influence, these variables will be excluded from further analysis.

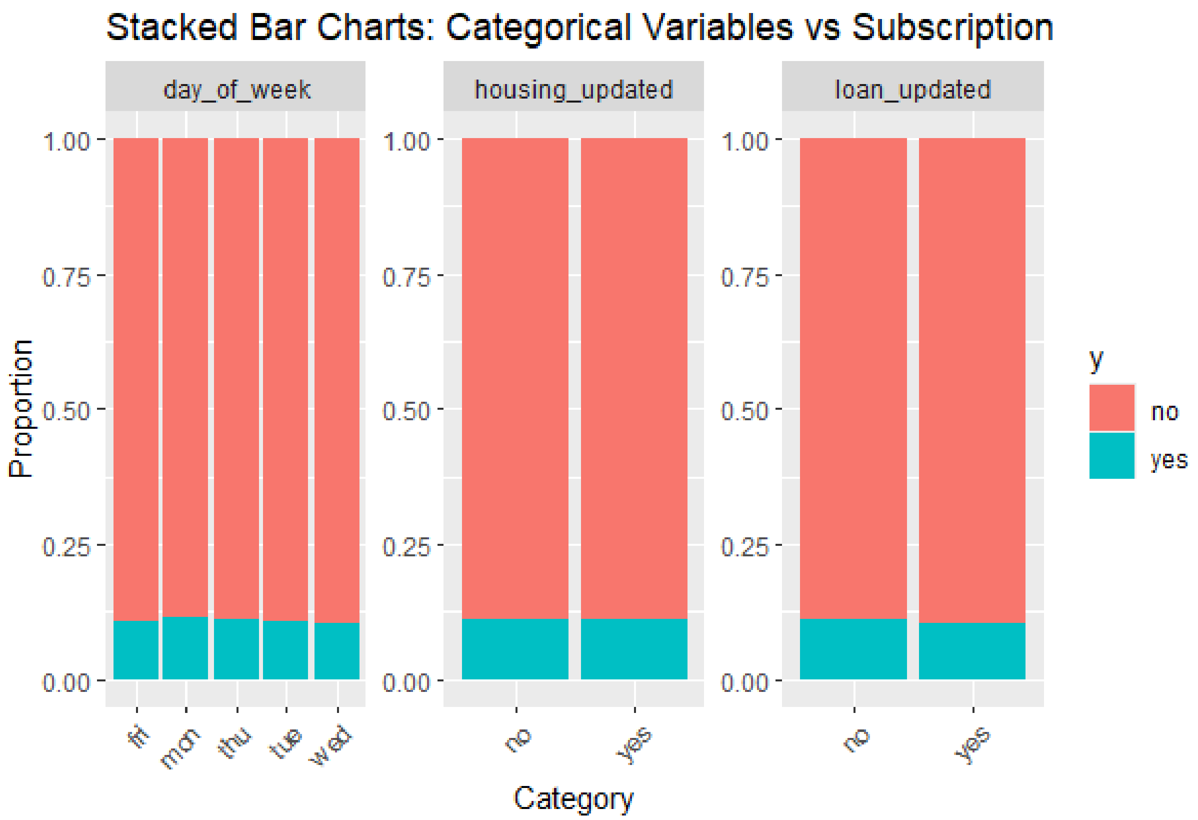

Figure 12 Stacked Bar Charts of Weak Categorical Predictors

Stacked bar charts (Figure 13) show that cellular *contact* leads to higher subscription rates than landlines, suggesting greater effectiveness. Single clients are slightly more likely to subscribe than married or divorced ones, though the difference is minor. Impact of prior engagement is highlighted in *pdays_category* since recently contacted clients subscribe most, while those never contacted are least likely. Similarly, *poutcome* shows that previous subscribers are far more likely to subscribe again, making it a strong predictor.

Figure 13 Stacked Bar Charts for Stronger Categorical Predictors

Heatmaps (Figure 14) suggest financial stability influences decisions as students and retirees have the highest subscription rates while blue-collar workers and entrepreneurs are least likely.

A seasonal trend emerges as May had the lowest conversion rate despite having the highest outreach. March and December saw higher success rates, indicating timing affects campaign success.

Although higher-educated clients subscribe frequently, their share among non-subscribers is also high. With minimal differences between education levels and the "unknown" category having the highest acceptance rate, *education* lacks predictive power and will be excluded.

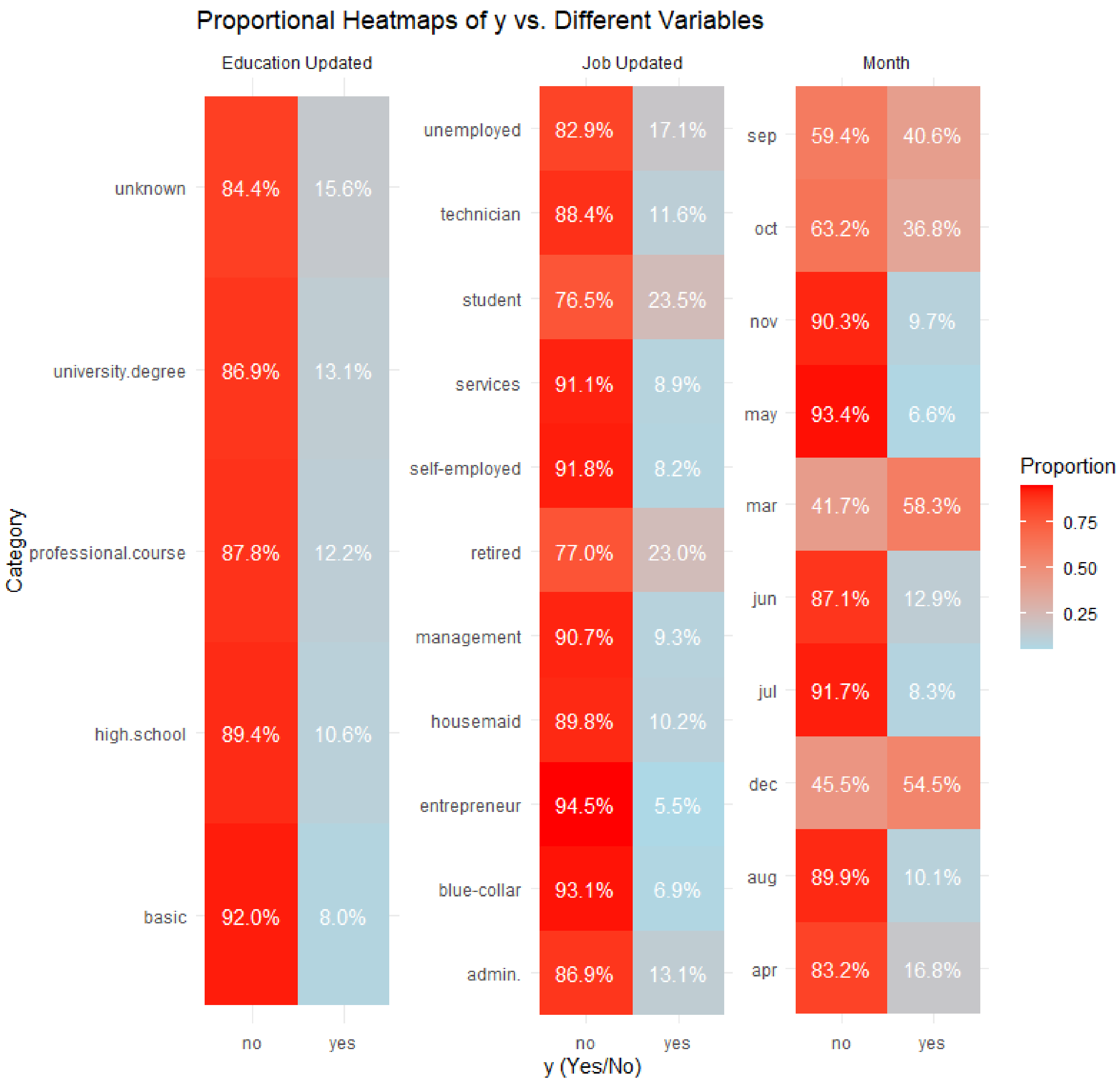

Figure 14 Heatmaps for education_updated, job_updated, and month

Chi-square tests (Table 3) confirm that *job_updated* and *month* significantly impact subscriptions at the 0.05 level. Despite statistical significance, *education* is excluded due to interpretability limitations.

Variable	Chi-Square Statistic	df	p-value	Significance
day_of_week	0.51992	4	0.9715	Not Significant
housing_updated	8.9562e-31	1	1	Not Significant
loan_updated	0.3434	1	0.5579	Not Significant
contact	77.026	1	< 2.2e-16	Significant
marital_updated	10.323	2	0.005733	Significant
pdays_category	453.02	2	< 2.2e-16	Significant
poutcome	451.96	2	< 2.2e-16	Significant
education_updated	21.558	4	0.0002453	Significant
job_updated	70.254	10	3.96e-11	Significant
month	299.32	9	< 2.2e-16	Significant

Table 3 Chi-Square Tests

2.5 Bivariate Analysis

Since call *duration* emerged as the strongest predictor, it was further analysed in relation to key variables.

Figure 15 shows that longer calls are linked to higher subscription rates, but excessive follow-ups (>5) show diminishing returns. Short calls (<250 sec) rarely lead to conversions, highlighting the need for quality engagement over quantity.

Figure 16 highlights that never-contacted clients have the widest call duration range – longer calls lead to conversions, while disinterested ones drop off quickly. Recent follow-ups are shorter and more effective, while delayed follow-ups (>1 week) see lower success rates.

Previous subscribers require less persuasion to resubscribe as they have shorter calls. Cold leads who subscribe take the longest calls, indicating new customers need more engagement. Previously unsuccessful clients who convert tend to have longer calls, implying more effort is needed to change their decision (Figure 17).

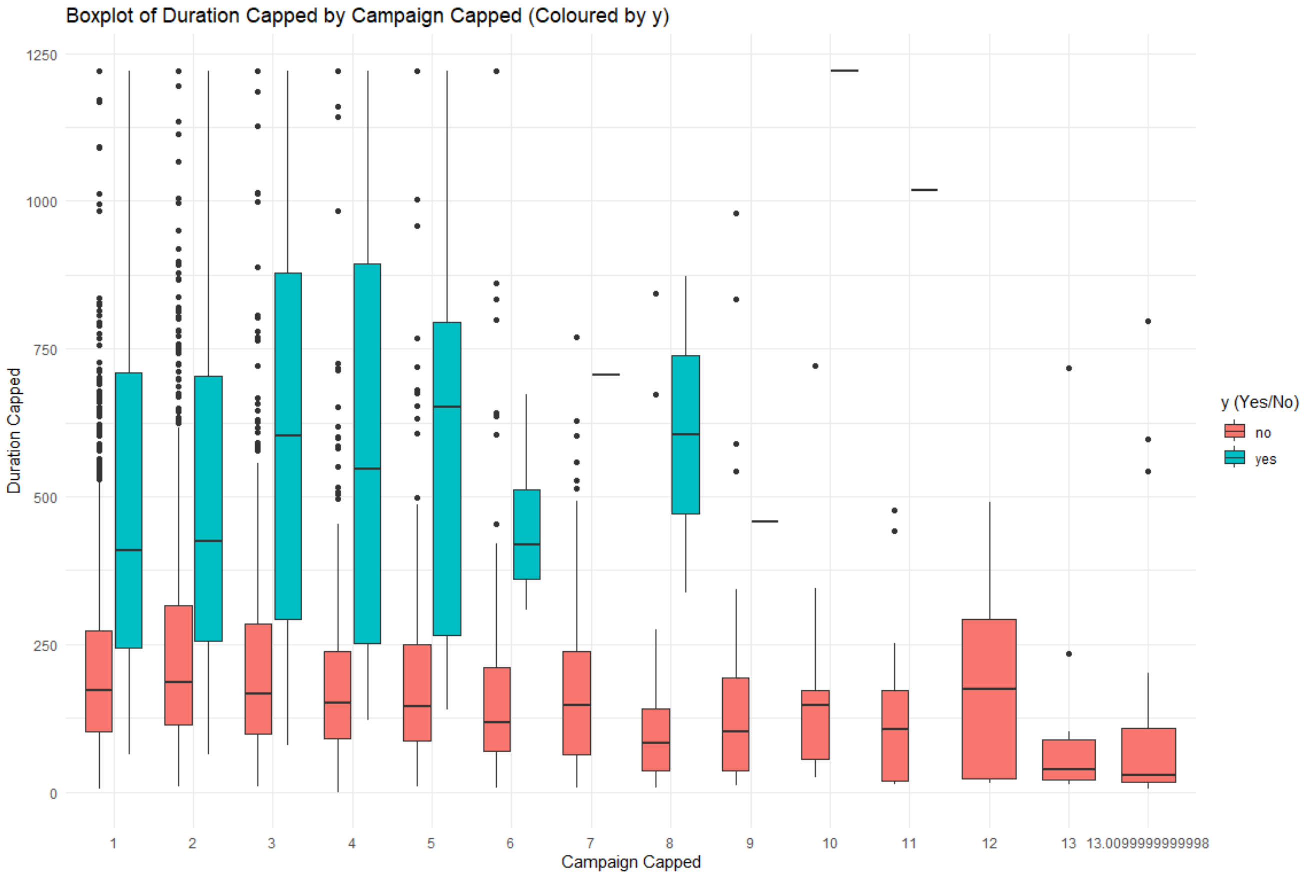

Figure 15 Boxplot of duration_capped by campaign_capped

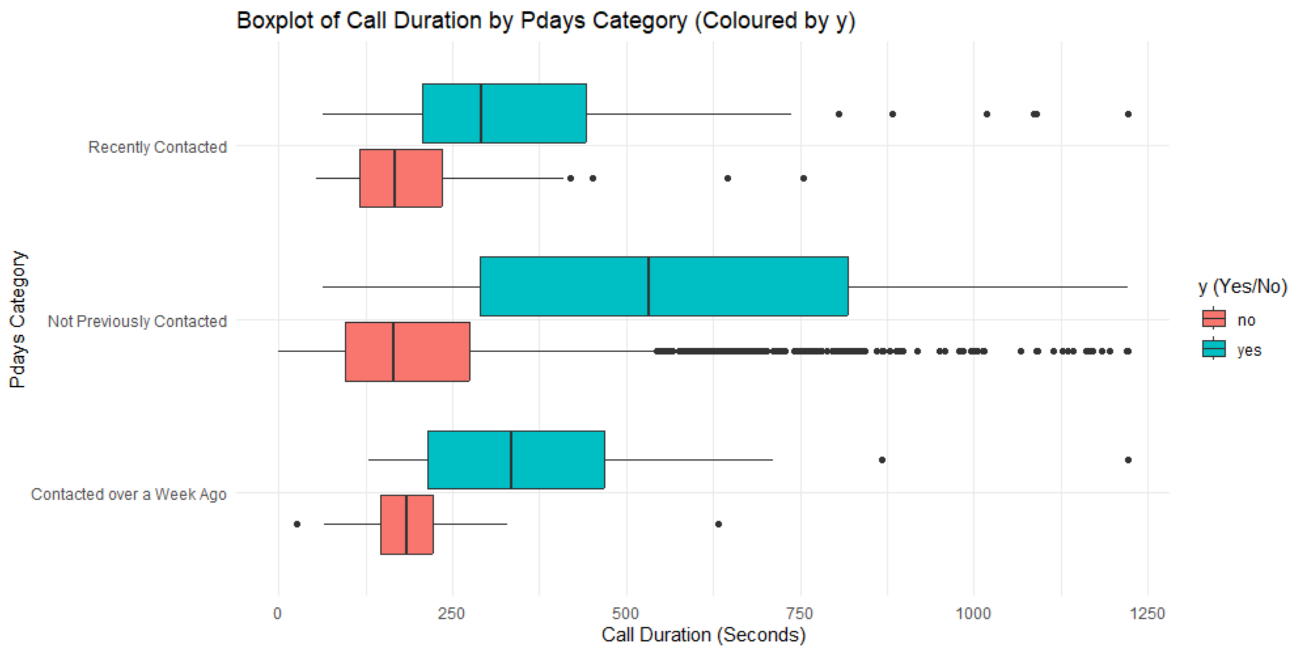

Figure 16 Boxplot of duration_capped by pdays_category

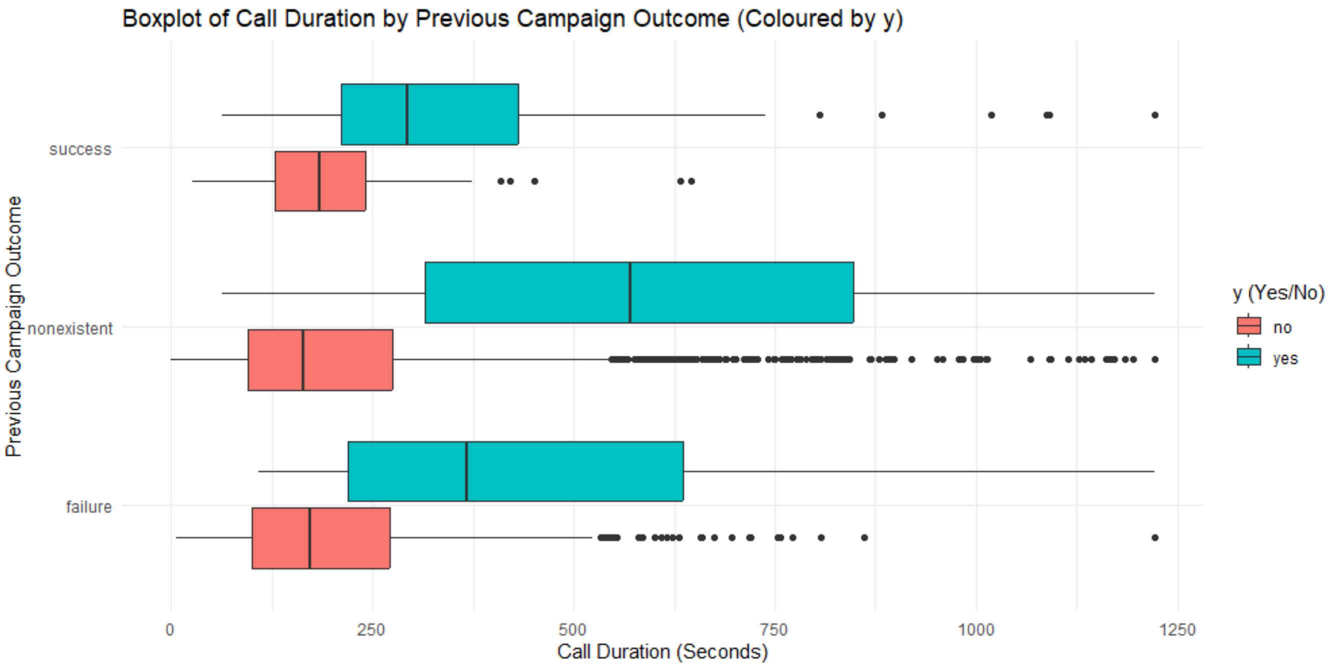

Figure 17 Boxplot of duration_capped by poutcome

2.6 Correlation Analysis

The correlation heatmap (Figure 18) confirms call *duration* as the strongest predictor, with longer calls driving higher conversions. *Previous* contacts having a moderate positive correlation hints that prior interactions improve subscription likelihood.

Age and *campaign* have weak correlations and will be excluded. Due to high multicollinearity among macroeconomic indicators, only *nr.employed*, the most correlated with *y*, will be retained to ensure model relevance and independence.

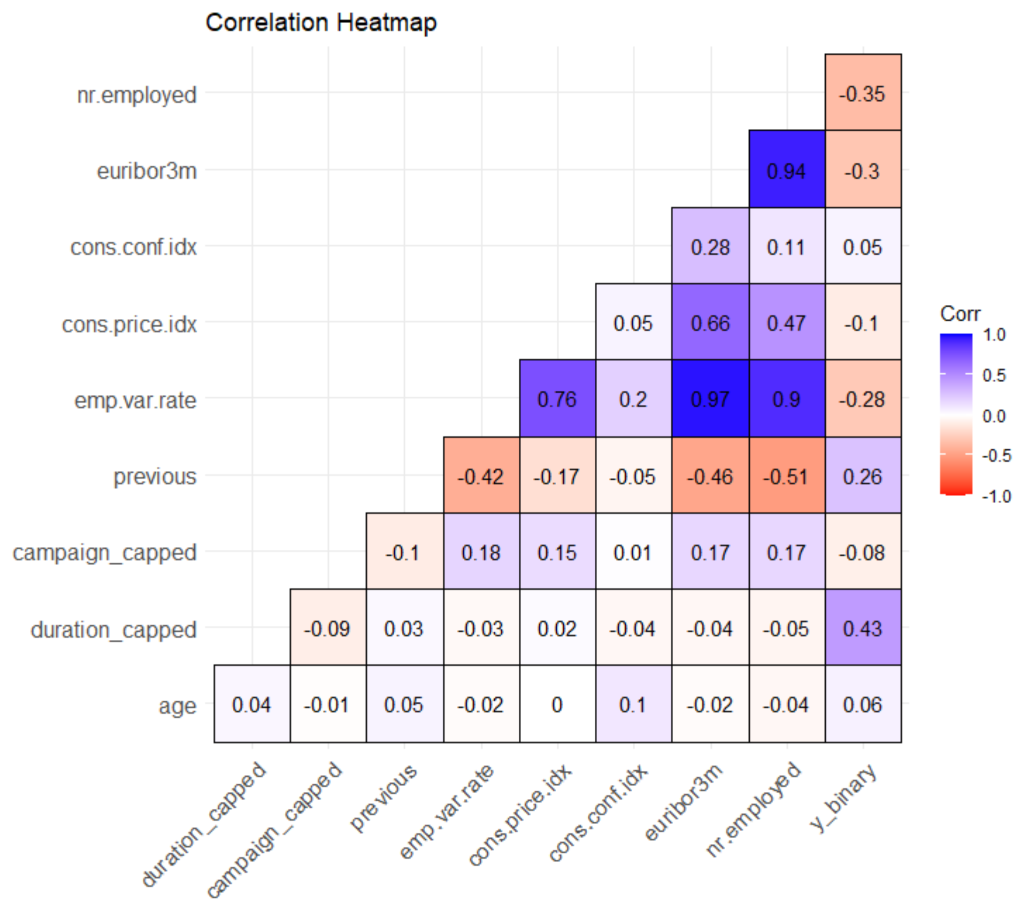

Figure 18 Correlation Heatmap

3. Model Development

3.1 Model Selection and Training

To build an effective predictive model, predictors were selected based on the EDA: *duration_capped, previous, nr.employed, job_updated, marital_updated, contact, month, poutcome,* and *pdays_category*. The dataset was then split into an 80-20 training and testing set (Figure 19).

Logistic Regression was chosen as a baseline model due to its interpretability and suitability for binary classification (Starbuck, 2023). Decision Trees were chosen for their ability to model structured decision-making while providing clear, human-interpretable forms that differentiate subscribers from non-subscribers (Quinlan, 2002). Random Forest, an ensemble method, was included to enhance accuracy by handling non-linearity and reducing overfitting through multiple decision trees, making it one of the most effective machine learning techniques (Salman, Kalakech & Steiti, 2024).

Since Logistic Regression is sensitive to scale, numerical variables were standardised to improve performance. However, Random Forest and Decision Trees are unaffected by scale, so standardisation was skipped for these models. Categorical variables were converted into factors to ensure proper handling during training.

```
> selected_vars <- updated_data %>%
+   select(duration_capped, previous, nr.employed,
+          job_updated, marital_updated, contact, month,
+          poutcome, pdays_category, y)
> selected_vars$y <- as.numeric(selected_vars$y == "yes")
> #Splitting data into training (80%) and testing (20%) sets
> set.seed(42)
> train_index <- createDataPartition(selected_vars$y, p=0.8, list=FALSE)
> train_data <- selected_vars[train_index, ]
> test_data <- selected_vars[-train_index, ]
>
> #Preprocessing numerical variables - standardisation
> standardisation <- preProcess(train_data %>%
+                                 select(duration_capped, previous,
+                                        nr.employed),
+                               method = c("center", "scale"))
>
> #Applying preprocessing to train and test sets
> train_data_scaled <- predict(standardisation, train_data)
> test_data_scaled <- predict(standardisation, test_data)
>
>
> #Converting categorical variables to factors
> train_data_scaled <- train_data_scaled %>%
+   mutate(across(where(is.character), as.factor))
>
> test_data_scaled <- test_data_scaled %>%
+   mutate(across(where(is.character), as.factor))
```

Figure 19 Splitting Data into 80-20 Train-Test

3.2 Performance and Findings

The three classification models were evaluated using accuracy, sensitivity, specificity, and AUC to assess performance. These metrics are derived from the confusion matrices (Hastie, Tibshirani and Friedman, 2009).

Logistic Regression (Figure 20) achieved the highest AUC (0.943) indicating strong predictive capability. The model exhibited an accuracy of 92.56%, high sensitivity (98.07%) but moderate specificity (50.00%). While the model performed well in detecting the majority class, its ability to differentiate minority class instances remained limited.

```
> #Logistic Regression
> log_reg <- glm(y ~ ., data = train_data_scaled, family = "binomial")
> log_reg_pred <- predict(log_reg, newdata = test_data_scaled, type = "response")
> log_reg_pred_class <- ifelse(log_reg_pred > 0.5, 1, 0)
>
> #Evaluating Logistic Regression Model
> log_reg_auc <- roc(test_data_scaled$y, log_reg_pred)
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> auc(log_reg_auc)
Area under the curve: 0.9426
> print(confusionMatrix(as.factor(log_reg_pred_class), as.factor(test_data_scaled$y)))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 712  47
         1  14  47

               Accuracy : 0.9256
                 95% CI : (0.9055, 0.9426)
    No Information Rate : 0.8854
    P-Value [Acc > NIR] : 8.511e-05

                  Kappa : 0.5674

 Mcnemar's Test P-Value : 4.182e-05

            Sensitivity : 0.9807
            Specificity : 0.5000
         Pos Pred Value : 0.9381
         Neg Pred Value : 0.7705
             Prevalence : 0.8854
         Detection Rate : 0.8683
   Detection Prevalence : 0.9256
      Balanced Accuracy : 0.7404

       'Positive' Class : 0
```

Figure 20 Evaluating the Logistic Regression Model

The model (Figure 21) confirms previous assertions that call *duration* is the strongest predictor. Moreover, employment rates are inversely proportional to subscription – hinting that financial stability reduces demand for financial products. Cellular *contact* proved more effective than landlines, and March, June, and December were the best months for conversions, while recent follow-ups increased success.

```
> #Equation of Logistic Regression Model
> #Display logistic regression model coefficients
> summary(log_reg)

Call:
glm(formula = y ~ ., family = "binomial", data = train_data_scaled)

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)    
(Intercept)                       -3.69533    0.93910  -3.935 8.32e-05 ***
duration_capped                    1.31413    0.06777  19.392  < 2e-16 ***
previous                           0.02489    0.10586   0.235 0.814144    
nr.employed                       -1.03942    0.09442 -11.008  < 2e-16 ***
job_updatedblue-collar            -0.03466    0.24594  -0.141 0.887927    
job_updatedentrepreneur           -0.48170    0.52328  -0.921 0.357291    
job_updatedhousemaid               0.33285    0.48620   0.685 0.493607    
job_updatedmanagement             -0.10525    0.31852  -0.330 0.741066    
job_updatedretired                 0.15562    0.32586   0.478 0.632949    
job_updatedself-employed          -0.58267    0.46953  -1.241 0.214617    
job_updatedservices                0.24281    0.29854   0.813 0.416035    
job_updatedstudent                -0.35320    0.43580  -0.810 0.417670    
job_updatedtechnician              0.35695    0.23064   1.548 0.121707    
job_updatedunemployed              0.66881    0.41492   1.612 0.106980    
marital_updatedmarried             0.23844    0.27604   0.864 0.387705    
marital_updatedsingle              0.31325    0.29444   1.064 0.287372    
contacttelephone                  -0.63723    0.22788  -2.796 0.005170 ** 
monthaug                           0.62919    0.33696   1.867 0.061867 .  
monthdec                           1.31477    0.66784   1.969 0.048991 *  
monthjul                           0.36283    0.34582   1.049 0.294089    
monthjun                           1.18199    0.34339   3.442 0.000577 ***
monthmar                           2.26209    0.45521   4.969 6.72e-07 ***
monthmay                          -0.55149    0.30719  -1.795 0.072611 .  
monthnov                          -0.18364    0.36341  -0.505 0.613324    
monthoct                           0.27098    0.44684   0.606 0.544221    
monthsep                          -0.20749    0.46617  -0.445 0.656255    
poutcomenonexistent                0.48607    0.34311   1.417 0.156592    
poutcomesuccess                    0.86149    0.83768   1.028 0.303750    
pdays_categoryNot Previously Contacted -0.34877    0.77971  -0.447 0.654650    
pdays_categoryRecently Contacted   0.94281    0.55403   1.702 0.088805 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2257.2  on 3279  degrees of freedom
Residual deviance: 1258.5  on 3250  degrees of freedom
AIC: 1318.5

Number of Fisher Scoring iterations: 6
```

Figure 21 Logistic Regression Model Coefficients

The Decision Tree model (Figure 22) had a lower AUC (0.73) and 91.59% accuracy, with high sensitivity (97.11%) but lower specificity (48.94%). The model's structure provided interpretability, but its performance lagged compared to logistic regression.

```
> #Decision Tree
> #Use unscaled data
> tree_model <- rpart(y ~ ., data = train_data, method = "class")
> rpart.plot(tree_model)
> tree_pred <- predict(tree_model, newdata = test_data, type = "class")
> tree_prob <- predict(tree_model, newdata = test_data, type = "prob")[,2]
>
> rpart.plot(tree_model,
+            type = 5,          #Type of plot
+            extra = 101,       #Shows probability at each node
+            under = TRUE,      #Displays samples under each node
+            tweak = 1.2,       #Adjusts text size
+            box.palette = "RdBu",  #Adds a color gradient for class separation
+            fallen.leaves = TRUE)  #Puts leaves at the bottom for clarity
>
>
>
> #Evaluating Decision Tree Model
> tree_auc <- roc(test_data$y, as.numeric(tree_pred))
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> print(auc(tree_auc))
Area under the curve: 0.7302
> print(confusionMatrix(as.factor(tree_pred), as.factor(test_data$y)))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 705  48
         1  21  46

               Accuracy : 0.9159
                 95% CI : (0.8947, 0.9339)
    No Information Rate : 0.8854
    P-Value [Acc > NIR] : 0.002671

                  Kappa : 0.5262

 Mcnemar's Test P-Value : 0.001748

            Sensitivity : 0.9711
            Specificity : 0.4894
         Pos Pred Value : 0.9363
         Neg Pred Value : 0.6866
             Prevalence : 0.8854
         Detection Rate : 0.8598
   Detection Prevalence : 0.9183
      Balanced Accuracy : 0.7302

       'Positive' Class : 0
```

Figure 22 Evaluating the Decision Tree Model

Figure 23 confirmed call *duration* as the key predictor, where longer calls (>=391s) strongly increased conversion likelihood. Past campaign success, employment rate, contact method, and timing also influenced outcomes, while short calls (<166s) and past failures predicted non-subscription.

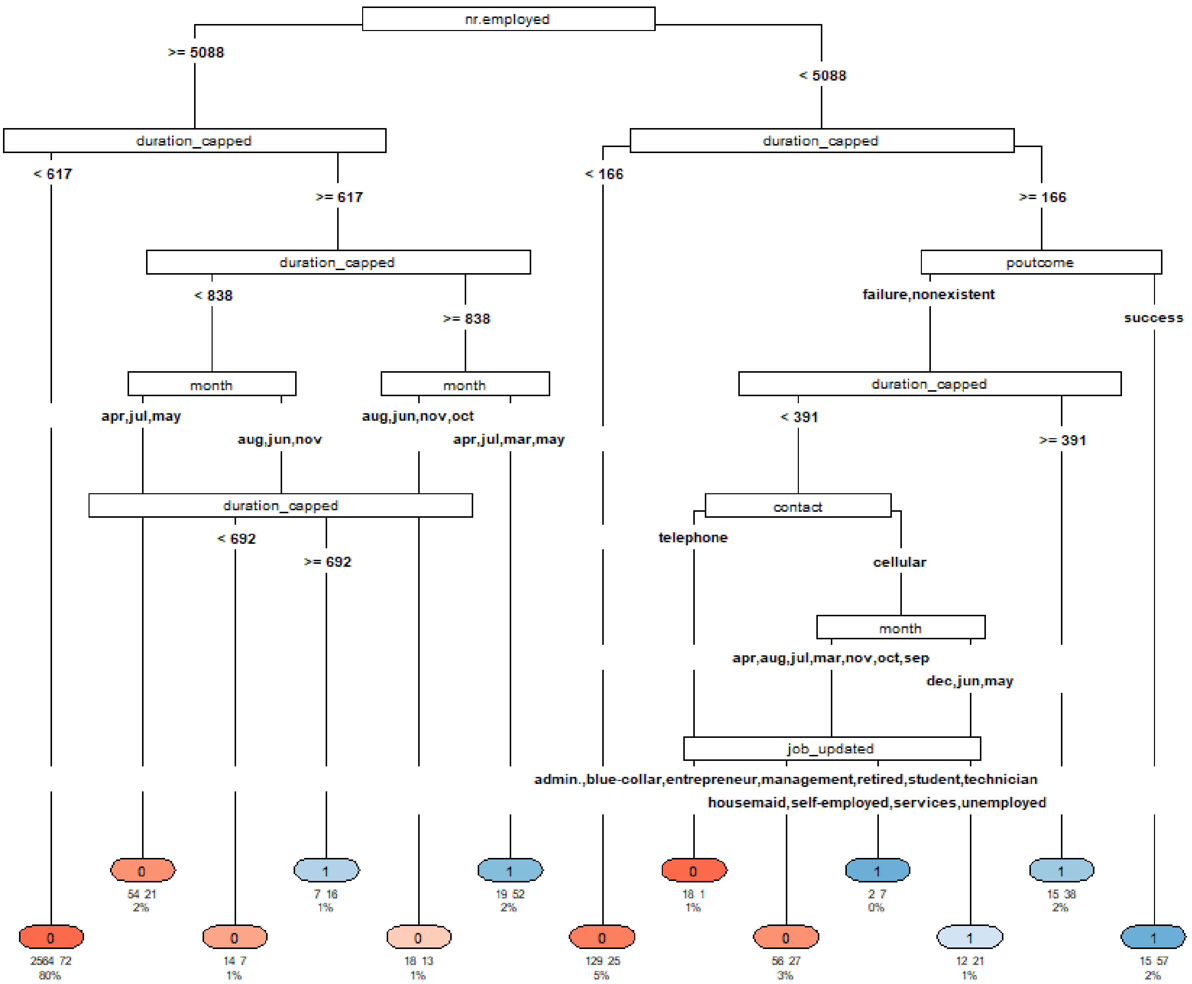

Figure 23 Decision Tree Plot

Random Forest (Figure 24) performed slightly better than the decision tree, with AUC (0.747) and 92.07% accuracy. Sensitivity remained high (97.25%), while specificity improved to 52.13%, demonstrating better generalisation.

```
> #Random Forest
> #Use unscaled data
> train_data$y <- as.factor(train_data$y)
> test_data$y <- as.factor(test_data$y)
> rf_model <- randomForest(y ~ ., data = train_data, ntree = 100, importance = TRUE)
> rf_pred <- predict(rf_model, newdata = test_data)
> rf_prob <- predict(rf_model, newdata = test_data, type = "prob")[,2]
>
> #Evaluating Random Forest Model
> rf_auc <- roc(test_data$y, as.numeric(rf_pred))
Setting levels: control = 0, case = 1
Setting direction: controls < cases
> print(auc(rf_auc))
Area under the curve: 0.7469
> print(confusionMatrix(as.factor(rf_pred), as.factor(test_data$y)))
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 706  45
         1  20  49

               Accuracy : 0.9207
                 95% CI : (0.9001, 0.9383)
    No Information Rate : 0.8854
    P-Value [Acc > NIR] : 0.000541

                  Kappa : 0.5584

 Mcnemar's Test P-Value : 0.002912

            Sensitivity : 0.9725
            Specificity : 0.5213
         Pos Pred Value : 0.9401
         Neg Pred Value : 0.7101
             Prevalence : 0.8854
         Detection Rate : 0.8610
   Detection Prevalence : 0.9159
      Balanced Accuracy : 0.7469

       'Positive' Class : 0
```

Figure 24 Evaluating the Random Forest Model

Figure 25 reaffirmed call duration, economic conditions, timing, and previous campaign success as key factors.

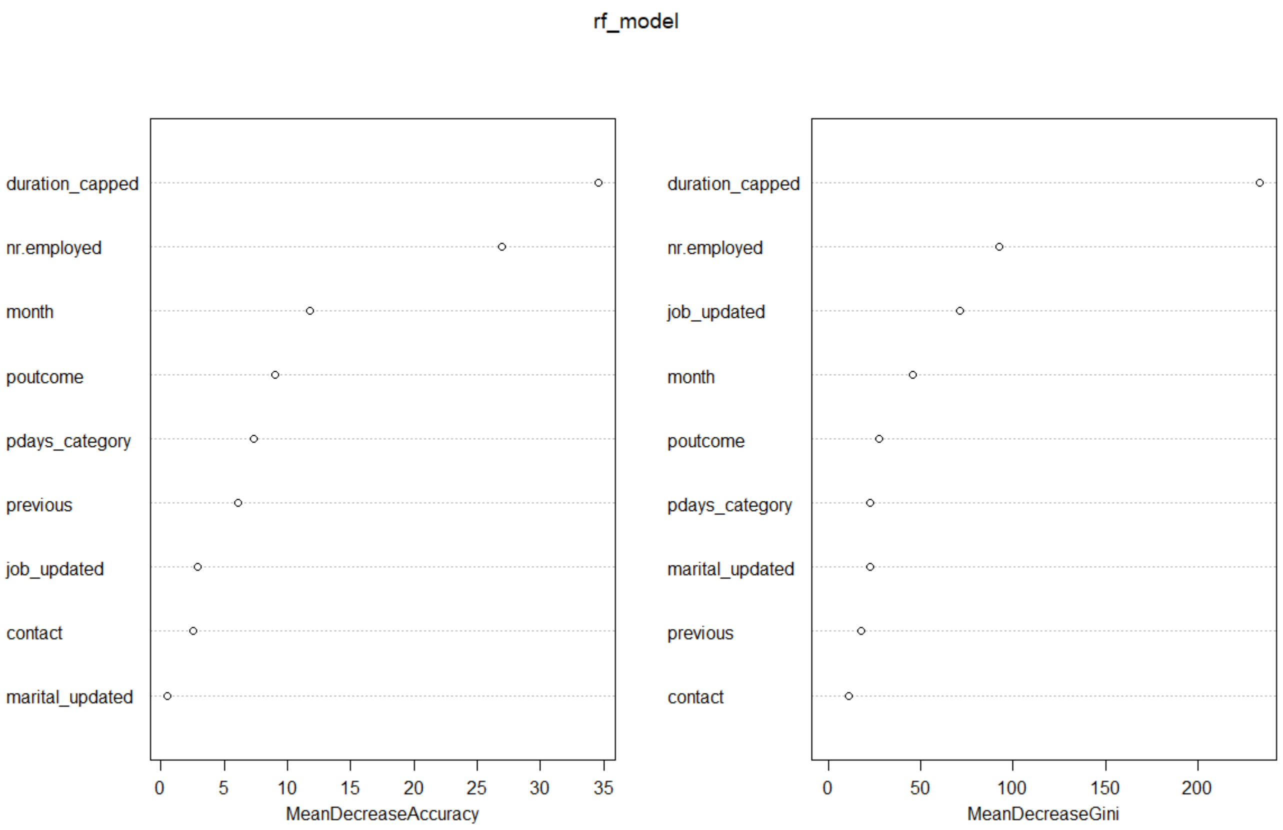

Figure 25 Random Forest Variable Importance Plot

The ROC curves (Figure 26) highlight Logistic Regression as the best classifier, followed by Random Forest and Decision Tree. While tree-based models captured non-linearity, they struggled with complex feature interactions compared to Logistic Regression, which benefited from a linear decision boundary and well-distributed class probabilities.

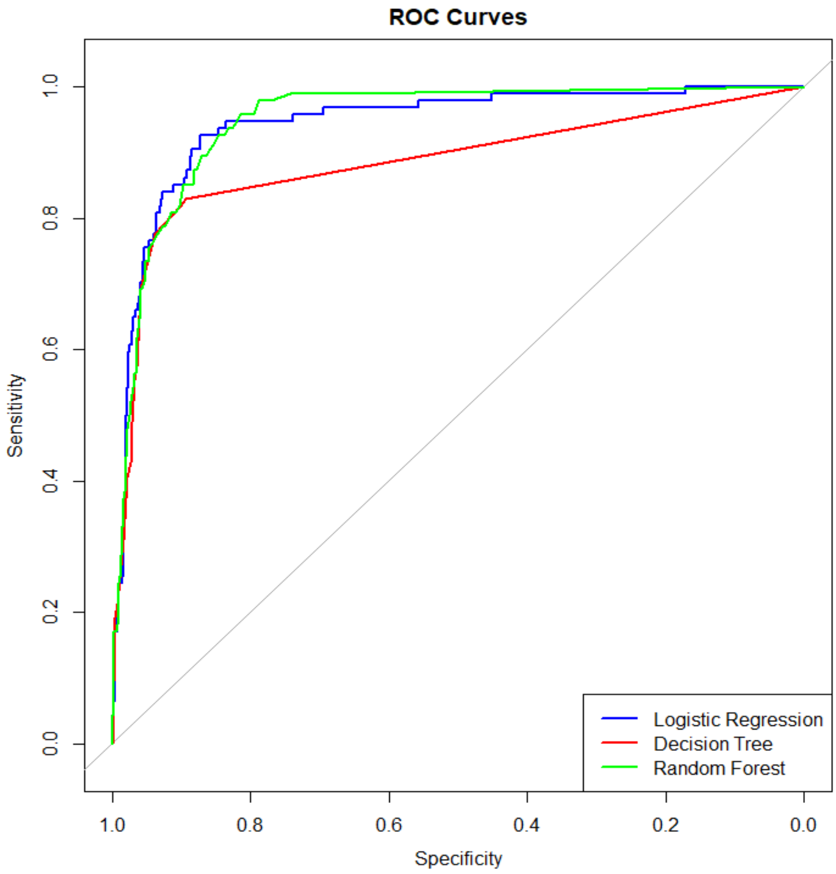

Figure 26 ROC Curves

4. Conclusion and Recommendations

The analysis revealed that call *duration* is the strongest predictor of subscription, though only available post-call. While unsuitable for pre-call predictions, it remains valuable for optimising engagement strategies and assessing campaign effectiveness.

A significant class imbalance in the dataset led to model bias towards non-subscribers. Addressing this through oversampling subscribed individuals, or class weighting would improve predictive accuracy. Ensemble methods such as XGBoost (Kavlakoglu and Russi, 2024) could enhance performance by prioritising misclassified cases, while synthetic techniques like SMOTE could generate balanced training data (Pugh, 2019).

Refining feature engineering, incorporating interaction terms, and considering time-based effects may improve predictive power. Collecting additional customer data and refining outlier treatment would strengthen future analyses, ensuring a more balanced and effective model.

5. References

Basha, R. (2024) 'A Study on the Effectiveness of Telemarketing in the Banking Industry', *Shanlax International Journal of Management*, 11(1), pp. 134-143. Available at: 10.34293/management.v11iS1-Mar.8101

Hastie, T., Tibshirani, R. and Friedman, J.H. (2009) *The Elements of Statistical Learning: Data Mining Inference, and Prediction*. 2nd edn. Springer Series in Statistics.

IBM. (no date) *What is exploratory data analysis (EDA)?* Available at: https://www.ibm.com/think/topics/exploratory-data-analysis (Accessed: 19 March 2025).

Kavlakoglu, E. and Russi, E. (2024) *What is XGBoost?* Available at: https://www.ibm.com/think/topics/xgboost (Accessed: 23 March 2025).

Moro, S., Rita, P. and Cortez, P. (2014) *Bank Marketing [Dataset]*. Available at: https://archive.ics.uci.edu/dataset/222/bank+marketing (Accessed: 15 March 2025).

Pugh, D. (2019) *Balancing Datasets and Generating Synthetic Data with SMOTE*. Available at: https://datasciencecampus.github.io/balancing-data-with-smote/ (Accessed: 23 March 2025).

Quinlan, J.R. (2002) 'Decision trees and decision-making', *IEEE Transactions on Systtems, Man, and Cybernetics*, 20(2), pp. 339-346. Available at: 10.1109/21.52545

Salman, H.A., Kalakech, A. and Steiti, A. (2024) 'Random Forest Algorithm Overview', *Babylonian Journal of Machine Learning*, pp. 69-79. Available at: 10.58496/BJML/2024/007

Starbuck, C. (2023) *The Fundamentals of People Analytics: With Applications in R*. Cham, Switzerland: Springer International Publishing. Available at: https://doi.org/10.1007/978-3-031-28674-2

Tukey, J.W. (1977) *Exploratory Data Analysis*. Reading, Massachusetts: Addison-Wesley.