ML_Project

Sagar and Michael Ahana

2024-04-07

```r
pkg_list <- c("MASS", "ISLR", "dplyr", "caret", "ggplot2", "corrplot", "boot", "car", "glmnet", "tidyr"

# Install packages if needed
for (pkg in pkg_list) {
  # Try loading the library.
  if (!require(pkg, character.only = TRUE)) {
    # If the library cannot be loaded, install it from CRAN repository; then load.
    install.packages(pkg, repos = "https://cran.r-project.org")
    library(pkg, character.only = TRUE)
  }
}
```

```
## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.3.2

## Loading required package: ISLR

## Warning: package 'ISLR' was built under R version 4.3.3

## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 4.3.2

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: caret

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.3.2

## Loading required package: lattice

## Loading required package: corrplot

## Warning: package 'corrplot' was built under R version 4.3.3

## corrplot 0.92 loaded

## Loading required package: boot

##
## Attaching package: 'boot'

## The following object is masked from 'package:lattice':
##
##     melanoma

## Loading required package: car

## Warning: package 'car' was built under R version 4.3.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.2

##
## Attaching package: 'car'

## The following object is masked from 'package:boot':
##
##     logit

## The following object is masked from 'package:dplyr':
##
##     recode

## Loading required package: glmnet

## Warning: package 'glmnet' was built under R version 4.3.3

## Loading required package: Matrix
```

```
## Warning: package 'Matrix' was built under R version 4.3.2

## Loaded glmnet 4.1-8

## Loading required package: tidyr

## Warning: package 'tidyr' was built under R version 4.3.2

##
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack

## Loading required package: reshape2

## Warning: package 'reshape2' was built under R version 4.3.2

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

## Loading required package: neuralnet

## Warning: package 'neuralnet' was built under R version 4.3.3

##
## Attaching package: 'neuralnet'

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

1) EDA

```r
bank_data = read.csv("bank.csv")

# Display the structure of the dataset
str(bank_data)
```

```
## 'data.frame':    4521 obs. of  17 variables:
##  $ age      : int  30 33 35 30 59 35 36 39 41 43 ...
##  $ job      : chr  "unemployed" "services" "management" "management" ...
##  $ marital  : chr  "married" "married" "single" "married" ...
##  $ education: chr  "primary" "secondary" "tertiary" "tertiary" ...
##  $ default  : chr  "no" "no" "no" "no" ...
```

```
## $ balance  : int  1787 4789 1350 1476 0 747 307 147 221 -88 ...
## $ housing  : chr  "no" "yes" "yes" "yes" ...
## $ loan     : chr  "no" "yes" "no" "yes" ...
## $ contact  : chr  "cellular" "cellular" "cellular" "unknown" ...
## $ day      : int  19 11 16 3 5 23 14 6 14 17 ...
## $ month    : chr  "oct" "may" "apr" "jun" ...
## $ duration : int  79 220 185 199 226 141 341 151 57 313 ...
## $ campaign : int  1 1 1 4 1 2 1 2 2 1 ...
## $ pdays    : int  -1 339 330 -1 -1 176 330 -1 -1 147 ...
## $ previous : int  0 4 1 0 0 3 2 0 0 2 ...
## $ poutcome : chr  "unknown" "failure" "failure" "unknown" ...
## $ y        : chr  "no" "no" "no" "no" ...
```

```
# Summary statistics for numerical variables
summary(bank_data)
```

```
##       age            job              marital           education
## Min.   :19.00   Length:4521        Length:4521        Length:4521
## 1st Qu.:33.00   Class :character   Class :character   Class :character
## Median :39.00   Mode  :character   Mode  :character   Mode  :character
## Mean   :41.17
## 3rd Qu.:49.00
## Max.   :87.00
##   default             balance        housing              loan
## Length:4521        Min.   :-3313   Length:4521        Length:4521
## Class :character   1st Qu.:   69   Class :character   Class :character
## Mode  :character   Median :  444   Mode  :character   Mode  :character
##                    Mean   : 1423
##                    3rd Qu.: 1480
##                    Max.   :71188
##   contact               day           month             duration
## Length:4521        Min.   : 1.00   Length:4521        Min.   :   4
## Class :character   1st Qu.: 9.00   Class :character   1st Qu.: 104
## Mode  :character   Median :16.00   Mode  :character   Median : 185
##                    Mean   :15.92                      Mean   : 264
##                    3rd Qu.:21.00                      3rd Qu.: 329
##                    Max.   :31.00                      Max.   :3025
##    campaign         pdays          previous          poutcome
## Min.   : 1.000   Min.   : -1.00   Min.   : 0.0000   Length:4521
## 1st Qu.: 1.000   1st Qu.: -1.00   1st Qu.: 0.0000   Class :character
## Median : 2.000   Median : -1.00   Median : 0.0000   Mode  :character
## Mean   : 2.794   Mean   : 39.77   Mean   : 0.5426
## 3rd Qu.: 3.000   3rd Qu.: -1.00   3rd Qu.: 0.0000
## Max.   :50.000   Max.   :871.00   Max.   :25.0000
##       y
## Length:4521
## Class :character
## Mode  :character
##
##
##
```

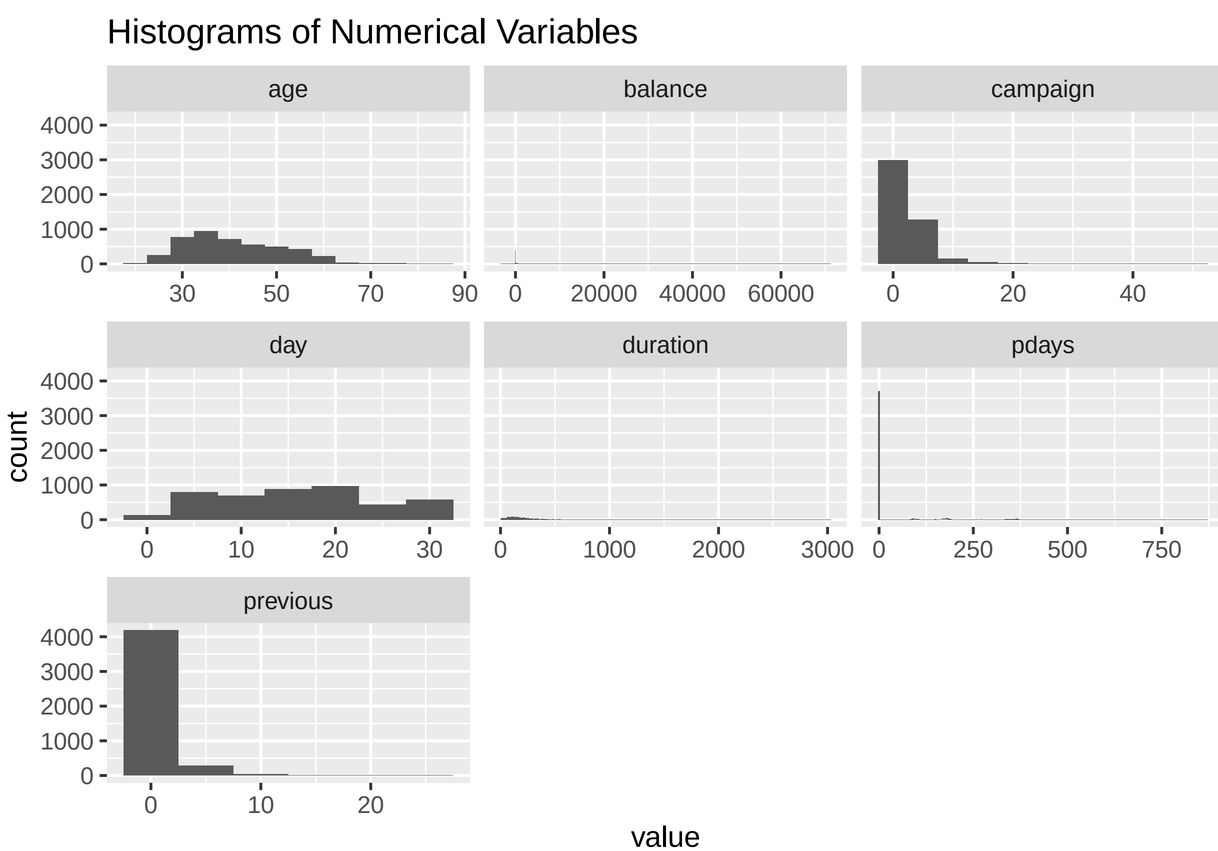
```r
# Summary statistics for categorical variables
#sapply(bank_data[, sapply(bank_data, is.factor)], table)

# Check for missing values
missing_values <- colSums(is.na(bank_data))
print(missing_values)
```

```
##       age       job   marital education   default   balance   housing      loan
##         0         0         0         0         0         0         0         0
##   contact       day     month  duration  campaign     pdays  previous  poutcome
##         0         0         0         0         0         0         0         0
##         y
##         0
```

```r
# Histograms for numerical variables
num_vars <- select_if(bank_data, is.numeric)
num_vars_long <- pivot_longer(num_vars, everything())

ggplot(num_vars_long, aes(value)) +
  geom_histogram(binwidth = 5) +
  facet_wrap(~ name, scales = "free_x") +
  labs(title = "Histograms of Numerical Variables")
```



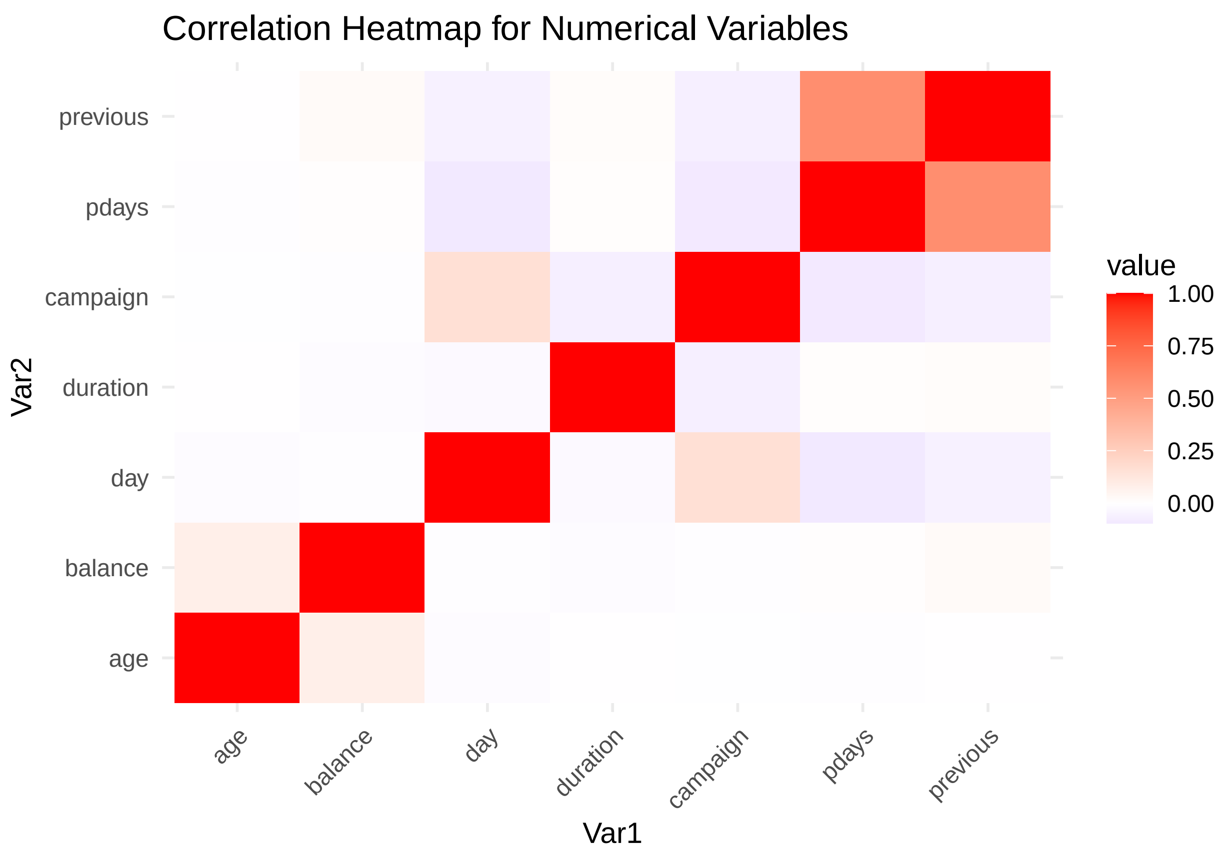
```r
# Correlation matrix for numerical variables
cor_matrix <- cor(num_vars)
print("Correlation Matrix for Numerical Variables:")
```

```
## [1] "Correlation Matrix for Numerical Variables:"
```

```r
print(cor_matrix)
```

```
##                   age      balance          day     duration     campaign
## age       1.000000000  0.083820142 -0.017852632 -0.002366889 -0.005147905
## balance   0.083820142  1.000000000 -0.008677052 -0.015949918 -0.009976166
## day      -0.017852632 -0.008677052  1.000000000 -0.024629306  0.160706069
## duration -0.002366889 -0.015949918 -0.024629306  1.000000000 -0.068382000
## campaign -0.005147905 -0.009976166  0.160706069 -0.068382000  1.000000000
## pdays    -0.008893530  0.009436676 -0.094351520  0.010380242 -0.093136818
## previous -0.003510917  0.026196357 -0.059114394  0.018080317 -0.067832630
##                 pdays     previous
## age      -0.008893530 -0.003510917
## balance   0.009436676  0.026196357
## day      -0.094351520 -0.059114394
## duration  0.010380242  0.018080317
## campaign -0.093136818 -0.067832630
## pdays     1.000000000  0.577561827
## previous  0.577561827  1.000000000
```

```r
# Heatmap of the correlation matrix
ggplot(melt(cor_matrix), aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Correlation Heatmap for Numerical Variables")
```



EDA Summary:

We first understood the data by viewing structure and summary statistics. We then checked the missing values and generated histograms for numerical variables to understand their distributions.We then calculated the correlation matrix for numerical variables and plotted a heat-map to visualize the correlations.

Overall, we did a comprehensive exploration of the dataset, including its structure, summary statistics, missing values, distribution of numerical variables, and correlations between numerical variables.

Analisis:

Classification - Predicting Term Deposit Subscription by using logistic regression, rf, knn:

1. GLM (logistic Regression):

```
bank_data = read.csv("bank.csv")

# Train-test split
set.seed(730216)

train_index <- createDataPartition(bank_data$y, p = 0.8, list = FALSE)
train_data <- bank_data[train_index, ]
test_data <- bank_data[-train_index, ]

train_with_cv <- function(data, method, family, k) {
  # Define cross-validation settings
  ctrl <- trainControl(method = "cv", number = k)

  # Initialize variable to store total computation time
```

```r
  total_elapsed_time <- 0

  # Initialize vector to store accuracy values for each fold
  accuracy <- numeric(k)


# Train model with cross-validation
  for (i in 1:k) {
    # Start measuring computation time
    start_time <- Sys.time()

    # Train model with cross-validation
    formula <- as.formula(paste("y ~ ."))
    model <- train(formula,
                   data = data,
                   method = method,
                   family = family,
                   trControl = ctrl)

    # End measuring computation time
    end_time <- Sys.time()

    # Compute elapsed time for this fold
    elapsed_time <- round(end_time - start_time, 4)

    # Accumulate total elapsed time
    total_elapsed_time <- total_elapsed_time + elapsed_time

    # Predict on test data
    predictions <- predict(model, newdata = test_data)

    # Convert predictions and test_data$y to factors with the same levels
    predictions <- factor(predictions, levels = c("yes", "no"))
    test_data$y <- factor(test_data$y, levels = c("yes", "no"))

    # Create confusion matrix
    conf_matrix <- confusionMatrix(predictions, test_data$y)

    # Extract accuracy and store in accuracy vector
    accuracy[i] <- conf_matrix$overall["Accuracy"]
  }

  # Return list containing confusion matrix, total elapsed time, and accuracy values
  return(list(conf_matrix = conf_matrix, total_elapsed_time = total_elapsed_time, accuracy = accuracy[k
}

# Logistic
logit <- train_with_cv(data = train_data, method = "glm", family = "binomial", k = 5)
print(logit)
```

```
## $conf_matrix
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction yes  no
##        yes  33  14
##        no   71 786
##
##                Accuracy : 0.906
##                  95% CI : (0.885, 0.9242)
##     No Information Rate : 0.885
##     P-Value [Acc > NIR] : 0.02451
##
##                   Kappa : 0.3937
##
##  Mcnemar's Test P-Value : 1.247e-09
##
##             Sensitivity : 0.31731
##             Specificity : 0.98250
##          Pos Pred Value : 0.70213
##          Neg Pred Value : 0.91715
##              Prevalence : 0.11504
##          Detection Rate : 0.03650
##    Detection Prevalence : 0.05199
##       Balanced Accuracy : 0.64990
##
##        'Positive' Class : yes
##
##
## $total_elapsed_time
## Time difference of 4.5904 secs
##
## $accuracy
## [1] 0.9059735
```

Logistic Regression Analysis: The model achieved an accuracy of 90.60%, with a sensitivity of 31.73% and specificity of 98.25%, indicating its ability to correctly identify term deposit subscriptions and non-subscriptions. Additionally, the positive predictive value stands at 70.21%, suggesting its reliability in predicting actual term deposit subscriptions.

2. KNN:

```r
train_with_rf_knn <- function(data, method, k) {
  # Define cross-validation settings
  ctrl <- trainControl(method = "cv", number = k)

  # Initialize variables to store total computation time and accuracy
  total_elapsed_time <- 0
  accuracy <- numeric(k)

  # Train model with cross-validation
  for (i in 1:k) {
    # Start measuring computation time
    start_time <- Sys.time()

    # Train model with cross-validation
    formula <- as.formula(paste("y ~ ."))
```

```r
    model <- train(formula,
                   data = data,
                   method = method,
                   trControl = ctrl)

    # End measuring computation time
    end_time <- Sys.time()

    # Compute elapsed time for this fold
    elapsed_time <- round(end_time - start_time, 4)

    # Accumulate total elapsed time
    total_elapsed_time <- total_elapsed_time + elapsed_time

  }

   # Predict on test data
  predictions <- predict(model, newdata = test_data)

  # Convert predictions and test_data$y to factors with the same levels
  predictions <- factor(predictions, levels = c("yes", "no"))
  test_data$y <- factor(test_data$y, levels = c("yes", "no"))

  # Create confusion matrix
  conf_matrix <- confusionMatrix(predictions, test_data$y)

   # Extract accuracy
    accuracy[i] <- conf_matrix$overall["Accuracy"]


  # Return list containing confusion matrix and total elapsed time
  return(list(conf_matrix = conf_matrix, total_elapsed_time = total_elapsed_time,accuracy = accuracy[k])
}


# KNN:
KNN = train_with_rf_knn(data = train_data,  method = "knn", k = 5)
print(KNN)
```

```
## $conf_matrix
## Confusion Matrix and Statistics
##
##           Reference
## Prediction yes  no
##        yes  12  19
##        no   92 781
##
##                Accuracy : 0.8772
##                  95% CI : (0.854, 0.8979)
##     No Information Rate : 0.885
##     P-Value [Acc > NIR] : 0.7844
##
##                   Kappa : 0.1319
```

```
##
##  Mcnemar's Test P-Value : 8.261e-12
##
##             Sensitivity : 0.11538
##             Specificity : 0.97625
##          Pos Pred Value : 0.38710
##          Neg Pred Value : 0.89462
##              Prevalence : 0.11504
##          Detection Rate : 0.01327
##    Detection Prevalence : 0.03429
##       Balanced Accuracy : 0.54582
##
##        'Positive' Class : yes
##
##
## $total_elapsed_time
## Time difference of 9.9082 secs
##
## $accuracy
## [1] 0.8772124
```

KNN Interpretation: The model achieved an accuracy of 87.72%, with a sensitivity of 11.538% and specificity of 97.62%, indicating its ability to correctly identify term deposit subscriptions and non-subscriptions. Additionally, the positive predictive value stands at 38.71%, suggesting its reliability in predicting actual term deposit subscriptions.

3. Random Forest

```
model_rf = train_with_rf_knn(data = train_data,  method = "rf", k = 5)
print(model_rf)
```

```
## $conf_matrix
## Confusion Matrix and Statistics
##
##           Reference
## Prediction yes  no
##        yes  42  24
##        no   62 776
##
##                Accuracy : 0.9049
##                  95% CI : (0.8838, 0.9232)
##     No Information Rate : 0.885
##     P-Value [Acc > NIR] : 0.03156
##
##                   Kappa : 0.4445
##
##  Mcnemar's Test P-Value : 6.613e-05
##
##             Sensitivity : 0.40385
##             Specificity : 0.97000
##          Pos Pred Value : 0.63636
##          Neg Pred Value : 0.92601
##              Prevalence : 0.11504
##          Detection Rate : 0.04646
```

```
##    Detection Prevalence : 0.07301
##       Balanced Accuracy : 0.68692
##
##        'Positive' Class : yes
##
##
## $total_elapsed_time
## Time difference of 8.8339 mins
##
## $accuracy
## [1] 0.9048673
```

RF Interpretation: The model achieved an accuracy of 90.93%, with a sensitivity of 40.38% and specificity of 97.50%, indicating its ability to correctly identify term deposit subscriptions and non-subscriptions. Additionally, the positive predictive value stands at 67.74%, suggesting its reliability in predicting actual term deposit subscriptions.

Bootstrapping to check the authenticity of my models:

```r
accuracy_logit = logit$accuracy
accuracy_knn = KNN$accuracy
accuracy_rf = model_rf$accuracy

accuracy_logit; accuracy_knn; accuracy_rf
```

```
## [1] 0.9059735
```

```
## [1] 0.8772124
```

```
## [1] 0.9048673
```

```r
num_bootstrap = 1000

# Store accuracies in a list
accuracies <- list(logit = accuracy_logit, knn = accuracy_knn, rf = accuracy_rf)

# Perform bootstrapping for each model
bootstrap_results <- lapply(accuracies, function(accuracy) {
  boot(data = accuracy, statistic = function(x, i) mean(x[i]), R = num_bootstrap)
})

# Summarize bootstrap results for each model
summary_bootstrap_results <- lapply(bootstrap_results, summary)
summary_bootstrap_results
```

```
## $logit
##      R original bootBias bootSE bootMed
## 1 1000  0.90597        0      0 0.90597
##
## $knn
##      R original bootBias bootSE bootMed
## 1 1000  0.87721        0      0 0.87721
```

```
##
## $rf
##      R original bootBias bootSE bootMed
## 1 1000  0.90487        0      0 0.90487
```

Extracting important with RF based on score

```
    # Train random forest model
    #model <- train(y ~ ., data = bank_data, method = "rf")

    # Extract variable importance
    #importance_scores <- varImp(rf)

    # Plot feature importance
    #print(importance_scores)
```

We've used random forest classification model to analyze the feature importance, indicating which features have the most significant impact on the prediction of term deposit subscription.

Above feature importance analysis reveals the relative importance of each feature in predicting the target variable. The most influential feature is "duration," which has a importance score of 100. This suggests that the duration of the call plays a significant role in determining whether a client subscribes to a term deposit. Other important features include "balance," "age," and "day," which also contribute significantly to the predictive power of the model. Additionally, factors such as the outcome of the previous marketing campaign ("poutcomesuccess") and the number of days since the client was last contacted ("pdays") are among the top predictors of term deposit subscription. Overall, understanding these influential features can provide valuable insights for targeted marketing strategies and customer engagement efforts.

3.Customer Segmentation based on 2 of the important features extracted from above by using K-means Clustering:

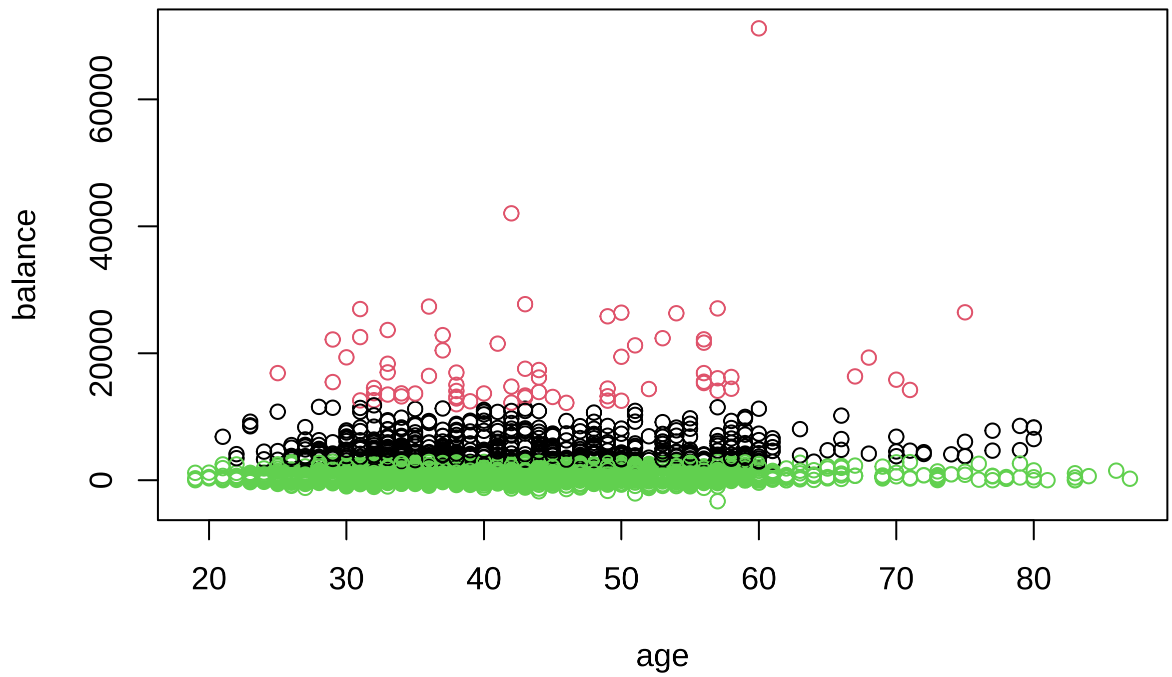
```
# Select features for clustering
cluster_data <- bank_data[, c("age", "balance")]

# Perform K-means clustering
kmeans_result <- kmeans(cluster_data, centers = 3)

#Centres
print(kmeans_result$centers)
```

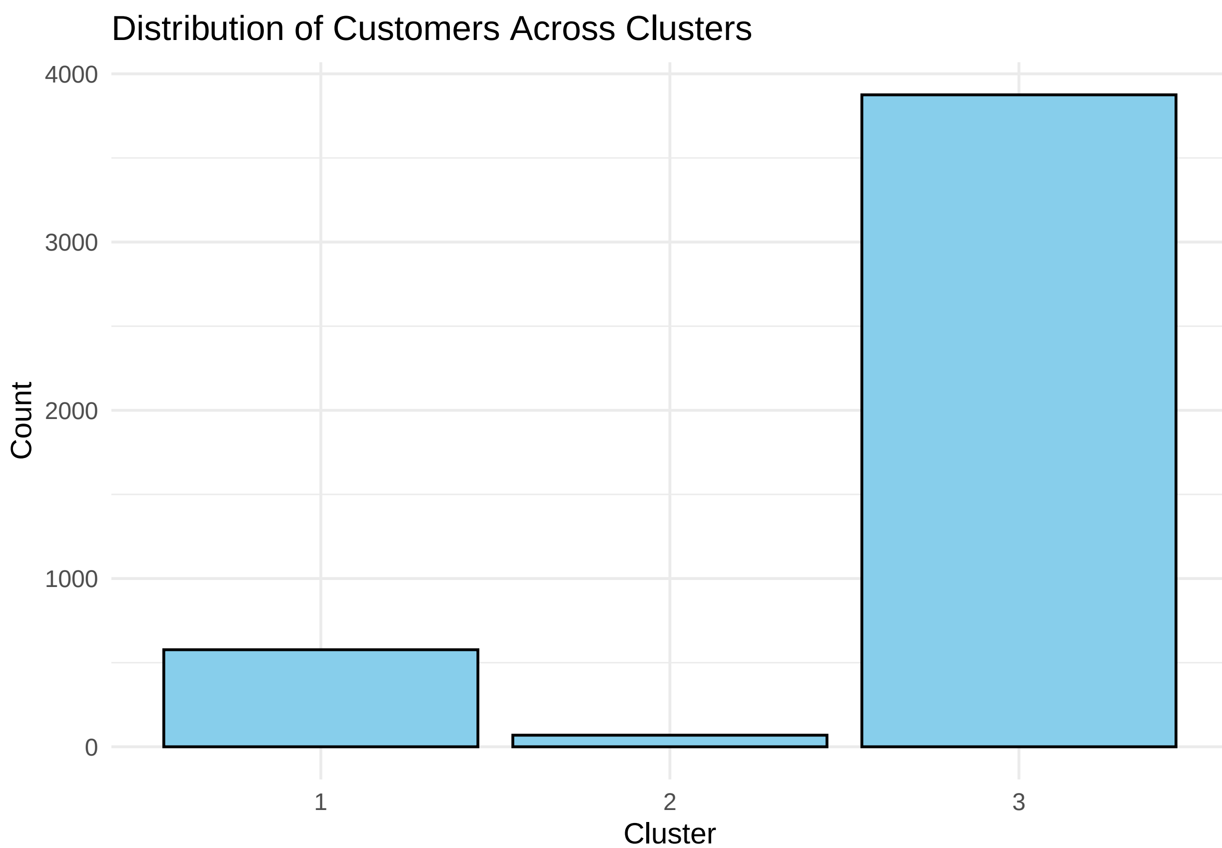
```
##        age    balance
## 1 42.93414  5250.4246
## 2 44.72464 18456.9130
## 3 40.84413   549.3714
```

```
# Visualize clusters
plot(cluster_data, col = kmeans_result$cluster)
```




```r
# Create a data frame with cluster labels
cluster_data <- data.frame(Cluster = factor(kmeans_result$cluster))

# Plot distribution of customers across clusters
ggplot(cluster_data, aes(x = Cluster)) +
  geom_bar(fill = "skyblue", color = "black") +
  labs(title = "Distribution of Customers Across Clusters",
       x = "Cluster", y = "Count") +
  theme_minimal()
```



By conducting K-means clustering based on age and balance, we grouped clients with similar characteristics, facilitating targeted marketing and service offerings tailored to different client segments, ultimately enhancing customer satisfaction and retention.

Intepretation:

Based on the centroids obtained from the K-means clustering analysis:

Cluster 1: Customers in this cluster have an average age of approximately 42.93 years.The average balance for customers in this cluster is $5250.42. This cluster might represent a segment of middle-aged customers with moderate to high account balances.

Cluster 2: Customers in this cluster have an average age of approximately 44.72 years.The average balance for customers in this cluster is significantly higher, at $18456.91.This cluster could represent older customers with substantially higher account balances, potentially indicating a segment of affluent or high-net-worth individuals.

Cluster 3: Customers in this cluster have an average age of approximately 40.84 years.The average balance for customers in this cluster is much lower, at $549.37.This cluster may represent a younger segment of customers with lower account balances, possibly including students, young professionals, or individuals with limited financial resources.