# ProjectPhase3

## 1. Model: Logistic Regression

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.2
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
#Import Data
bank_data<- read.csv2("bank-full.csv")
```

```r
str(bank_data)
```

```
'data.frame':   45211 obs. of  17 variables:
 $ age      : int  58 44 33 47 33 35 28 42 58 43 ...
 $ job      : chr  "management" "technician" "entrepreneur" "blue-collar" ...
 $ marital  : chr  "married" "single" "married" "married" ...
 $ education: chr  "tertiary" "secondary" "secondary" "unknown" ...
 $ default  : chr  "no" "no" "no" "no" ...
 $ balance  : int  2143 29 2 1506 1 231 447 2 121 593 ...
 $ housing  : chr  "yes" "yes" "yes" "yes" ...
 $ loan     : chr  "no" "no" "yes" "no" ...
 $ contact  : chr  "unknown" "unknown" "unknown" "unknown" ...
 $ day      : int  5 5 5 5 5 5 5 5 5 5 ...
 $ month    : chr  "may" "may" "may" "may" ...
 $ duration : int  261 151 76 92 198 139 217 380 50 55 ...
 $ campaign : int  1 1 1 1 1 1 1 1 1 1 ...
 $ pdays    : int  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ previous : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome : chr  "unknown" "unknown" "unknown" "unknown" ...
 $ y        : chr  "no" "no" "no" "no" ...
```

```r
summary(bank_data)
```

```
     age            job              marital            education        
 Min.   :18.00   Length:45211       Length:45211       Length:45211      
 1st Qu.:33.00   Class :character   Class :character   Class :character  
 Median :39.00   Mode  :character   Mode  :character   Mode  :character  
 Mean   :40.94                                                           
 3rd Qu.:48.00                                                           
 Max.   :95.00                                                           
   default             balance          housing              loan          
 Length:45211       Min.   : -8019   Length:45211       Length:45211      
 Class :character   1st Qu.:    72   Class :character   Class :character  
 Mode  :character   Median :   448   Mode  :character   Mode  :character  
                    Mean   :  1362                                        
                    3rd Qu.:  1428                                        
                    Max.   :102127                                        
   contact               day           month              duration     
 Length:45211       Min.   : 1.00   Length:45211       Min.   :   0.0  
 Class :character   1st Qu.: 8.00   Class :character   1st Qu.: 103.0  
 Mode  :character   Median :16.00   Mode  :character   Median : 180.0  
                    Mean   :15.81                      Mean   : 258.2  
                    3rd Qu.:21.00                      3rd Qu.: 319.0  
                    Max.   :31.00                      Max.   :4918.0  
    campaign          pdays          previous           poutcome        
 Min.   : 1.000   Min.   : -1.0   Min.   :  0.0000   Length:45211      
 1st Qu.: 1.000   1st Qu.: -1.0   1st Qu.:  0.0000   Class :character  
 Median : 2.000   Median : -1.0   Median :  0.0000   Mode  :character  
 Mean   : 2.764   Mean   : 40.2   Mean   :  0.5803                     
 3rd Qu.: 3.000   3rd Qu.: -1.0   3rd Qu.:  0.0000                     
 Max.   :63.000   Max.   :871.0   Max.   :275.0000                     
      y            
 Length:45211      
 Class :character
 Mode  :character
```

```
# set the target as factor "y"
bank_data$y <- factor(bank_data$y, levels = c("no", "yes"))
```

```
# Remove leakage variable
bank_data$duration <- NULL

# Fix pdays based on your dataset ( -1 = not contacted )
bank_data$previous_contact <- ifelse(bank_data$pdays == -1, "no", "yes")
bank_data$previous_contact <- factor(bank_data$previous_contact)
```

```
#Convert categorical variables to factors
factor_vars <- c("job","marital","education","default","housing",
                 "loan","contact","month","poutcome","previous_contact")

bank_data[factor_vars] <- lapply(bank_data[factor_vars], factor)
```

```
# Train-test split
set.seed(123)  # ensures reproducibility

n <- nrow(bank_data)
train_index <- sample(1:n, size = 0.7 * n)

train <- bank_data[train_index, ]
test  <- bank_data[-train_index, ]

# Check dimensions
dim(train)
```

```
[1] 31647    17
```

```
dim(test)
```

```
[1] 13564    17
```

```r
prop.table(table(train$y))
```

```
       no       yes 
0.8823585 0.1176415 
```

```r
prop.table(table(test$y))
```

```
       no       yes 
0.8845473 0.1154527 
```

```r
#Logistic Regression

full_model <- glm(
  y ~ age + job + marital + education + default + housing +
      loan + contact + month + campaign + previous + previous_contact + poutcome + balance,
  data = train,
  family = binomial
)

summary(full_model)
```

```

Call:
glm(formula = y ~ age + job + marital + education + default + 
    housing + loan + contact + month + campaign + previous + 
    previous_contact + poutcome + balance, family = binomial, 
    data = train)

Coefficients:
                      Estimate Std. Error z value Pr(>|z|)    
(Intercept)         -3.298e+00  1.449e+00  -2.276 0.022854 *  
age                 -1.185e-03  2.343e-03  -0.506 0.613074    
jobblue-collar      -1.128e-01  7.708e-02  -1.464 0.143240    
jobentrepreneur     -1.057e-01  1.304e-01  -0.810 0.417764    
jobhousemaid        -2.950e-01  1.418e-01  -2.080 0.037567 *  
jobmanagement       -4.392e-02  7.842e-02  -0.560 0.575409    
jobretired           4.246e-01  1.044e-01   4.068 4.75e-05 ***
jobself-employed    -1.038e-01  1.176e-01  -0.882 0.377550    
```

```
jobservices         -1.210e-01  8.922e-02  -1.356 0.175031    
jobstudent           2.646e-01  1.201e-01   2.203 0.027625 *  
jobtechnician       -8.526e-02  7.367e-02  -1.157 0.247140    
jobunemployed        1.559e-01  1.152e-01   1.353 0.176137    
jobunknown          -3.275e-01  2.656e-01  -1.233 0.217525    
maritalmarried      -2.257e-01  6.154e-02  -3.667 0.000245 ***
maritalsingle        5.682e-02  7.045e-02   0.807 0.419899    
educationsecondary   1.376e-01  6.764e-02   2.034 0.041986 *  
educationtertiary    2.821e-01  7.888e-02   3.577 0.000348 ***
educationunknown     1.459e-01  1.116e-01   1.307 0.191130    
defaultyes          -1.520e-01  1.744e-01  -0.872 0.383229    
housingyes          -5.302e-01  4.559e-02 -11.629  < 2e-16 ***
loanyes             -3.765e-01  6.293e-02  -5.983 2.19e-09 ***
contacttelephone    -2.558e-01  7.879e-02  -3.247 0.001166 ** 
contactunknown      -1.398e+00  7.579e-02 -18.442  < 2e-16 ***
monthaug            -9.783e-01  8.237e-02 -11.877  < 2e-16 ***
monthdec             5.188e-01  1.930e-01   2.688 0.007185 ** 
monthfeb            -4.799e-01  8.875e-02  -5.407 6.41e-08 ***
monthjan            -1.147e+00  1.253e-01  -9.150  < 2e-16 ***
monthjul            -7.504e-01  7.960e-02  -9.427  < 2e-16 ***
monthjun             9.130e-02  9.565e-02   0.955 0.339824    
monthmar             9.311e-01  1.343e-01   6.935 4.06e-12 ***
monthmay            -5.415e-01  7.461e-02  -7.257 3.94e-13 ***
monthnov            -9.921e-01  8.887e-02 -11.163  < 2e-16 ***
monthoct             5.305e-01  1.168e-01   4.542 5.56e-06 ***
monthsep             6.696e-01  1.269e-01   5.278 1.30e-07 ***
campaign            -7.487e-02  9.709e-03  -7.711 1.25e-14 ***
previous             8.086e-03  6.215e-03   1.301 0.193279    
previous_contactyes  2.239e+00  1.442e+00   1.553 0.120402    
poutcomeother        3.049e-01  9.492e-02   3.212 0.001318 ** 
poutcomesuccess      2.276e+00  8.787e-02  25.900  < 2e-16 ***
poutcomeunknown      2.375e+00  1.443e+00   1.646 0.099776 .  
balance              2.558e-05  5.476e-06   4.670 3.01e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22925  on 31646  degrees of freedom
Residual deviance: 19071  on 31606  degrees of freedom
AIC: 19153

Number of Fisher Scoring iterations: 6
```

```r
step_model <- step(full_model, direction = "both", trace = FALSE)
summary(step_model)
```

```
Call:
glm(formula = y ~ job + marital + education + housing + loan + 
    contact + month + campaign + previous_contact + poutcome + 
    balance, family = binomial, data = train)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -3.350e+00  1.444e+00  -2.320 0.020357 *  
jobblue-collar     -1.134e-01  7.702e-02  -1.473 0.140808    
jobentrepreneur    -1.108e-01  1.303e-01  -0.850 0.395382    
jobhousemaid       -3.014e-01  1.415e-01  -2.130 0.033177 *  
jobmanagement      -4.628e-02  7.832e-02  -0.591 0.554561    
jobretired          4.004e-01  9.424e-02   4.249 2.14e-05 ***
jobself-employed   -1.073e-01  1.175e-01  -0.913 0.361377    
jobservices        -1.209e-01  8.918e-02  -1.356 0.175202    
jobstudent          2.762e-01  1.181e-01   2.339 0.019322 *  
jobtechnician      -8.580e-02  7.366e-02  -1.165 0.244053    
jobunemployed       1.538e-01  1.152e-01   1.336 0.181685    
jobunknown         -3.333e-01  2.655e-01  -1.255 0.209344    
maritalmarried     -2.217e-01  6.131e-02  -3.616 0.000299 ***
maritalsingle       6.935e-02  6.607e-02   1.050 0.293862    
educationsecondary  1.404e-01  6.724e-02   2.089 0.036747 *  
educationtertiary   2.878e-01  7.817e-02   3.681 0.000232 ***
educationunknown    1.444e-01  1.116e-01   1.294 0.195619    
housingyes         -5.266e-01  4.538e-02 -11.605  < 2e-16 ***
loanyes            -3.795e-01  6.279e-02  -6.045 1.50e-09 ***
contacttelephone   -2.595e-01  7.805e-02  -3.325 0.000883 ***
contactunknown     -1.400e+00  7.576e-02 -18.481  < 2e-16 ***
monthaug           -9.796e-01  8.231e-02 -11.902  < 2e-16 ***
monthdec            5.206e-01  1.930e-01   2.698 0.006984 ** 
monthfeb           -4.782e-01  8.868e-02  -5.393 6.93e-08 ***
monthjan           -1.146e+00  1.254e-01  -9.141  < 2e-16 ***
monthjul           -7.516e-01  7.954e-02  -9.450  < 2e-16 ***
monthjun            9.272e-02  9.565e-02   0.969 0.332360    
monthmar            9.316e-01  1.342e-01   6.941 3.90e-12 ***
monthmay           -5.405e-01  7.457e-02  -7.248 4.24e-13 ***
monthnov           -9.944e-01  8.880e-02 -11.198  < 2e-16 ***
monthoct            5.306e-01  1.167e-01   4.546 5.46e-06 ***
```

```
monthsep              6.710e-01  1.268e-01   5.291 1.22e-07 ***
campaign             -7.458e-02  9.696e-03  -7.692 1.45e-14 ***
previous_contactyes   2.255e+00  1.441e+00   1.565 0.117619    
poutcomeother         3.143e-01  9.459e-02   3.323 0.000890 ***
poutcomesuccess       2.278e+00  8.785e-02  25.930  < 2e-16 ***
poutcomeunknown       2.367e+00  1.442e+00   1.642 0.100657    
balance               2.558e-05  5.452e-06   4.692 2.70e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22925  on 31646  degrees of freedom
Residual deviance: 19074  on 31609  degrees of freedom
AIC: 19150

Number of Fisher Scoring iterations: 6
```

```
library(car)
```

```
Loading required package: carData
```

```
Attaching package: 'car'
```

```
The following object is masked from 'package:dplyr':

    recode
```

```
The following object is masked from 'package:purrr':

    some
```

```
vif(step_model)
```

```
                        GVIF Df GVIF^(1/(2*Df))
job                 2.887692 11        1.049383
marital             1.168093  2        1.039607
education           2.219748  3        1.142135
housing             1.369677  1        1.170332
```

```
loan                  1.057187  1        1.028196
contact               1.801797  2        1.158581
month                 2.703957 11        1.046252
campaign              1.069901  1        1.034360
previous_contact 1111.354867  1       33.336989
poutcome         1149.581011  3        3.236607
balance               1.036594  1        1.018133
```

```
final_model <- glm(
  y ~ job + marital + education + housing + loan + contact +
      month + campaign + poutcome + balance,
  data = train,
  family = binomial
)

summary(final_model)
```

```

Call:
glm(formula = y ~ job + marital + education + housing + loan +
    contact + month + campaign + poutcome + balance, family = binomial,
    data = train)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.098e+00  1.278e-01  -8.587  < 2e-16 ***
jobblue-collar     -1.133e-01  7.702e-02  -1.471 0.141258
jobentrepreneur    -1.109e-01  1.303e-01  -0.851 0.394807
jobhousemaid       -3.009e-01  1.415e-01  -2.126 0.033482 *
jobmanagement      -4.639e-02  7.832e-02  -0.592 0.553642
jobretired          4.029e-01  9.422e-02   4.276 1.90e-05 ***
jobself-employed   -1.072e-01  1.175e-01  -0.912 0.361683
jobservices        -1.209e-01  8.918e-02  -1.356 0.175082
jobstudent          2.765e-01  1.181e-01   2.342 0.019166 *
jobtechnician      -8.491e-02  7.365e-02  -1.153 0.248938
jobunemployed       1.543e-01  1.152e-01   1.339 0.180475
jobunknown         -3.327e-01  2.655e-01  -1.253 0.210118
maritalmarried     -2.214e-01  6.131e-02  -3.611 0.000305 ***
maritalsingle       7.014e-02  6.607e-02   1.062 0.288412
educationsecondary  1.408e-01  6.724e-02   2.094 0.036259 *
educationtertiary   2.884e-01  7.817e-02   3.689 0.000225 ***
educationunknown    1.448e-01  1.116e-01   1.298 0.194396
```

```
housingyes         -5.252e-01  4.537e-02 -11.576  < 2e-16 ***
loanyes            -3.766e-01  6.272e-02  -6.006 1.91e-09 ***
contacttelephone   -2.599e-01  7.805e-02  -3.329 0.000870 ***
contactunknown     -1.402e+00  7.576e-02 -18.504  < 2e-16 ***
monthaug           -9.784e-01  8.230e-02 -11.889  < 2e-16 ***
monthdec            5.211e-01  1.930e-01   2.700 0.006924 **
monthfeb           -4.780e-01  8.868e-02  -5.390 7.04e-08 ***
monthjan           -1.146e+00  1.254e-01  -9.138  < 2e-16 ***
monthjul           -7.519e-01  7.953e-02  -9.454  < 2e-16 ***
monthjun            9.579e-02  9.563e-02   1.002 0.316519
monthmar            9.320e-01  1.342e-01   6.944 3.80e-12 ***
monthmay           -5.402e-01  7.457e-02  -7.245 4.33e-13 ***
monthnov           -9.943e-01  8.880e-02 -11.197  < 2e-16 ***
monthoct            5.310e-01  1.167e-01   4.550 5.36e-06 ***
monthsep            6.716e-01  1.268e-01   5.296 1.18e-07 ***
campaign           -7.451e-02  9.692e-03  -7.688 1.49e-14 ***
poutcomeother       3.144e-01  9.459e-02   3.324 0.000888 ***
poutcomesuccess     2.278e+00  8.784e-02  25.932  < 2e-16 ***
poutcomeunknown     1.126e-01  6.161e-02   1.829 0.067470 .
balance             2.560e-05  5.452e-06   4.695 2.67e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 22925  on 31646  degrees of freedom
Residual deviance: 19076  on 31610  degrees of freedom
AIC: 19150

Number of Fisher Scoring iterations: 6
```

```
vif(final_model)
```

```
              GVIF Df GVIF^(1/(2*Df))
job       2.885854 11        1.049353
marital   1.167934  2        1.039572
education 2.219891  3        1.142147
housing   1.369114  1        1.170091
loan      1.055712  1        1.027479
contact   1.801943  2        1.158605
month     2.702727 11        1.046230
campaign  1.069844  1        1.034332
```

```
poutcome  1.216169  3        1.033155
balance   1.036596  1        1.018133
```

```r
test$prob <- predict(final_model, newdata = test, type = "response")
test$pred <- ifelse(test$prob > 0.5, "yes", "no")

#Confusion Metrics
table(Predicted = test$pred, Actual = test$y)
```

```
         Actual
Predicted    no   yes
      no  11842  1282
      yes   156   284
```

```r
#Accuracy
mean(test$pred == test$y)
```

```
[1] 0.8939841
```

```r
# ROC & AUC
library(pROC)
```

```
Type 'citation("pROC")' for a citation.


Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var
```

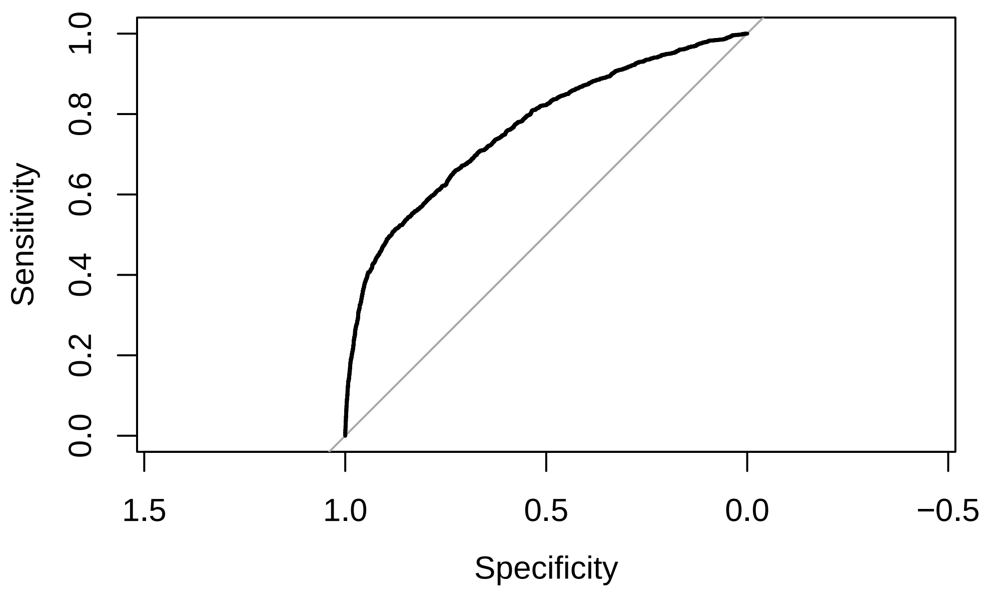
```r
roc_obj <- roc(test$y, test$prob)
```

```
Setting levels: control = no, case = yes

Setting direction: controls < cases
```

```r
auc(roc_obj)
```

```
Area under the curve: 0.7644
```

```r
plot(roc_obj)
```



```r
#Try different cutoff

thresholds <- seq(0.1, 0.5, by = 0.05)

for (t in thresholds) {
  pred <- ifelse(test$prob > t, "yes", "no")
  cm <- table(Predicted = pred, Actual = test$y)
  sensitivity <- cm["yes","yes"] / (cm["yes","yes"] + cm["no","yes"])
  specificity <- cm["no","no"] / (cm["no","no"] + cm["yes","no"])
  cat("\nThreshold:", t,
      "Sensitivity:", round(sensitivity,3),
      "Specificity:", round(specificity,3))
}
```

```
Threshold: 0.1 Sensitivity: 0.71 Specificity: 0.659
Threshold: 0.15 Sensitivity: 0.522 Specificity: 0.865
Threshold: 0.2 Sensitivity: 0.434 Specificity: 0.926
Threshold: 0.25 Sensitivity: 0.383 Specificity: 0.95
Threshold: 0.3 Sensitivity: 0.335 Specificity: 0.96
Threshold: 0.35 Sensitivity: 0.29 Specificity: 0.968
Threshold: 0.4 Sensitivity: 0.255 Specificity: 0.976
Threshold: 0.45 Sensitivity: 0.213 Specificity: 0.981
Threshold: 0.5 Sensitivity: 0.181 Specificity: 0.987
```

```r
coords(roc_obj, "best", ret="threshold", best.method="youden")
```

```
  threshold
1 0.1550712
```

**We selected a 0.155 threshold**

The default 0.5 threshold produced very low sensitivity because only a small proportion of customers subscribe, causing the model to classify most cases as ‘no.’ To avoid losing high-potential customers, we applied Youden’s Index to identify the optimal decision threshold. The best cutoff was 0.155, which provides the strongest balance between sensitivity and specificity. Lowering the threshold allows the model to capture substantially more potential subscribers while maintaining acceptable precision—making it a more effective strategy for targeted marketing.

```r
# Best threshold from Youden Index
best_t <- 0.1550712   # or use: coords(roc_obj, "best", ret="threshold", best.method="youden"

# Predict using the new threshold
test$pred_best <- ifelse(test$prob > best_t, "yes", "no")

# Confusion matrix
cm_best <- table(Predicted = test$pred_best, Actual = test$y)
cm_best
```

```
         Actual
Predicted    no   yes
      no  10509   762
      yes  1489   804
```

```
# Accuracy
accuracy_best <- mean(test$pred_best == test$y)
accuracy_best
```

```
[1] 0.834046
```

```
# Sensitivity (Recall for "yes")
sensitivity_best <- cm_best["yes", "yes"] /
                    (cm_best["yes", "yes"] + cm_best["no", "yes"])
sensitivity_best
```

```
[1] 0.51341
```

```
# Specificity (Recall for "no")
specificity_best <- cm_best["no", "no"] /
                    (cm_best["no", "no"] + cm_best["yes", "no"])
specificity_best
```

```
[1] 0.875896
```

```
# Optional: Print everything clearly
cat("\nThreshold:", best_t,
    "\nAccuracy:", round(accuracy_best,4),
    "\nSensitivity:", round(sensitivity_best,4),
    "\nSpecificity:", round(specificity_best,4), "\n")
```

```
Threshold: 0.1550712
Accuracy: 0.834
Sensitivity: 0.5134
Specificity: 0.8759
```

## 1. Model: Logistic Regression (Summary)

### Data Cleaning & Preparation

We converted categorical variables into factors and engineered the `previous_contact` variable to reflect prior outreach more clearly. This ensured that the logistic regression model could interpret the predictors correctly without structural issues.

### Logistic Regression Model

The full model allowed us to assess the initial influence of all predictors, but it also revealed several insignificant or overlapping variables, so we used AIC-based stepwise selection to remove redundant or non-informative variables while retaining predictors that meaningfully improved overall model fit. This produced a more stable and interpretable model.

### Key Predictors Identified

After stepwise refinement, several predictors remained highly influential. Positive drivers included higher education levels (secondary and tertiary), retired or student job status, successful prior campaign outcomes, and higher account balance. Negative influences included holding housing or personal loans, contacting customers via telephone or unknown channels, and certain campaign months such as July, August, and November. These directional effects provide an initial understanding of the customer segments more likely or less likely to respond.

### Model Evaluation (ROC & AUC)

The logistic model achieved an **AUC** of approximately **0.76,** indicating solid baseline discrimination despite the dataset's class imbalance.

### Threshold Optimization (Youden Index)

Because the default 0.50 threshold missed many potential subscribers, we applied Youden's Index to find the optimal cutoff. A threshold of **0.155** provided the best balance between sensitivity and specificity, making the model more effective for identifying potential 'yes' cases.

## 2. Random Forest

```r
library(randomForest)
```

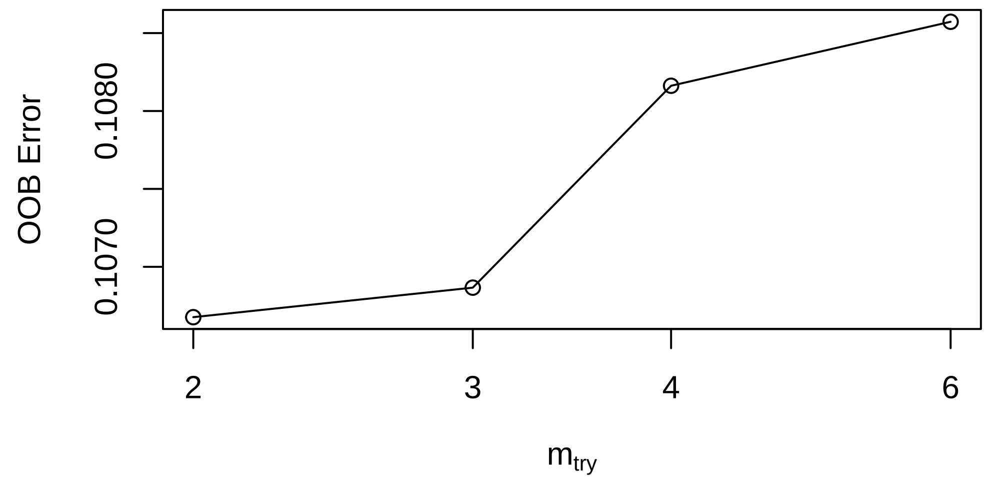
```
randomForest 4.7-1.2

Type rfNews() to see new features/changes/bug fixes.


Attaching package: 'randomForest'
```

```
The following object is masked from 'package:dplyr':

    combine


The following object is masked from 'package:ggplot2':

    margin
```

```
set.seed(123)

x_train <- subset(train, select = -y)
y_train <- train$y
```

```
set.seed(123)

tune_res <- tuneRF(
  x = x_train,
  y = y_train,
  stepFactor = 1.5,
  improve    = 0.005,
  ntreeTry   = 300,
  trace      = TRUE,
  plot       = TRUE   )
```

```
mtry = 4  OOB error = 10.82%
Searching left ...
mtry = 3    OOB error = 10.69%
0.0119778 0.005
mtry = 2    OOB error = 10.67%
0.001774098 0.005
Searching right ...
mtry = 6    OOB error = 10.86%
-0.01596688 0.005
```



```r
set.seed(123)

rf_model <- randomForest(
  y ~ .,
  data = train,
  ntree = 500,
  mtry  = 2,
  importance = TRUE
)

rf_model
```

```

Call:
 randomForest(formula = y ~ ., data = train, ntree = 500, mtry = 2,      importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 10.68%
Confusion matrix:
       no yes class.error
no  27626 298  0.01067182
yes  3081 642  0.82755842
```

```
rf_prob <- predict(rf_model, newdata=test, type="prob")[,2]
library(pROC)
roc_rf <- roc(test$y, rf_prob)
```

```
Setting levels: control = no, case = yes

Setting direction: controls < cases
```

```
coords(roc_rf, "best", ret="threshold")
```

```
  threshold
1     0.057
```

```
library(pROC)
rf_prob <- predict(rf_model, newdata=test, type="prob")[,2]
roc_rf <- roc(test$y, rf_prob)
```

```
Setting levels: control = no, case = yes

Setting direction: controls < cases
```

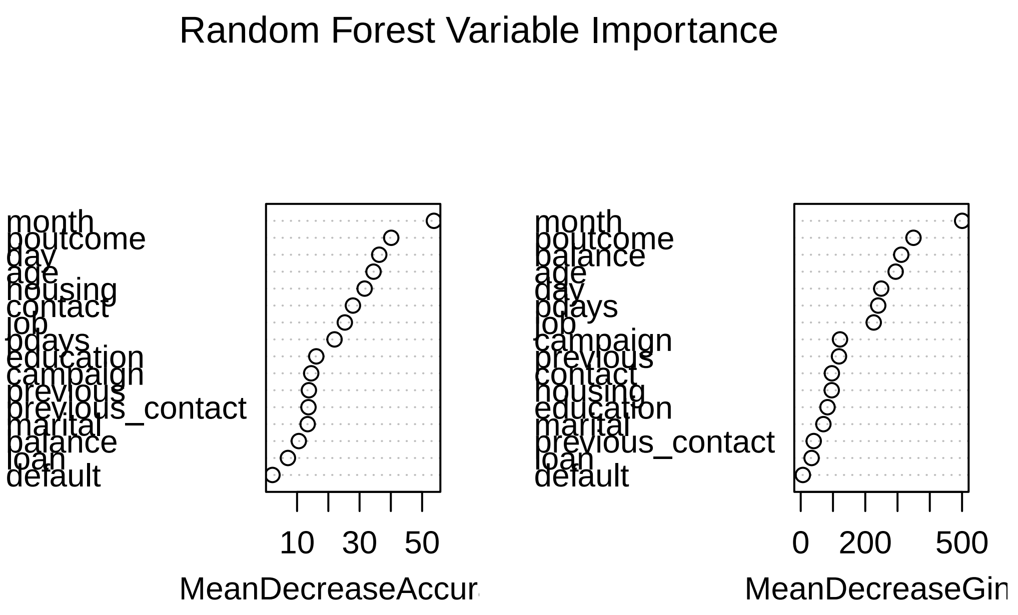
```
best_t <- as.numeric(coords(roc_rf, "best", ret="threshold"))

rf_pred2 <- factor(ifelse(rf_prob > best_t, "yes", "no"),
                   levels = c("no","yes"))

table(Predicted = rf_pred2, Actual = test$y)
```

```
         Actual
Predicted    no   yes
      no  10099   601
      yes  1899   965
```

```r
importance(rf_model)
```

```
                        no       yes MeanDecreaseAccuracy MeanDecreaseGini
age              31.534066  7.072938            34.464131        294.19204
job              26.859409 -5.871211            25.279882        226.24705
marital          10.757869  5.798110            13.386577         70.18623
education        17.148381 -1.599597            16.146582         83.02434
default           1.006874  3.094812             2.132085          6.64752
balance           7.209322  6.992791            10.557528        311.56701
housing          28.547696  8.445231            31.614042         96.11769
loan             -3.318283 16.033670             7.077996         33.74744
contact          25.621392 12.582831            27.872002         96.54402
day              36.246415 -2.960642            36.284896        249.48974
month            51.445738 12.709346            53.772727        500.32924
campaign          9.622106 11.863759            14.524867        121.75171
pdays            20.227198 14.981491            21.984929        240.12281
previous         12.950615 11.588766            13.780203        118.59273
poutcome         31.941247  2.695573            40.115277        349.41530
previous_contact 13.049454  9.885696            13.646109         40.26791
```

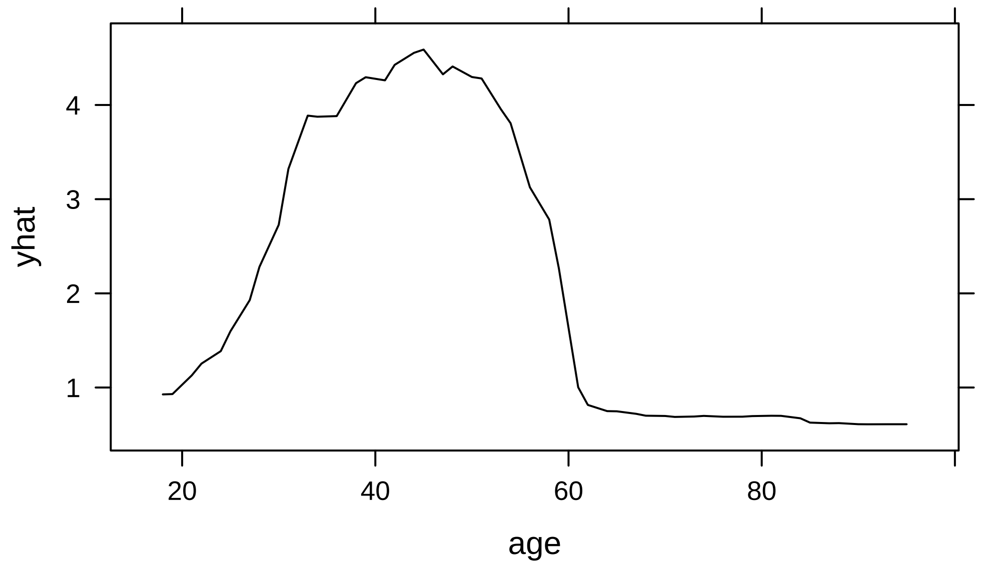
```r
varImpPlot(rf_model,
           sort = TRUE,
           n.var = min(20, ncol(train)-1),
           main = "Random Forest Variable Importance")
```



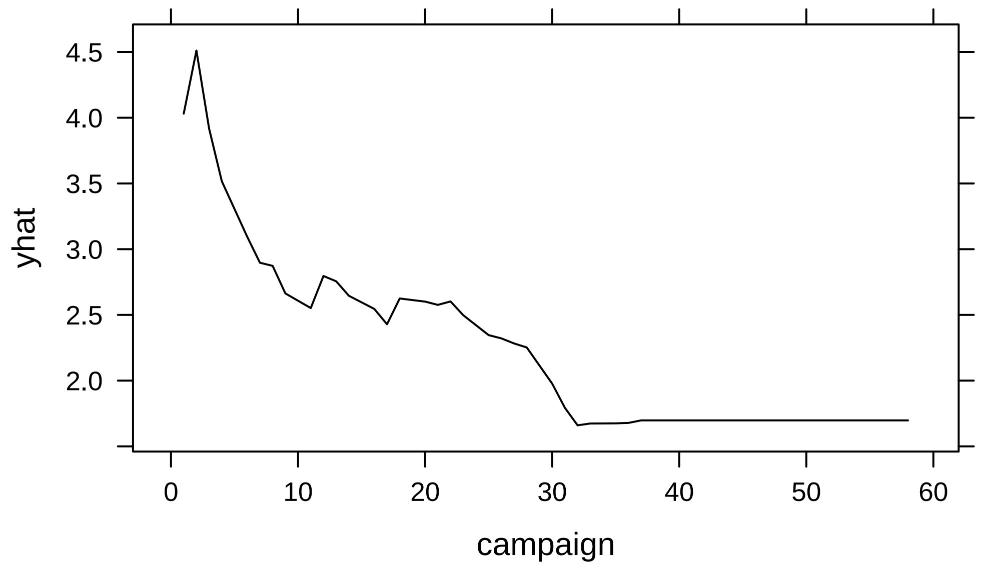
```r
library(pdp)
```

```

Attaching package: 'pdp'

The following object is masked from 'package:purrr':

    partial
```

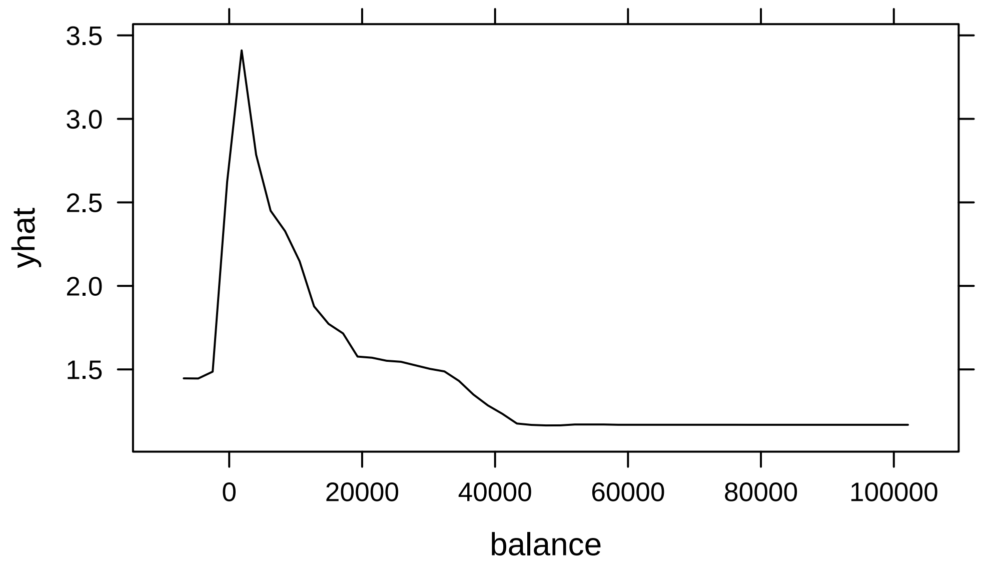
```r
partial(rf_model, pred.var = "age", plot = TRUE)
```



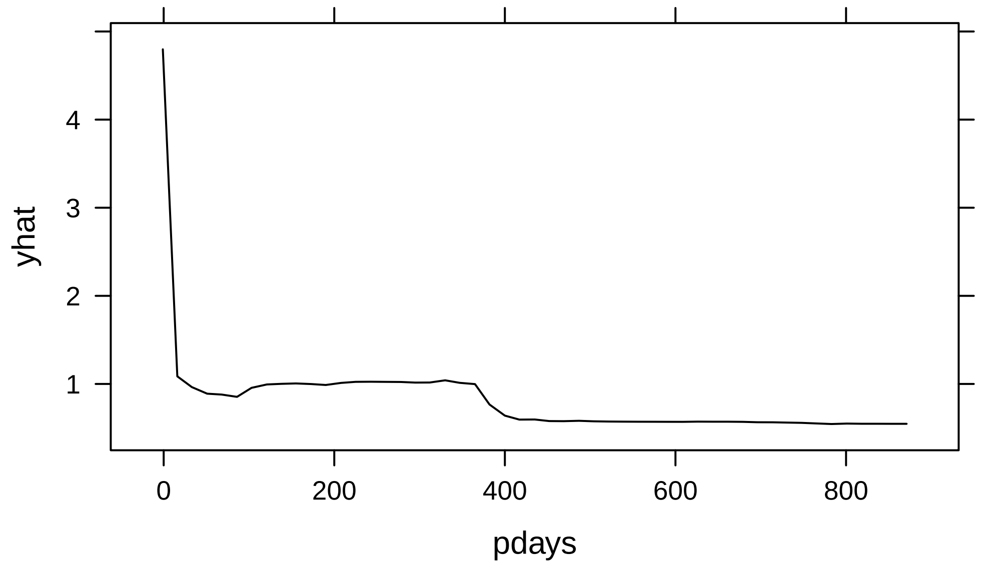
```r
partial(rf_model, pred.var = "campaign", plot = TRUE)
```



```r
partial(rf_model, pred.var = "balance", plot = TRUE)
```



```r
partial(rf_model, pred.var = "pdays", plot = TRUE)
```



## 2. Random Forest model (Summary)

We trained a Random Forest model (500 trees) to capture nonlinear patterns and identify which variables most strongly influence subscription to the term deposit. We extracted variable importance rankings and generated partial dependence plots for the top predictors.

### Key Findings

1. **Month is the strongest predictor:** Subscription probability varies significantly by contact month, indicating clear seasonality effects.
2. **Previous campaign outcome strongly predicts success:** Customers with a prior successful interaction have a much higher likelihood of subscribing.
3. **Age shows a nonlinear pattern:** Middle-aged customers (around 30–55) have the highest predicted probability. Younger and older customers show lower interest.
4. **pdays indicates recency matters:** Customers contacted recently, or first-time contacts, are more likely to subscribe than those contacted a long time ago.
5. **Balance shows diminishing returns:** Moderate balances correspond to higher predicted probability, while very low and very high balances are associated with lower probability.
6. **Campaign frequency has a negative effect:** As the number of contact attempts increases, subscription probability declines, suggesting diminishing effectiveness with repeated calls.

The Random Forest model suggests that campaign timing, prior interactions, age, and contact recency are key drivers of customer response. Targeting should prioritize middle-aged customers during high-performing months, especially those with successful prior outcomes or recent contact

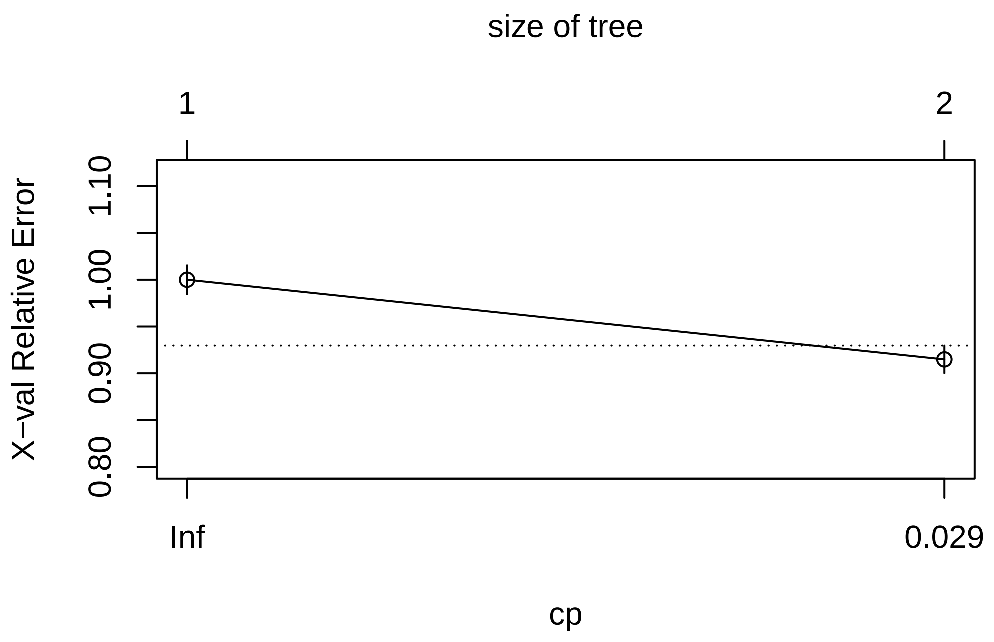
## 3. Decision Tree

```r
library(rpart)
library(rpart.plot)

# Split data
set.seed(123)
index <- sample(1:nrow(bank_data), 0.7 * nrow(bank_data))
train <- bank_data[index, ]
test  <- bank_data[-index, ]
```

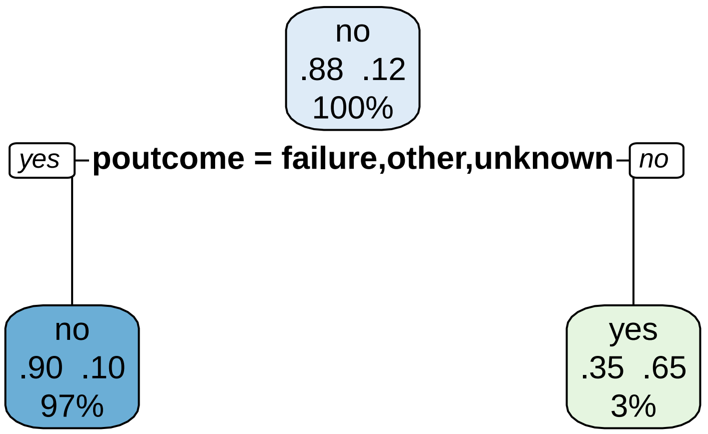
```r
# Fit decision tree
dt_model <- rpart(y ~ ., data = train, method = "class", cp = 0.01)

printcp(dt_model)
```

```
Classification tree:
rpart(formula = y ~ ., data = train, method = "class", cp = 0.01)

Variables actually used in tree construction:
[1] poutcome

Root node error: 3723/31647 = 0.11764

n= 31647

        CP nsplit rel error  xerror     xstd
1 0.085146      0   1.00000 1.00000 0.015395
2 0.010000      1   0.91485 0.91485 0.014808
```

```r
plotcp(dt_model)
```



```r
best_cp <- 0.029
dt_pruned <- prune(dt_model, cp = best_cp)

library(rpart.plot)
rpart.plot(dt_pruned, type = 2, extra = 104, fallen.leaves = TRUE)
```



```r
dt_pred <- predict(dt_pruned, test, type = "class")
table(Predicted = dt_pred, Actual = test$y)
```

```
         Actual
Predicted    no   yes
      no  11839  1279
      yes   159   287
```

## 3. Decision Tree (Summary)

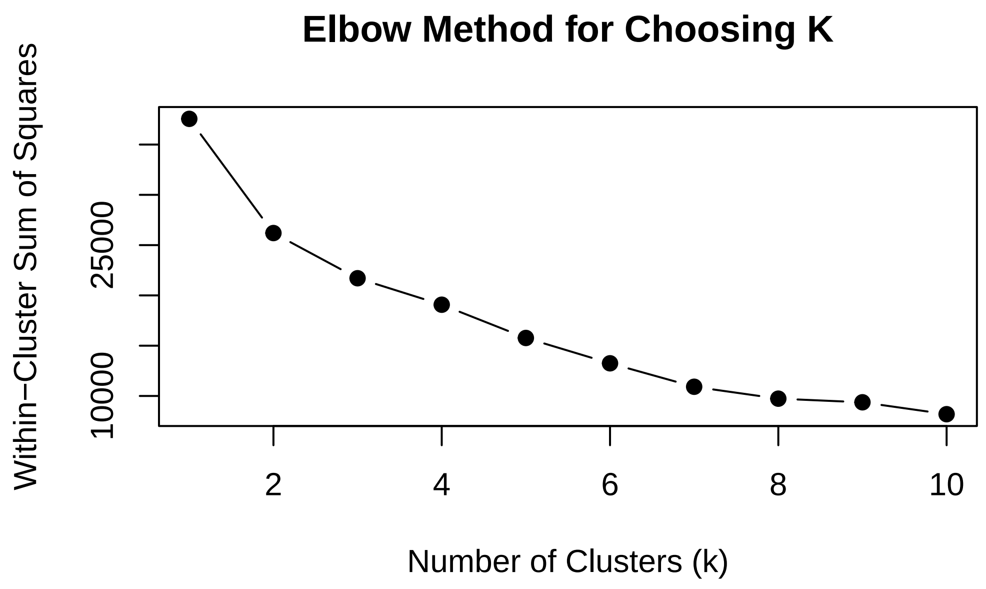
A pruned classification tree was built using cross-validation to determine the optimal complexity parameter (cp = 0.029). After pruning, the tree produced a single split based on *poutcome* (previous marketing outcome). This reflects that no other variable provided a sufficiently stable improvement in cross-validated accuracy.

The resulting model indicates that previous campaign success is the strongest discriminator: customers whose previous outcome was “success” show a markedly higher probability of subscribing ( 65%), whereas all other groups (“failure”, “other”, “unknown”) show very low subscription rates ( 10%). Although the tree has limited predictive performance due to the extreme class imbalance, it provides a clear and interpretable insight into which variable most differentiates customer behavior.

## 4. Cluster

```
# Data rangling
cluster_data <- train %>%
  select(age, balance, pdays, campaign, previous, poutcome)

cluster_dummy <- model.matrix(~ poutcome, data = cluster_data)[, -1]

cluster_numeric <- cluster_data %>%
  select(-poutcome) %>%
  cbind(cluster_dummy)

cluster_scaled <- scale(cluster_numeric)
```

```
set.seed(123)
sample_rows <- sample(1:nrow(cluster_scaled), size = 0.15 * nrow(cluster_scaled))
cluster_sample <- cluster_scaled[sample_rows, ]

wss <- numeric(10)
for (k in 1:10) {
  km_temp <- kmeans(cluster_sample, centers = k, nstart = 20)
  wss[k] <- km_temp$tot.withinss
}

plot(1:10, wss, type = "b", pch = 19,
     xlab = "Number of Clusters (k)",
     ylab = "Within-Cluster Sum of Squares",
     main = "Elbow Method for Choosing K")
```



```r
# run kmeans
k4 <- kmeans(cluster_scaled, centers = 4, nstart = 20)

# assign clusters to train
train$cluster <- k4$cluster
```

```r
k4$centers
```

```
           age     balance     pdays    campaign   previous poutcomeother
1 -0.097528878  0.02503037  1.848604 -0.09146813  1.3840437     4.8122015
2  0.198937561  0.22742820  1.222338 -0.30292825  1.0284279    -0.2077985
3 -0.001459108 -0.01514888 -0.412740  0.05113077 -0.2392114    -0.2077985
4 -0.013367815  0.03330833  1.991824 -0.25216554  0.9385283    -0.2077985
  poutcomesuccess poutcomeunknown
1       -0.186610      -2.0999058
2        5.358599      -2.0999058
3       -0.186610       0.4761968
4       -0.186610      -2.0991647
```

```r
train$cluster <- k4$cluster

cluster_yes_rate <- train %>%
  group_by(k4$cluster) %>%
  summarise(
    total = n(),
    yes = sum(y == "yes"),
    yes_rate = mean(y == "yes")
  )

cluster_yes_rate
```

```
# A tibble: 4 x 4
  `k4$cluster` total   yes yes_rate
         <int> <int> <int>    <dbl>
1            1  1310   220   0.168
2            2  1065   691   0.649
3            3 25796  2383   0.0924
4            4  3476   429   0.123
```

```
train %>%
  group_by(cluster) %>%
  summarise(across(c(age, balance, pdays, campaign, previous), mean))
```

```
# A tibble: 4 x 6
  cluster   age balance   pdays campaign  previous
    <int> <dbl>   <dbl>   <dbl>    <dbl>     <dbl>
1       1  39.9   1426. 228.        2.49 4.01
2       2  43.0   2034. 164.        1.82 3.13
3       3  40.9   1305.  -0.996     2.94 0.0000388
4       4  40.7   1451. 242.        1.98 2.91
```

## 4. Cluster (Summary)

To identify distinct customer segments and uncover which groups are most likely to subscribe to the term deposit, we performed a K-means clustering analysis using key behavioral and demographic variables: age, account balance, days since last contact (pdays), number of campaign contacts, and previous contact history (previous and poutcome). After scaling the data and evaluating multiple values of $k$, a four-cluster solution was selected as the most interpretable and meaningful for marketing purposes.

The four clusters revealed clear differences in customer engagement patterns:

- **Cluster 3 – Warm Re-engagement Leads (Highest success rate: 64.9%)**

  Customers in this group have been contacted multiple times in the past and show a moderate time gap since their last interaction. Their strong historical engagement makes them the most responsive segment and the primary target for focused follow-up campaigns.

- **Cluster 4 – Reactivation Candidates (Yes rate: 13.5%)**

  Similar to Cluster 3 but with longer gaps since last contact. These customers still show above-average conversion potential and can be effectively reached through reactivation efforts.

- **Cluster 1 – Young, Low-Engagement Customers (Yes rate: 9.4%)**

  This is the youngest segment with low balances and no prior contact history. Their low response rate suggests they are not ideal for high-cost outreach.

- **Cluster 2 – Older, Unfamiliar Customers (Yes rate: 9.0%)**

  Although financially stable, this group has almost no previous contact experience. They tend to be less receptive to marketing messages and are not recommended as a priority segment.