Università degli Studi di Padova

Dipartimento Matematica

Dipartimento di Matematica "Tullio Levi-Civita"

Statistical Learning Final Project

Employee Attrition Classification

Authors

Zeynep TUTAR - 2106038

Aysenur Oya ÖZEN - 2107501

Supervisor

Prof. Alberto ROVERATO

Academic Year:
2023/2024

Contents

Introduction to Dataset 2

Description of the Features . . . 2

Data Analysis 3

Data Preprocessing . . . 4

Data Preparation 15

Handling Categorical Features . . . 15

Normalization . . . 16

Train-Test-Split . . . 16

Predictive Classification Models 17

Logistic Regression . . . 17

Basic Logistic Classifier . . . 17

Logistic Regression with Backward Stepwise Search . . . 20

Logistic Regression with Shrinkage Method . . . 22

Comparison of Logistic Classifiers . . . 23

LDA & QDA . . . 26

Model Results 28

Performance Metrics and Confusion Matrix . . . 28

Introduction to Dataset

The aim of this project is to develop two predictive models to determine employee attrition of a company. The dataset[1] used for this project is a simulated dataset designed for the analysis and prediction of employee attrition. It contains detailed information about various aspects of an employee's profile, including demographics, job-related features, and personal circumstances.The dataset contains 74,498 samples. Each record includes a unique Employee ID and features that influence employee attrition. The goal is to understand the factors contributing to attrition and develop predictive models to identify at-risk employees.

The dataset is already split into train and test but in order to better understand the data, it is crucial to analyse the dataset as a whole.

```r
# import the train and test datasets
data_train <- read.csv("data/train.csv", stringsAsFactors = TRUE)
data_test <- read.csv("data/test.csv", stringsAsFactors = TRUE)

# merge the datasets
data <- rbind(data_train, data_test)
attach(data)
```

Description of the Features

The features of the dataset are presented below:

- **Employee ID:** A unique identifier assigned to each employee.
- **Age:** The age of the employee, ranging from 18 to 60 years.
- **Gender:** The gender of the employee
- **Years at Company:** The number of years the employee has been working at the company.
- **Monthly Income:** The monthly salary of the employee, in dollars.
- **Job Role:** The department or role the employee works in, encoded into categories such as Finance, Healthcare, Technology, Education, and Media.
- **Work-Life Balance:** The employee's perceived balance between work and personal life, (Poor, Below Average, Good, Excellent)
- **Job Satisfaction:** The employee's satisfaction with their job: (Very Low, Low, Medium, High)
- **Performance Rating:** The employee's performance rating: (Low, Below Average, Average, High)
- **Number of Promotions:** The total number of promotions the employee has received.
- **Distance from Home:** The distance between the employee's home and workplace, in miles.
- **Education Level:** The highest education level attained by the employee: (High School, Associate Degree, Bachelor's Degree, Master's Degree, PhD)
- **Marital Status:** The marital status of the employee: (Divorced, Married, Single)
- **Job Level:** The job level of the employee: (Entry, Mid, Senior)

[1] https://www.kaggle.com/datasets/stealthtechnologies/employee-attrition-dataset/data

- **Company Size:** The size of the company the employee works for: (Small,Medium,Large)
- **Company Tenure:** The total number of years the employee has been working in the industry.
- **Remote Work:** Whether the employee works remotely: (Yes or No)
- **Leadership Opportunities:** Whether the employee has leadership opportunities: (Yes or No)
- **Innovation Opportunities:** Whether the employee has opportunities for innovation: (Yes or No)
- **Company Reputation:** The employee's perception of the company's reputation: (Very Poor, Poor,Good, Excellent)
- **Employee Recognition:** The level of recognition the employee receives:(Very Low, Low, Medium, High)
- **Attrition:** Whether the employee has left the company, encoded as 0 (stayed) and 1 (Left).

Data Analysis

In order to develop predictive models, first it is necessary to perform exploratory data analysis (EDA) and modify the format of the data if necessary.

```
# installing required libraries
library(class)
library(car)
library(corrplot)
library(glmnet)
library(pROC)
library(knitr)
library(leaps)
library(MASS)
```

```
# Descriptive statistics
str(data)
```

```
'data.frame':   74498 obs. of  24 variables:
 $ Employee.ID          : int  8410 64756 30257 65791 65026 24368 64970 36999 32714 15944 ...
 $ Age                     : int  31 59 24 36 56 38 47 48 57 24 ...
 $ Gender                  : Factor w/ 2 levels "Female","Male": 2 1 1 1 2 1 2 2 2 1 ...
 $ Years.at.Company        : int  19 4 10 7 41 3 23 16 44 1 ...
 $ Job.Role             : Factor w/ 5 levels "Education","Finance",..: 1 4 3 1 1 5 1 2 1 3 ...
 $ Monthly.Income          : int  5390 5534 8159 3989 4821 9977 3681 11223 3773 7319 ...
 $ Work.Life.Balance      : Factor w/ 4 levels "Excellent","Fair",..: 1 4 3 3 2 2 2 1 3 4 ...
 $ Job.Satisfaction      : Factor w/ 4 levels "High","Low","Medium",..: 3 1 1 1 4 1 1 4 3 1 ...
 $ Performance.Rating    : Factor w/ 4 levels "Average","Below Average",..: 1 4 4 3 1 2 3 3 3 1 ...
 $ Number.of.Promotions    : int  2 3 0 1 0 3 1 2 1 1 ...
 $ Overtime                : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 2 1 2 2 ...
 $ Distance.from.Home      : int  22 21 11 27 71 37 75 5 39 57 ...
 $ Education.Level        : Factor w/ 5 levels "Associate Degree",..: 1 4 2 3 3 2 3 4 3 5 ...
 $ Marital.Status        : Factor w/ 3 levels "Divorced","Married",..: 2 1 2 3 1 2 1 2 2 3 ...
 $ Number.of.Dependents    : int  0 3 3 2 0 0 3 4 4 4 ...
 $ Job.Level               : Factor w/ 3 levels "Entry","Mid",..: 2 2 2 2 3 2 1 1 1 1 ...
 $ Company.Size            : Factor w/ 3 levels "Large","Medium",..: 2 2 2 3 2 2 3 2 2 1 ...
 $ Company.Tenure          : int  89 21 74 50 68 47 93 88 75 45 ...
 $ Remote.Work             : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 1 1 1 1 ...
 $ Leadership.Opportunities: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
 $ Innovation.Opportunities: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 2 ...
 $ Company.Reputation      : Factor w/ 4 levels "Excellent","Fair",..: 1 2 4 3 2 2 3 1 2 3 ...
 $ Employee.Recognition    : Factor w/ 4 levels "High","Low","Medium",..: 3 2 2 3 3 1 3 2 3 2 ...
 $ Attrition               : Factor w/ 2 levels "Left","Stayed": 2 2 2 2 2 1 1 2 2 1 ...
```

Data Preprocessing

To prepare the dataset for further analysis, several data preprocessing steps are performed:

1. Removing Columns

Employee.ID and Company.Tenure dropped as they are not useful for predictive modeling. Company.Tenure column gives logically incorrect numerical values.

```r
# Drop Employee Id and Company.Tenure
data <- data[, !names(data) %in% "Employee.ID"]
data <- data[, !names(data) %in% "Company.Tenure"]

# Copy Data for further Analysis
data_detailed <- data
```

2. Numerical and Categorical Variables Separation

Numerical and categorical variables were separated for targeted analysis. Summary statistics were then used to provide a quick overview of the distribution of numerical features.

Summary statistics for numerical variables reveal that the ages of employees range from 18 to 59,indicating a workforce that spans multiple generations.Employees have been with the company for a wide range of 1 to 51 years suggesting a mix of new and long-term staff.Monthly incomes vary widely, from 1,226 to 16,149 units, indicating financial situation specific to the individual, industry or other variables. Lastly, the distance from home ranges from 1 to 99 units, which suggests that while some employees live close to their workplace, others may have longer commutes. This variability highlights the diverse nature of the workforce.

```r
# Numerical and categorical variables separation
numeric_vars <- sapply(data, is.numeric)
categoric_vars <- sapply(data, function(x) is.factor(x) || is.character(x))

# Taking names from numerical and categorical variables for
# distribution graph
categoric_var_names <- names(data)[categoric_vars]
numeric_var_names <- names(data)[numeric_vars]

# Numeric values summary
summary(data[, numeric_vars])
```

```
      Age        Years.at.Company Monthly.Income   Number.of.Promotions
 Min.   :18.00   Min.   : 1.00    Min.   : 1226   Min.   :0.0000
 1st Qu.:28.00   1st Qu.: 7.00    1st Qu.: 5652   1st Qu.:0.0000  
 Median :39.00   Median :13.00    Median : 7348   Median :1.0000  
 Mean   :38.53   Mean   :15.72    Mean   : 7299   Mean   :0.8329  
 3rd Qu.:49.00   3rd Qu.:23.00    3rd Qu.: 8876   3rd Qu.:2.0000  
 Max.   :59.00   Max.   :51.00    Max.   :16149   Max.   :4.0000  
 Distance.from.Home Number.of.Dependents
 Min.   : 1.00      Min.   :0.00        
 1st Qu.:25.00      1st Qu.:0.00        
 Median :50.00      Median :1.00        
 Mean   :49.99      Mean   :1.65        
 3rd Qu.:75.00      3rd Qu.:3.00        
 Max.   :99.00      Max.   :6.00        
```

3. Missing Values Analysis

No missing values were found in the data set, so there was no need to apply removing of null values operation.

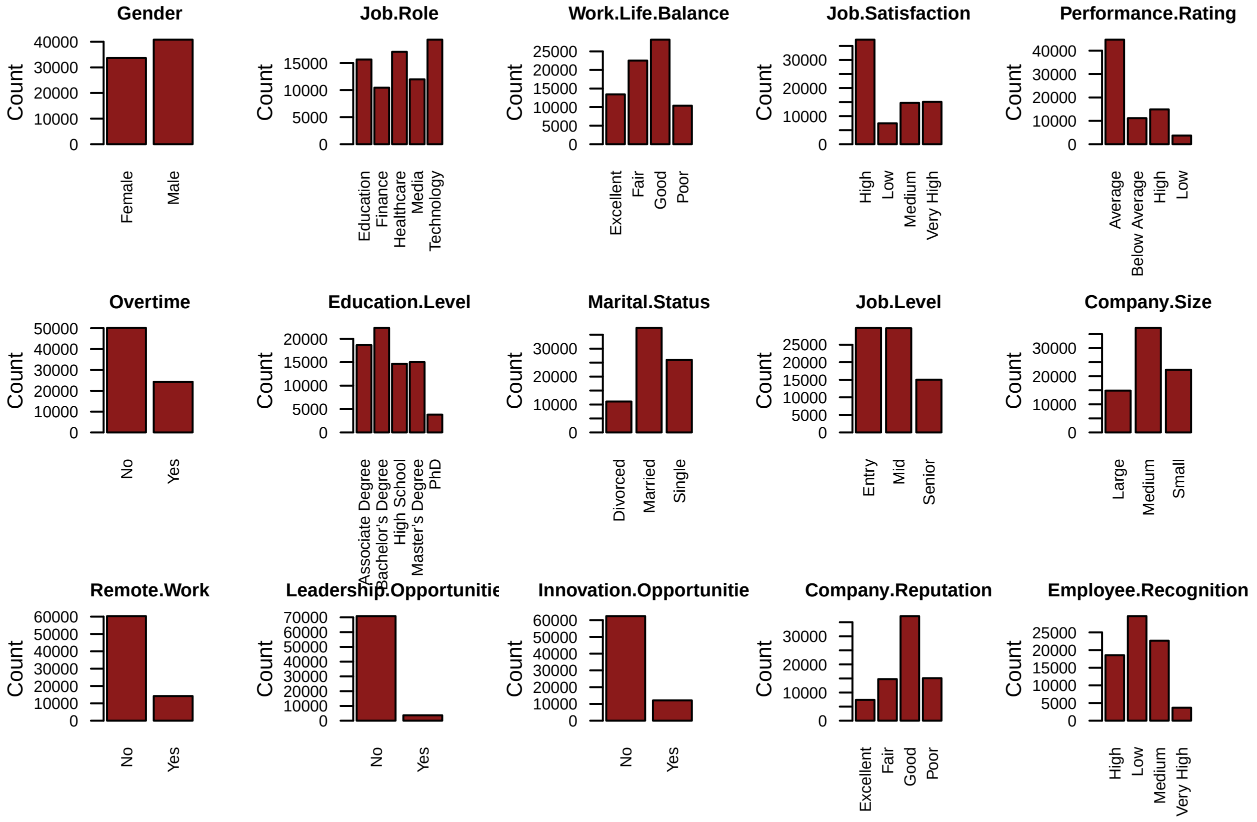
```r
# Checking if there are missing values in the entire dataset
total_na <- sum(is.na(data))
cat("Total number of missing values:", total_na, "\n")
```

```
Total number of missing values: 0 
```

4. Categorical Variables Distribution

Inferences from the bar charts of categorical variables: The fact that the gender distribution is almost equal shows that there is no gender bias in the data.Job roles skew towards technology and media, indicating a focus on these sectors.Most employees rate their work-life balance as good or excellent; This indicates high satisfaction; however, job satisfaction varies, with some rating it as low or medium.Performance ratings are mostly average, indicating a need for better performance management.The fact that most employees have not received any promotions highlights potential career development issues.Overtime is rare, indicating manageable workloads.Education levels vary, and many have bachelor’s degrees.Marital status indicates more married workers.Job levels are mostly entry and intermediate, indicating a younger workforce.Company size is evenly distributed, indicating different operational scales.The rarity of remote work points to traditional office culture.Limited leadership and innovation opportunities suggest areas for development.The company’s reputation is mostly good, but employee recognition is low to moderate, indicating room for improvement.

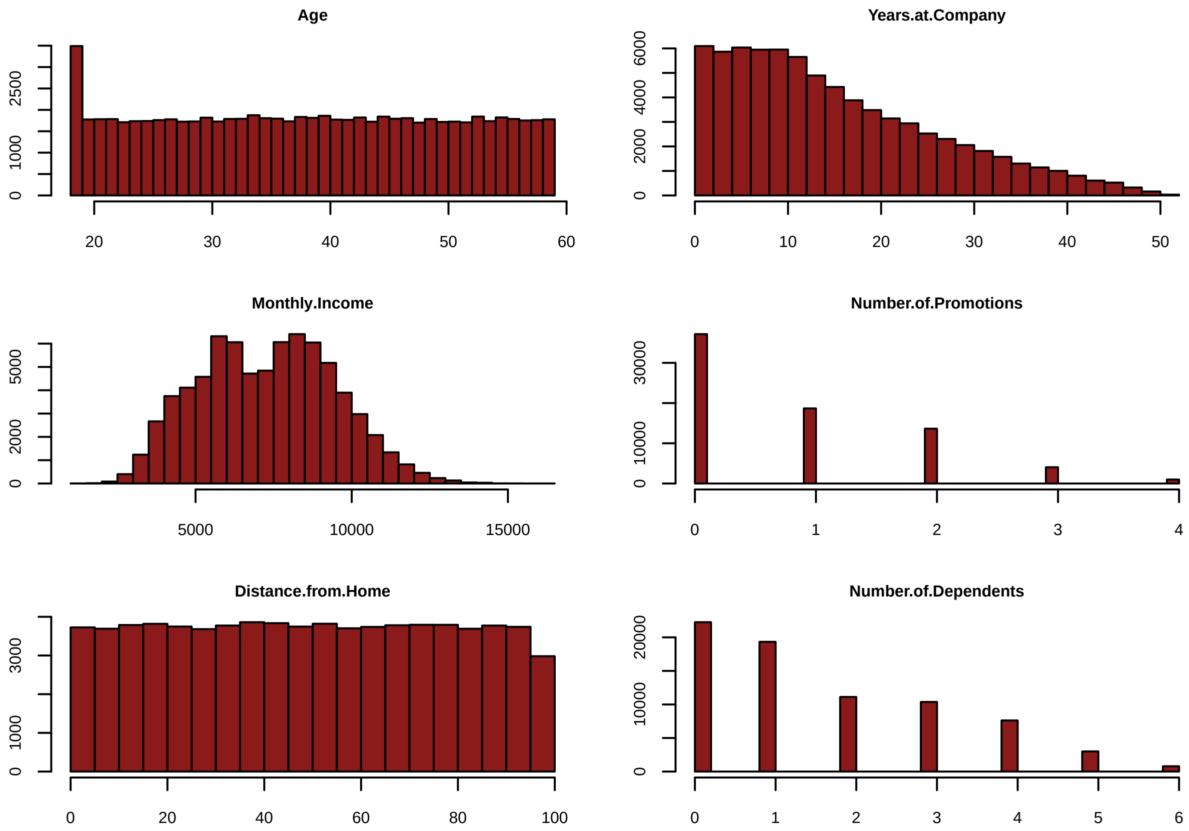
```r
# Categorical variables distribution
par(mfrow = c(3, 5), mar = c(5, 4, 2, 2) + 0.1)
for (cat_var in categoric_var_names[-16]) {
    barplot(table(data[[cat_var]]), main = paste(cat_var), ylab = "Count",
        col = "firebrick4", cex.names = 0.7, las = 2, cex.main = 0.9, cex.axis = 0.8)
}
```



5. Numerical Values Distribution

Inferences from the histograms for the numerical variables: The age distribution is fairly uniform, indicating a wide age range among employees.The years at the company show a right-skewed distribution, with most employees having shorter tenures and fewer employees staying beyond 30 years.Monthly income has a roughly normal distribution, peaking around 5,000 to 10,000 units, suggesting that most employees earn within this range.The distance from home distribution is relatively uniform, indicating that employees live at various distances from their workplace.

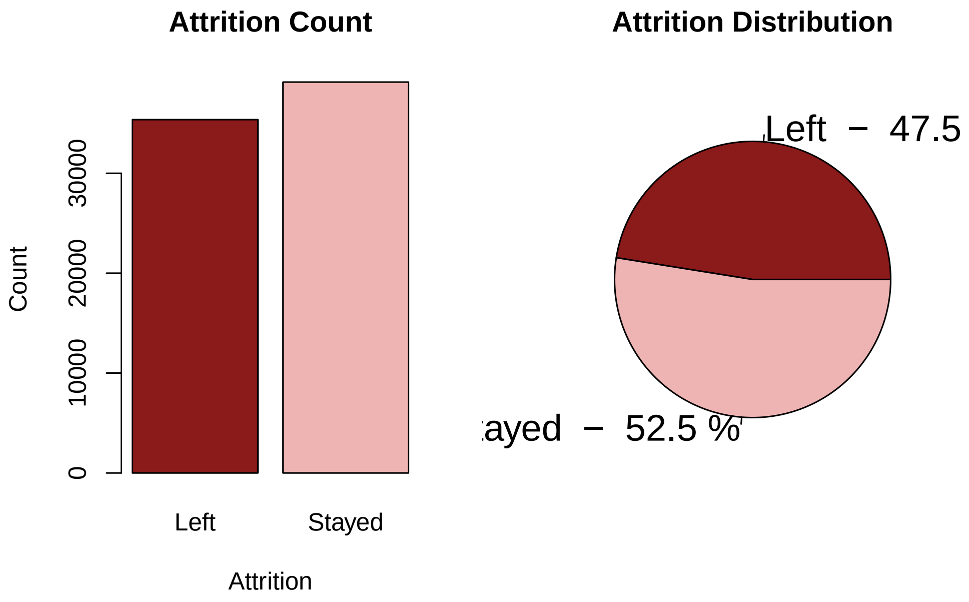
```
# Numerical variables distribution
par(mfrow = c(3, 2), mar = c(3, 3, 2, 1))
for (num_var in numeric_var_names) {
    hist(data[[num_var]], main = paste(num_var), xlab = "", ylab = "", col = "firebrick4",
        breaks = 30, cex.main = 0.8, cex.axis = 0.8, cex.lab = 0.8)
}
```



6. Target Value Distribution

The pie chart depicting attrition distribution shows that 47.5% of employees left the company, while 52.5% stayed. This indicates a relatively balanced split between those who left and those who remained. Such a near-equal distribution suggests that significant turnover, highlighting the need for strategies to improve retention. Understanding the factors contributing to attrition could help in developing targeted initiatives to retain employees and enhance overall workforce stability.

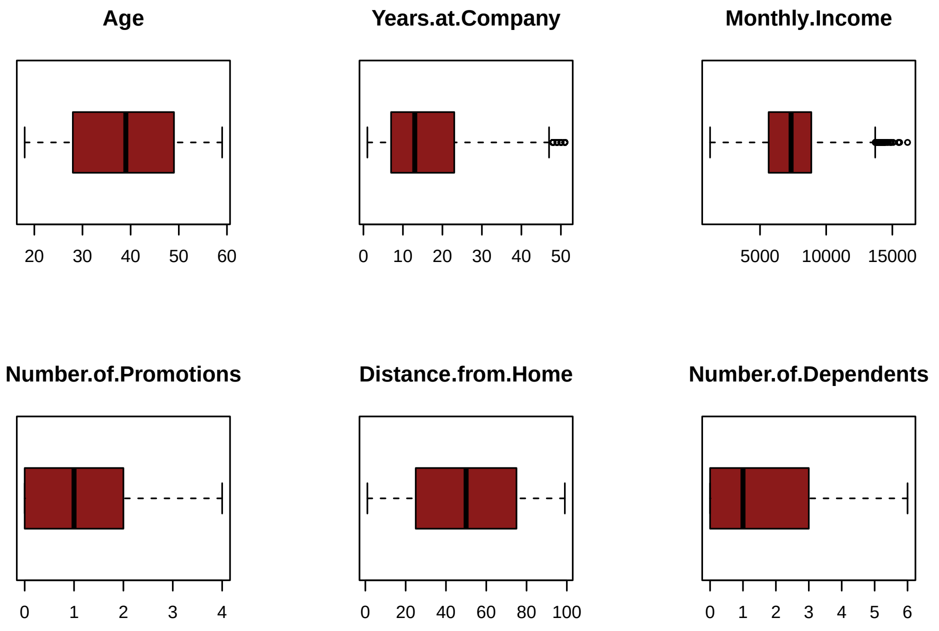
```r
# Target value distribution
par(mfrow = c(1, 2))
barplot(table(data$Attrition), main = "Attrition Count", xlab = "Attrition",
    ylab = "Count", col = c("firebrick4", "rosybrown2"))

# Target values distribution with pie chart
attrition_table <- table(data$Attrition)
attrition_df <- as.data.frame(attrition_table)
colnames(attrition_df) <- c("Attrition", "Count")
attrition_df$Percentage <- round(100 * attrition_df$Count/sum(attrition_df$Count),
    1)
pie(attrition_df$Count, labels = paste(attrition_df$Attrition, " - ",
 ↪  attrition_df$Percentage,
    "%"), col = c("firebrick4", "rosybrown2"), main = "Attrition Distribution",
    cex = 1.5, radius = 1)
```



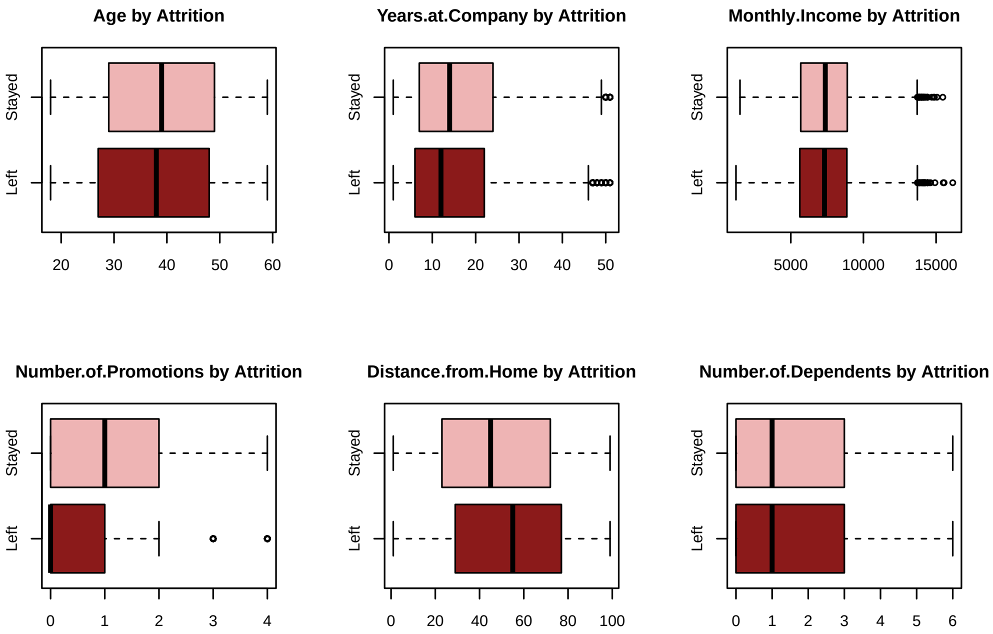
7. Outlier Analysis

The second boxplots show the relationship between attrition and various numerical variables: There are more outliers for longer tenures among employees who stayed, suggesting that employees who remain with the company for extended periods are less likely to leave. Monthly income distributions are almost equal for both groups, but there are more high-income outliers among those who left.This suggests that while average income may not differ much between those who left and stayed, higher earners are more likely to leave. Attrition appears to be more closely linked with tenure at the company and, to some extent, with the distribution of high-income outliers. Shorter tenures are associated with higher attrition rates, and while income levels are generally similar for those who leave and stay, the presence of high-income outliers among those who leave suggests that other factors, such as job satisfaction or career development opportunities, might be influencing their decision to leave.

```r
# Boxplot of outliers
par(mfrow = c(2, 3))
for (num_var in numeric_var_names) {
    boxplot(data[[num_var]], main = paste(num_var), xlab = "", col = "firebrick4",
        horizontal = TRUE)
}
```



```
# Boxplots of numeric variables by Attrition
cat_var <- "Attrition"
par(mfrow = c(2, 3), mar = c(5, 3, 3, 2) + 0.1, cex.main = 1, cex.lab = 0.9,
    cex.axis = 0.9)
for (num_var in numeric_var_names) {
    boxplot(data[[num_var]] ~ data[[cat_var]], main = paste(num_var, "by Attrition"),
        xlab = "", ylab = "", col = c("firebrick4", "rosybrown2"), horizontal = TRUE,
        names = c("Left", "Stayed"))
}
```



```r
# Function to count outliers using IQR
count_outliers <- function(x) {
    Q1 <- quantile(x, 0.25, na.rm = TRUE)
    Q3 <- quantile(x, 0.75, na.rm = TRUE)
    IQR <- Q3 - Q1
    outliers <- x[x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR)]
    return(length(outliers))
}
# Identify and count outliers for each numeric variable
outlier_counts <- list()
for (var in numeric_var_names) {
    outlier_count <- count_outliers(data[[var]])
    outlier_counts[[var]] <- outlier_count
    cat("\nNumber of outliers in", var, ":", outlier_count, "\n")
}
```

```
Number of outliers in Age : 0

Number of outliers in Years.at.Company : 338

Number of outliers in Monthly.Income : 65

Number of outliers in Number.of.Promotions : 0

Number of outliers in Distance.from.Home : 0

Number of outliers in Number.of.Dependents : 0
```

```r
# Function to identify outliers using IQR
identify_outliers <- function(x) {
    Q1 <- quantile(x, 0.25, na.rm = TRUE)
    Q3 <- quantile(x, 0.75, na.rm = TRUE)
    IQR <- Q3 - Q1
    outliers <- which(x < (Q1 - 1.5 * IQR) | x > (Q3 + 1.5 * IQR))  # Return indices
    return(outliers)
}
```

In methodology we used to identify and remove outlier employs the Interquartile Range (IQR).The IQR method was used because it is simple and effective. It focuses on the central 50% of the data, making it less sensitive to extreme values. It works by calculating the range between the first quartile (Q1) and third quartile (Q3) and identifying outliers as values falling below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR.

Functions are used to count outliers and identify their indices for each numeric variable, subsequently removing rows with outliers to create a non-outlier dataset. Despite this rigorous process, removing outliers did not enhance the model's performance and instead decreased its accuracy. This indicates that the outliers contain valuable information crucial for the model's predictive power. Consequently, it was decided to retain the outliers in the dataset, preserving model accuracy and ensuring effective predictive performance.

```r
# Identify and show outliers for each numeric variable and combine them
# into single vector
outliers_list <- list()
all_outlier_indices <- c()
for (var in numeric_var_names) {
    outliers <- identify_outliers(data[[var]])
    outliers_list[[var]] <- outliers
    all_outlier_indices <- c(all_outlier_indices, outliers)
    cat("\nOutliers in", var, ":\n")
    print(outliers)
}

all_outlier_indices <- unique(all_outlier_indices)
```

```r
# Remove rows with outliers
data_nonoutlier <- data[-all_outlier_indices, ]
# Print the number of rows removed and the resulting data
cat("\nTotal number of rows removed:", length(all_outlier_indices), "\n")
```

```
Total number of rows removed: 403
```

```r
cat("Original dataset rows:", nrow(data), "\n")
```

```
Original dataset rows: 74498
```

```
cat("Non-outlier dataset rows:", nrow(data_nonoutlier), "\n")
```

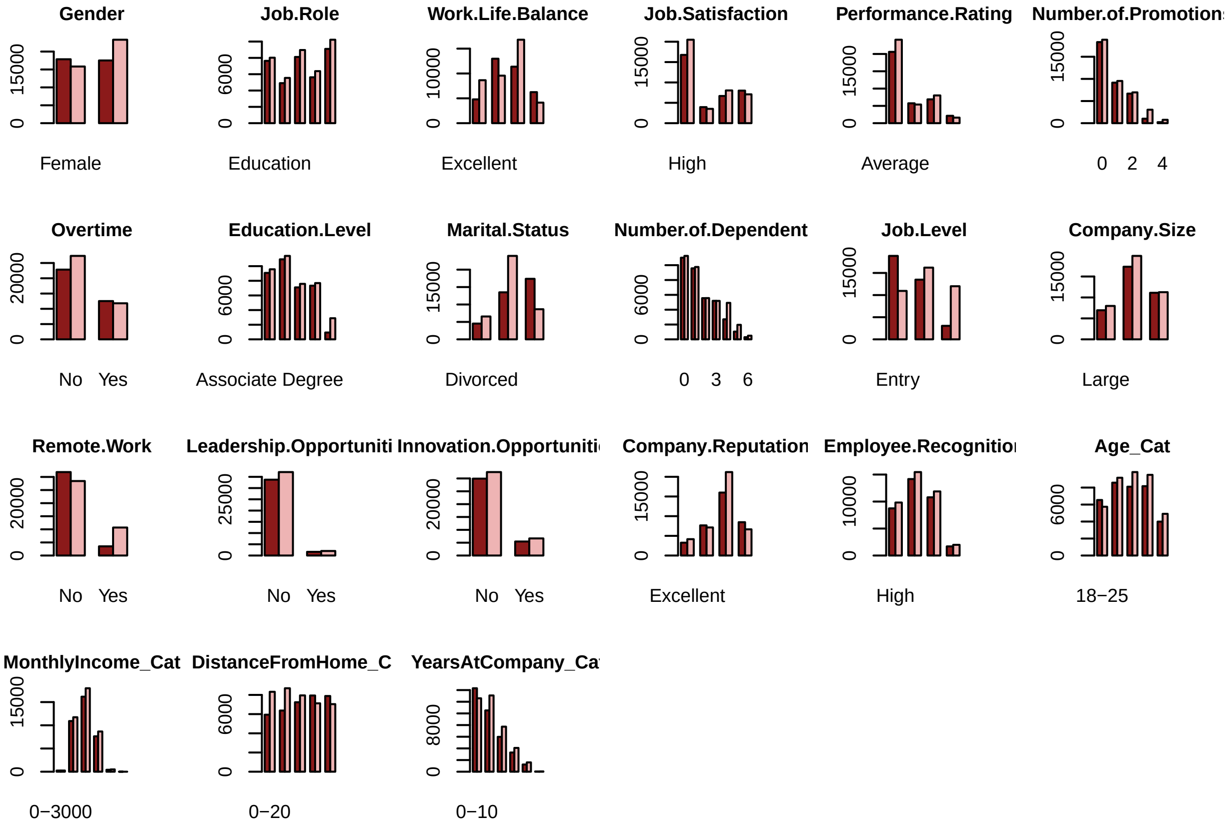
```
Non-outlier dataset rows: 74095
```

8. Transforming Numerical Variables into Categorical Variables

This methodology effectively transforms continuous numeric variables into categorical bins, facilitating better visualization and comparative analysis of attrition. The use of bar charts to compare the distributions of these categories between employees who left and those who stayed provides clear insights into how different factors may influence employee turnover.

Employees who do not work overtime tend to stay more than those who leave, indicating that excessive work hours may contribute to higher attrition rates. Married employees are less likely to leave compared to single employees, suggesting that marital status influences retention. Remote work and higher job levels are associated with lower attrition, emphasizing the value of flexibility and career growth opportunities. A good company reputation and recognition also correlate with lower attrition, highlighting the importance of a positive work environment and employee appreciation. The distribution across education levels, number of dependents, and distance from home shows similar patterns for both those who left and stayed, suggesting these factors have a moderate impact on turnover. Understanding these patterns helps in developing targeted strategies to improve employee satisfaction and reduce turnover rates.

```
# Define bins for numerical variables
data_detailed$Age_Cat <- cut(data_detailed$Age, breaks = c(18, 25, 35, 45,
    55, 60), labels = c("18-25", "25-35", "35-45", "45-55", "55-60"), right = FALSE)
data_detailed$MonthlyIncome_Cat <- cut(data_detailed$Monthly.Income, breaks = c(0,
    3000, 6000, 9000, 12000, 15000, 18000), labels = c("0-3000", "3000-6000",
    "6000-9000", "9000-12000", "12000-15000", "15000+"), right = FALSE)
data_detailed$DistanceFromHome_Cat <- cut(data_detailed$Distance.from.Home,
    breaks = c(0, 20, 40, 60, 80, 100), labels = c("0-20", "20-40", "40-60",
        "60-80", "80-100"), right = FALSE)
data_detailed$YearsAtCompany_Cat <- cut(data_detailed$Years.at.Company, breaks = c(0,
    10, 20, 30, 40, 50, 60), labels = c("0-10", "10-20", "20-30", "30-40",
    "40-50", "50-60"), right = FALSE)

# Exclude numerical columns
exclude_columns <- c("Age", "Monthly.Income", "Distance.from.Home", "Years.at.Company",
    "Attrition")
exclude_columns1 <- c("Age", "Monthly.Income", "Distance.from.Home", "Years.at.Company")
# Create data for correlation matrix
data_corr <- data_detailed[, !names(data_detailed) %in% exclude_columns1]
data_corr <- data_corr[, c(setdiff(names(data_corr), "Attrition"), "Attrition")]

# Plotting with target feature after transform numerical variables into
# categorical
par(mfrow = c(4, 6), mar = c(3, 3, 2, 2) + 0.1)
for (col in setdiff(names(data_detailed), exclude_columns)) {
    table_left <- table(data_detailed[data_detailed$Attrition == "Left",
        col])
    table_stayed <- table(data_detailed[data_detailed$Attrition == "Stayed",
        col])
    barplot(rbind(table_left, table_stayed), beside = TRUE, main = paste(col),
        xlab = "", col = c("firebrick4", "rosybrown2"), cex.names = 0.9,
        cex.main = 0.9, cex.axis = 0.9)
}
```



9. Checking Correlation

In order to look at the correlation between variables and their effect on Attrition, ordinal and label encoding are used on categorical variables, followed by generating a correlation matrix and visualizing it.

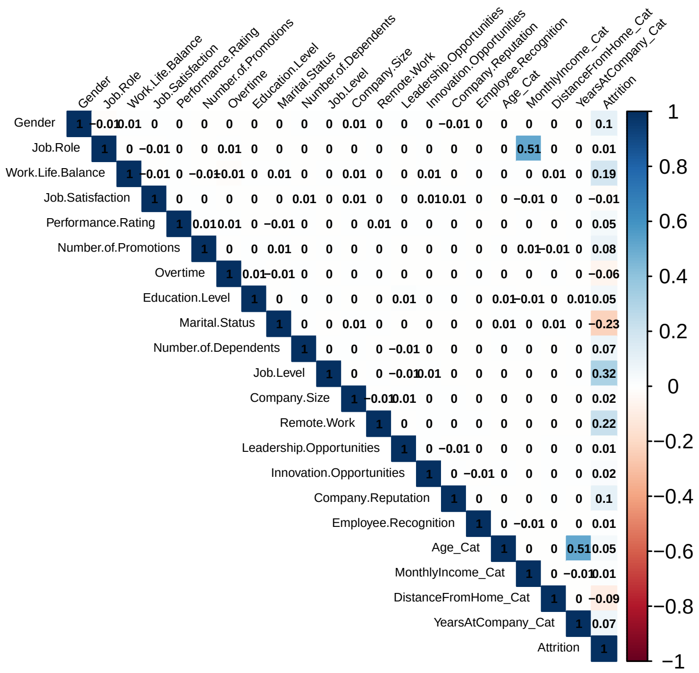
Ordinal mapping converts variables with inherent order, such as Work-Life Balance and Job Satisfaction, into numeric values to preserve their ordinal nature, enabling meaningful numerical analysis. Label encoding is applied to unordered categorical variables, converting them into numeric values necessary for numerical analyses.

Additionally, numeric variables were converted to categorical bins previous code and then encoded to avoid issues related to large or small numeric values and to standardize the data format, ensuring consistency in the correlation analysis. By encoding the data, we aim to avoid disproportionate influence from large or small values and ensure each variable contributes equally to the analysis. The correlation matrix calculates pairwise correlation coefficients, helping to understand the linear relationships between variables. The correlation plot visually represents these relationships, using colors and numbers to indicate the strength and direction of correlations.So this process prepares the dataset for sophisticated analysis, aids in feature selection, identifies multicollinearity issues, and providing insights into complex variable relationships.

However, it is important to note that this method may not capture the full complexity of relationships that exist in the original numeric data.

```
# Ordinal mappings
balance.map <- c(Poor = 1, Fair = 2, Good = 3, Excellent = 4)
```

```r
data_corr$Work.Life.Balance <- balance.map[as.character(data_corr$Work.Life.Balance)]

satisfaction.map <- c(Low = 1, Medium = 2, High = 3, `Very High` = 4)
data_corr$Job.Satisfaction <- satisfaction.map[as.character(data_corr$Job.Satisfaction)]

performance.map <- c(Low = 1, `Below Average` = 2, Average = 3, High = 4)
data_corr$Performance.Rating <-
 ↪  performance.map[as.character(data_corr$Performance.Rating)]

level.map <- c(Entry = 1, Mid = 2, Senior = 3)
data_corr$Job.Level <- level.map[as.character(data_corr$Job.Level)]

reputation.map <- c(Poor = 1, Fair = 2, Good = 3, Excellent = 4)
data_corr$Company.Reputation <-
 ↪  reputation.map[as.character(data_corr$Company.Reputation)]

recognition.map <- c(Low = 1, Medium = 2, High = 3, `Very High` = 4)
data_corr$Employee.Recognition <-
 ↪  recognition.map[as.character(data_corr$Employee.Recognition)]

size.map <- c(Small = 1, Medium = 2, Large = 3)
data_corr$Company.Size <- size.map[as.character(data_corr$Company.Size)]

# Perform label encoding for the columns that are not mapped ordinally
categoric_vars_enc <- sapply(data_corr, function(x) is.factor(x) || is.character(x))
for (var in names(data_corr)[categoric_vars_enc]) {
    if (!(var %in% c("Work.Life.Balance", "Job.Satisfaction", "Performance.Rating",
        "Job.Level", "Company.Reputation", "Employee.Recognition", "Company.Size"))) {
        data_corr[[var]] <- as.numeric(factor(data_corr[[var]]))
    }
}
# Define cor matrix
cor_matrix <- cor(data_corr)
# Correlation plot
corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black",
    tl.srt = 45, addCoef.col = "black", number.cex = 0.5, tl.cex = 0.5)
```



There is positive correlation between job role and monthly income (0.51), indicating that higher job roles (Media 4,Technology 5) are associated with higher incomes. Work-life balance shows a positive correlation with attrition (0.19), suggesting that employees with better work-life balance are more likely to stay.This emphasizes the importance of promoting a healthy work-life balance to reduce turnover rates. There is also a notable negative correlation between marital status and attrition (-0.23), indicating that married employees are less likely to leave the company. Remote work has a moderate negative correlation with attrition (-0.22), suggesting that employees who can work remotely are more likely to stay.The flexibility offered by remote work arrangements can be a significant factor in retaining employees, especially in the current global context where remote work has become more prevalent. Job level also shows a notable negative correlation with attrition (-0.32), implying that higher job levels are associated with lower attrition rates.This could be due to increased job satisfaction, responsibility, and rewards that come with higher-level positions, making employees more likely to stay. Additionally, there is a weak negative correlation between distance from home and attrition (-0.09), and a very weak positive correlation between years at the company and attrition (0.07). These insights highlight which factors are more strongly related to employee turnover, providing valuable information for developing targeted retention strategies.

Data Preparation

After completing the data analysis steps, it is necessary to prepare the data for model development.

Handling Categorical Features

In order to use the categorical features in the model, we need to convert categorical features to numeric (ordinal or nominal) representations.

```r
# Ordinal mappings:
data$Work.Life.Balance <- balance.map[as.numeric(data$Work.Life.Balance)]
data$Job.Satisfaction <- satisfaction.map[as.numeric(data$Job.Satisfaction)]
data$Performance.Rating <- performance.map[as.numeric(data$Performance.Rating)]
```

```r
data$Job.Level <- level.map[as.numeric(data$Job.Level)]
data$Company.Reputation <- reputation.map[as.numeric(data$Company.Reputation)]
data$Employee.Recognition <- recognition.map[as.numeric(data$Employee.Recognition)]
data$Company.Size <- size.map[as.numeric(data$Company.Size)]

# Nominal mappings: Create dummy variables for nominal data
data_numeric <- model.matrix(~., data = data)
# Convert the resulting matrix back to a data frame
data_numeric <- as.data.frame(data_numeric)[, -1]  # -1 to remove the intercept
```

Normalization

```r
normalize <- function(x) {
    return((x - min(x))/(max(x) - min(x)))
}

data_normalized <- as.data.frame(lapply(data_numeric, normalize))
```

Train-Test-Split

```r
set.seed(123)
trainIndex <- sample(1:nrow(data_normalized), 0.8 * nrow(data_normalized))

# 80% of data is used for training
train.data <- data_normalized[trainIndex, ]

# 20% of data is used for testing
test.data <- data_normalized[-trainIndex, ]
```

```r
# Splitting data into features and target:
X.train <- train.data[, !(colnames(data_normalized) %in% c("Employee.ID",
    "AttritionStayed"))]
y.train <- train.data$AttritionStayed

X.test <- test.data[, !(colnames(data_normalized) %in% c("Employee.ID",
 ↪  "AttritionStayed"))]
y.test <- test.data$AttritionStayed
```

Now, we can split the dataset for modelling.

Before moving to modelling step, it is beneficial to check the dimensions and balance of the datasets.

- Number of samples in train data: 59598
- Number of samples in test data: 14900

- Proportion of stayed employees for train data:

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

```
y.train
        0         1 
0.4733213 0.5266787 
```

- Proportion of stayed employees for test data:

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

```
y.test
       0        1 
0.480604 0.519396 
```

We can observe that the train and test datasets are balanced within themselves. Also the train data is representative of test data.

Predictive Classification Models

Predictive classification models are a type of machine learning algorithm used to predict the category or class label of new, unseen instances based on historical data. These models are trained using a labelled dataset where the input features (independent variables) are associated with known class labels (dependent variable). The goal of the model is to learn the relationship between the features and the class labels so that it can accurately classify new data points into one of the predefined categories.

In this project we aim to find the risk of an employee leaving the company (class 0) and the factors affecting employee retention. So we will develop several classification models and examine their performances.

Logistic Regression

The logistic regression model estimates the odds of the dependent variable occurring and applies the logit (log-odds) transformation to express this relationship.

Basic Logistic Classifier

```
# First of all we check the model statistics with all the features
glm.FULL <- glm(y.train ~ ., data = X.train, family = binomial)
summary(glm.FULL)
```

```

Call:
glm(formula = y.train ~ ., family = binomial, data = X.train)

Coefficients:
```

```
                                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)                     -0.553730   0.064437  -8.593  < 2e-16 ***
Age                              0.229757   0.039315   5.844 5.10e-09 ***
GenderMale                       0.543694   0.019788  27.476  < 2e-16 ***
Years.at.Company                 0.654574   0.051922  12.607  < 2e-16 ***
Job.RoleFinance                  0.109910   0.046309   2.373 0.017625 *  
Job.RoleHealthcare               0.061875   0.040506   1.528 0.126628    
Job.RoleMedia                    0.106027   0.034483   3.075 0.002107 ** 
Job.RoleTechnology               0.098439   0.046401   2.121 0.033880 *  
Monthly.Income                   0.018438   0.117791   0.157 0.875614    
Work.Life.Balance               -0.565409   0.031321 -18.052  < 2e-16 ***
Job.Satisfaction                -0.365624   0.024033 -15.213  < 2e-16 ***
Performance.Rating              -0.289214   0.030814  -9.386  < 2e-16 ***
Number.of.Promotions             0.928255   0.039924  23.250  < 2e-16 ***
OvertimeYes                     -0.336407   0.020902 -16.095  < 2e-16 ***
Distance.from.Home              -0.886487   0.033969 -26.097  < 2e-16 ***
Education.LevelBachelor.s.Degree -0.035818   0.026276  -1.363 0.172835    
Education.LevelHigh.School       0.001582   0.029370   0.054 0.957050    
Education.LevelMaster.s.Degree  -0.006862   0.029087  -0.236 0.813509    
Education.LevelPhD               1.506973   0.053149  28.354  < 2e-16 ***
Marital.StatusMarried            0.257001   0.028268   9.092  < 2e-16 ***
Marital.StatusSingle            -1.409863   0.030515 -46.203  < 2e-16 ***
Number.of.Dependents             0.842765   0.038204  22.060  < 2e-16 ***
Job.Level                        2.292477   0.028636  80.056  < 2e-16 ***
Company.Size                    -0.193325   0.027989  -6.907 4.94e-12 ***
Remote.WorkYes                   1.635385   0.027848  58.726  < 2e-16 ***
Leadership.OpportunitiesYes      0.162824   0.045057   3.614 0.000302 ***
Innovation.OpportunitiesYes      0.127904   0.026574   4.813 1.49e-06 ***
Company.Reputation              -0.342783   0.033757 -10.154  < 2e-16 ***
Employee.Recognition            -0.003910   0.034468  -0.113 0.909677    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82451  on 59597  degrees of freedom
Residual deviance: 62455  on 59569  degrees of freedom
AIC: 62513

Number of Fisher Scoring iterations: 4
```

The above model statistics indicate that p-value of Employee Recognition is above 0.5 indicating that this feature is insignificant to the results. Additionally, some Job.Roles and Monthly. Income also have high p-values indicating that their effect on Attrition is less significant compared to other features. However, for now we would like to keep all the features in the model and apply feature selection later.

In order to understand how well the model fits the data we can make use of R^2 statistics. R^2 provides an indication of how well the independent variables in the model explain the variability of the dependent variable. A higher R^2 value indicates a better fit of the model to the data. The formula for R^2 is:

$$R^2 = 1 - \frac{RSS}{ESS}$$

Where:

- RSS is the sum of squares of the residuals (the differences between observed and predicted values), i.e. the deviance of the fitted model
- ESS is the total sum of squares due to regression (the differences between the observed values and the mean of the observed values)

```
R2 <- 1 - (summary(glm.FULL)$deviance/summary(glm.FULL)$null.deviance)
R2
```

```
[1] 0.2425186
```

With the full model the value of R^2 0.2425186 indicates that approximately 24.2518642% of the variance in the target can be explained by the features in the model. Since 24.2518642% is relatively low, it suggests that the model is not capturing much of the underlying pattern in the data.

Multicollinearity can be a reason for a low R^2 value, as it can make it difficult to determine the individual effect of each predictor on the target. Calculating the Variance Inflation Factor (VIF) can help to check for multicollinearity among the features.

```
vif.FULL <- data.frame(features = names(vif(glm.FULL)), VIF = vif(glm.FULL),
    row.names = NULL)
vif.FULL[order(-vif.FULL$VIF), ]
```

```
                           features      VIF
7                 Job.RoleTechnology 4.317840
5                 Job.RoleHealthcare 3.028354
8                     Monthly.Income 3.014476
4                    Job.RoleFinance 2.699408
20             Marital.StatusSingle 2.164088
19            Marital.StatusMarried 2.085912
6                      Job.RoleMedia 1.682568
15 Education.LevelBachelor.s.Degree 1.528098
17   Education.LevelMaster.s.Degree 1.434283
16       Education.LevelHigh.School 1.424799
3                   Years.at.Company 1.405468
1                                Age 1.403168
18               Education.LevelPhD 1.124677
22                         Job.Level 1.098028
24                     Remote.WorkYes 1.060436
2                         GenderMale 1.013155
14                Distance.from.Home 1.011754
12              Number.of.Promotions 1.009919
21              Number.of.Dependents 1.009231
9                  Work.Life.Balance 1.007059
10                  Job.Satisfaction 1.005314
13                       OvertimeYes 1.005171
```

```
27               Company.Reputation 1.002671
11               Performance.Rating 1.001850
23                     Company.Size 1.001579
25      Leadership.OpportunitiesYes 1.000932
28             Employee.Recognition 1.000568
26      Innovation.OpportunitiesYes 1.000566
```

A VIF value of 1 indicates no correlation, values between 1 and 5 indicate moderate correlation and values above 5 suggest significant multicollinearity, which can lead to unreliable coefficient estimates. Most variables have VIF values close to 1, indicating very low or no multicollinearity. A few variables have VIF values between 1 and 5, suggesting moderate multicollinearity, which may not pose serious issues but should be monitored. These variables include Job.RoleTechnology, Job.RoleHealthcare, Monthly.Income, Job.RoleFinance, Marital.StatusSingle and Marital.StatusMarried. These are mostly dummy features of nominal variables and dummy variables are often correlated because they represent categories of the same nominal variable.

Logistic Regression with Backward Stepwise Search

Backward variable selection is a greedy search algorithm used to develop a predictive model by iteratively removing the least significant features The goal is to find the best subset of features that contribute to the model while eliminating those that do not improve its performance.

We can use the step() function to perform backward stepwise search. As a regularization criteria we can either use BIC or AIC. But BIC statistic generally places a heavier penalty on models with many variables, and hence results in the selection of smaller models than AIC.

```r
# Backward Stepwise Search with BIC statistics
glm.BACKWARD <- step(glm.FULL, direction = "backward", k = log(nrow(X.train)),
    trace = 0, steps = 100)
summary(glm.BACKWARD)
```

```

Call:
glm(formula = y.train ~ Age + GenderMale + Years.at.Company +
    Work.Life.Balance + Job.Satisfaction + Performance.Rating +
    Number.of.Promotions + OvertimeYes + Distance.from.Home +
    Education.LevelPhD + Marital.StatusMarried + Marital.StatusSingle +
    Number.of.Dependents + Job.Level + Company.Size + Remote.WorkYes +
    Leadership.OpportunitiesYes + Innovation.OpportunitiesYes +
    Company.Reputation, family = binomial, data = X.train)

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -0.48883    0.05177  -9.442  < 2e-16 ***
Age                          0.22990    0.03930   5.849 4.94e-09 ***
GenderMale                   0.54243    0.01978  27.423  < 2e-16 ***
Years.at.Company             0.65402    0.05191  12.599  < 2e-16 ***
Work.Life.Balance           -0.56426    0.03131 -18.022  < 2e-16 ***
Job.Satisfaction            -0.36546    0.02403 -15.210  < 2e-16 ***
Performance.Rating          -0.28983    0.03080  -9.409  < 2e-16 ***
Number.of.Promotions         0.92821    0.03991  23.256  < 2e-16 ***
```

```
OvertimeYes                   -0.33570    0.02089 -16.068  < 2e-16 ***
Distance.from.Home            -0.88551    0.03396 -26.076  < 2e-16 ***
Education.LevelPhD             1.51895    0.05049  30.086  < 2e-16 ***
Marital.StatusMarried          0.25788    0.02826   9.126  < 2e-16 ***
Marital.StatusSingle          -1.40864    0.03050 -46.183  < 2e-16 ***
Number.of.Dependents           0.84234    0.03820  22.052  < 2e-16 ***
Job.Level                      2.29026    0.02862  80.023  < 2e-16 ***
Company.Size                  -0.19248    0.02798  -6.879 6.02e-12 ***
Remote.WorkYes                 1.63481    0.02784  58.729  < 2e-16 ***
Leadership.OpportunitiesYes    0.16201    0.04504   3.597 0.000322 ***
Innovation.OpportunitiesYes    0.12815    0.02657   4.823 1.41e-06 ***
Company.Reputation            -0.34247    0.03374 -10.149  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 82451  on 59597  degrees of freedom
Residual deviance: 62476  on 59578  degrees of freedom
AIC: 62516

Number of Fisher Scoring iterations: 4
```

```
vif.BACKWARD <- data.frame(features = names(vif(glm.BACKWARD)), VIF = vif(glm.BACKWARD),
    row.names = NULL)
head(vif.BACKWARD[order(-vif.BACKWARD$VIF), ], 10)
```

```
                features      VIF
12  Marital.StatusSingle 2.163022
11 Marital.StatusMarried 2.085498
3       Years.at.Company 1.405299
1                    Age 1.402928
14             Job.Level 1.097342
16        Remote.WorkYes 1.060083
10    Education.LevelPhD 1.015134
2             GenderMale 1.012783
9     Distance.from.Home 1.011547
7   Number.of.Promotions 1.009742
```

It looks like backward stepwise search dropped “Job.RoleFinance”, “Job.RoleHealthcare”, “Job.RoleTechnology” and “Employee.Recognition” from the model. These features were also the ones with highest p-values so it seems logical. But oddly the model kept the “Job.RoleMedia” feature so this tells us that having a job role in Media may be more significant to employee Attrition than other job roles for this dataset.

With backward feature selection we were able to decrease the VIF scores, p-values and AIC score. Although R^2 slightly decreased to 0.2422558, this is expectable since the number of features in the model decreased. But most importantly we decreased the variance inflation factor of the model so the model is more stable.

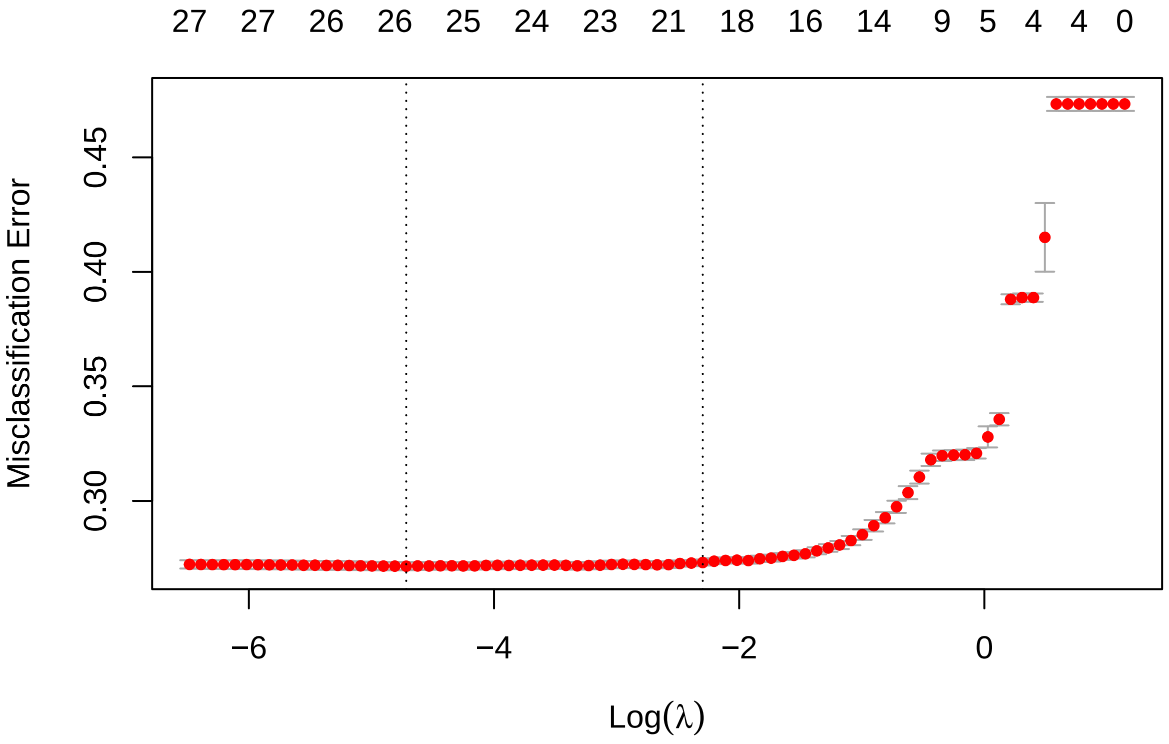
Logistic Regression with Shrinkage Method

Shrinkage methods are techniques used in regression analysis to prevent overfitting by introducing a penalty for large coefficients. These methods "shrink" the coefficients towards zero, effectively reducing their variance and, in turn, enhancing the model's generalizability. Two most used shrinkage methods are Ridge Regression and Lasso Regression.

- Ridge Regression adds a penalty to the regression model that shrinks all the coefficients, and is useful when we want to keep all features in the model.
- Lasso Regression introduces a penalty that can shrink some coefficients to zero. This method is useful for feature selection, yielding sparse models by retaining only the most important features.
- Elastic Net is a hybrid regularization technique that includes both the Ridge and Lasso penalties, allowing for variable selection (like Lasso) and handling multicollinearity (like Ridge). It aims to incorporate the strengths of both methods, providing a more flexible approach.

Choosing the value of $lambda$ (the tuning parameter) is crucial because it determines the strength of the penalty applied to the coefficients. We can choose the best value of $lambda$ using k-fold cross-validation method.

```r
# Elastic Net Regression with alpha=0.05 (more weight on Ridge Reg.)
set.seed(123)

glm.ELNET <- cv.glmnet(as.matrix(X.train), y.train, alpha = 0.05, family = "binomial",
    type.measure = "class")
best.lamda <- glm.ELNET$lambda.min
plot(glm.ELNET)
```

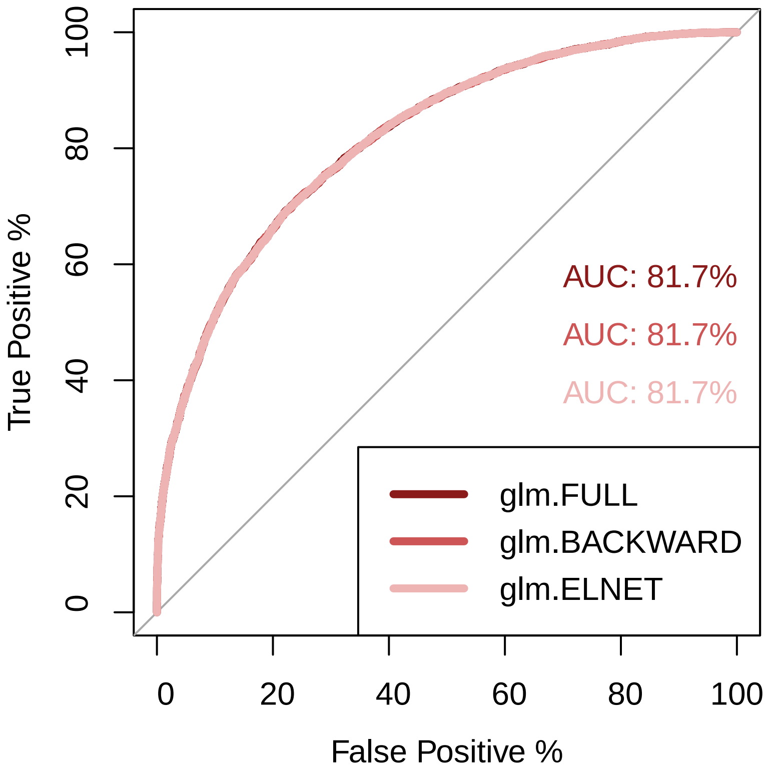


Comparison of Logistic Classifiers

For Logistic Regression we defined 3 Logistic classifiers. In order to identify the best model we can compare their performance on the test sets to see how well they captured the underlying pattern of the data and their ability to generalize to new data.

```
# 1. Basic Logistic Classifier
glm.FULL.predict <- predict(glm.FULL, newdata = X.test, type = "response")

# 2. Feature Selection with Backward Stepwise Search
glm.BACKWARD.predict <- predict(glm.BACKWARD, newdata = X.test, type = "response")

# 3. Elastic Net Shrinkage Method
glm.ELNET.predict <- predict(glm.ELNET, newx = as.matrix(X.test), type = "response",
    s = best.lamda)
```

We can use ROC curve and AUC values to compare the models. The ROC curve is a tool for assessing the performance of binary classification models, plotting true positive rate against false positive rate at various thresholds. The Area Under the Curve (AUC) provides a measure of the model’s ability to predict the target values, with higher values indicating better performance.



Although the AUC values are same, to better analyse the difference between the model performances we can calculate and compare the evaluation metrics.

```
# Function to evaluate prediction models at different thresholds
evaluate.model.performance <- function(predictions, y.test, thresholds = NULL) {
    output.list <- list()

    if (!is.null(thresholds)) {
        # Looping through each threshold to generate predictions and
        # store in output.list
        for (threshold in thresholds) {
            output <- ifelse(predictions > threshold, 1, 0)
            output.list[[as.character(threshold)]] <- output
        }
    } else {
        # Use the given predictions directly if no thresholds are
        # provided
        output.list[["No Threshold"]] <- predictions
        thresholds <- c("No Threshold")  # Create a single 'threshold' for consistent
 ↪  processing
    }

    # Initialize a data frame to store the evaluation metrics for each
    # threshold
    results <- data.frame(Threshold = character(length(thresholds)), Accuracy =
 ↪  numeric(length(thresholds)),
        F1.Score = numeric(length(thresholds)), Precision = numeric(length(thresholds)),
        Recall = numeric(length(thresholds)))

    # Compute evaluation metrics for each threshold
    for (i in 1:length(thresholds)) {
        threshold <- thresholds[i]
        predict.output <- output.list[[as.character(threshold)]]

        # Store results in the results dataframe
        results[i, "Threshold"] = threshold
        results[i, "Accuracy"] = mean(predict.output == y.test)
        results[i, "F1.Score"] = (2 * sum((predict.output == 1 & y.test ==
            1))/(2 * sum((predict.output == 1 & y.test == 1)) + sum(predict.output ==
            1 & y.test == 0) + sum(predict.output == 0 & y.test == 1)))
        results[i, "Precision"] = sum(predict.output == 1 & y.test ==
 ↪  1)/sum(predict.output ==
            1)
        results[i, "Recall"] = sum(predict.output == 1 & y.test == 1)/sum(y.test ==
            1)
    }

    # Rounding results
    results[, c("Accuracy", "F1.Score", "Precision", "Recall")] <- round(results[,
        c("Accuracy", "F1.Score", "Precision", "Recall")], 4)

    # Conditionally adjust the output to remove the Threshold column if
    # not needed
    if (length(thresholds) == 1 && thresholds[1] == "No Threshold") {
        results <- results[, -1]  # Remove the Threshold column
        colnames(results) <- c("Accuracy", "F1.Score", "Precision", "Recall")
        kable(results, align = "c")
    } else {
        kable(results, align = "c")
    }
}
```

```
thresholds <- c(0.2, 0.3, 0.4, 0.5, 0.6)

# 1. Basic Logistic Classifier
evaluate.model.performance(glm.FULL.predict, y.test, thresholds)
```

Threshold	Accuracy	F1.Score	Precision	Recall
0.2	0.6401	0.7367	0.5941	0.9692
0.3	0.6912	0.7554	0.6417	0.9179
0.4	0.7215	0.7593	0.6889	0.8456
0.5	0.7313	0.7456	0.7337	0.7578
0.6	0.7263	0.7120	0.7850	0.6515

```
# 2. Feature Selection with Backward Stepwise Search
evaluate.model.performance(glm.BACKWARD.predict, y.test, thresholds)
```

Threshold	Accuracy	F1.Score	Precision	Recall
0.2	0.6395	0.7363	0.5937	0.9691
0.3	0.6907	0.7549	0.6414	0.9170
0.4	0.7226	0.7601	0.6901	0.8458
0.5	0.7323	0.7467	0.7343	0.7595
0.6	0.7270	0.7129	0.7855	0.6525

```
# 3. Elastic Net Shrinkage Method
evaluate.model.performance(glm.ELNET.predict, y.test, thresholds)
```

Threshold	Accuracy	F1.Score	Precision	Recall
0.2	0.6266	0.7306	0.5842	0.9748
0.3	0.6829	0.7524	0.6328	0.9278
0.4	0.7214	0.7609	0.6864	0.8536

Threshold	Accuracy	F1.Score	Precision	Recall
0.5	0.7316	0.7462	0.7332	0.7598
0.6	0.7247	0.7083	0.7876	0.6435

All of the models have the highest F1 score at threshold level 0.4. And F1 score with 0.4 threshold for glm.ELNET is higher than other models on the test set. These results indicate that the model with Elastic Net regularization is better in generalization. Therefore we can select the glm.ELNET.predict as the best prediction for logistic regression model.

LDA & QDA

Linear Discriminant Analysis (LDA) is a classification algorithm that finds a linear combination of features that best separates two or more classes. It assumes that the features follow a multivariate normal distribution with a common mean and variance for all classes. LDA and logistic classifier have the same (linear) form but LDA makes more assumptions about the underlying data then the logistic classifier. For the model we decided to exclude the features that had high VIF and p-values as analysed for logistic classifiers.

```r
# Perform LDA modelling on train data and make predictions on test data
lda.model <- lda(AttritionStayed ~ . - Job.RoleFinance - Job.RoleTechnology -
    Job.RoleHealthcare - Monthly.Income - Education.LevelBachelor.s.Degree -
    Education.LevelHigh.School - Education.LevelMaster.s.Degree - Employee.Recognition,
    data = train.data)
lda.model
```

```
Call:
lda(AttritionStayed ~ . - Job.RoleFinance - Job.RoleTechnology -
    Job.RoleHealthcare - Monthly.Income - Education.LevelBachelor.s.Degree -
    Education.LevelHigh.School - Education.LevelMaster.s.Degree -
    Employee.Recognition, data = train.data)

Prior probabilities of groups:
        0         1
0.4733213 0.5266787

Group means:
        Age GenderMale Years.at.Company Job.RoleMedia Work.Life.Balance
0 0.4852905   0.497359        0.2793094     0.1595590         0.5121888
1 0.5144369   0.595782        0.3085520     0.1630189         0.4743169
  Job.Satisfaction Performance.Rating Number.of.Promotions OvertimeYes
0        0.3880795          0.2453354            0.1869616   0.3535397
1        0.3474890          0.2227213            0.2273886   0.3008697
  Distance.from.Home Education.LevelPhD Marital.StatusMarried
0          0.5287533         0.02552377             0.3808359
1          0.4736951         0.07359266             0.6108828
  Marital.StatusSingle Number.of.Dependents Job.Level Company.Size
0            0.4904818            0.2545287 0.2765252    0.5579071
1            0.2220205            0.2943335 0.5128548    0.5430246
```

```
  Remote.WorkYes Leadership.OpportunitiesYes Innovation.OpportunitiesYes
0     0.09901095                  0.04707717                   0.1548087
1     0.27130523                  0.05234318                   0.1699959
  Company.Reputation
0          0.6126768
1          0.5927979

Coefficients of linear discriminants:
                                    LD1
Age                          0.17822014
GenderMale                   0.42558074
Years.at.Company             0.50701891
Job.RoleMedia                0.03304174
Work.Life.Balance           -0.44238318
Job.Satisfaction            -0.28451537
Performance.Rating          -0.22952580
Number.of.Promotions         0.72414409
OvertimeYes                 -0.26331157
Distance.from.Home          -0.69131133
Education.LevelPhD           1.10625973
Marital.StatusMarried        0.20731266
Marital.StatusSingle        -1.14089514
Number.of.Dependents         0.65669259
Job.Level                    1.83688216
Company.Size                -0.15697901
Remote.WorkYes               1.23658524
Leadership.OpportunitiesYes  0.12997879
Innovation.OpportunitiesYes  0.09931127
Company.Reputation          -0.26718594
```

```r
lda.predict <- predict(lda.model, newdata = test.data)$class

# Evaluate the performance of the model
evaluate.model.performance(lda.predict, y.test)
```

Accuracy	F1.Score	Precision	Recall
0.7336	0.7471	0.7368	0.7576

Quadratic Discriminant Analysis (QDA) is similar to LDA but allows for different mean and variance for each class. This results in quadratic decision boundaries.

```r
qda.model <- qda(AttritionStayed ~ . - Job.RoleFinance - Job.RoleTechnology -
    Job.RoleHealthcare - Monthly.Income - Education.LevelBachelor.s.Degree -
    Education.LevelHigh.School - Education.LevelMaster.s.Degree - Employee.Recognition,
    data = train.data)
qda.predict <- predict(qda.model, newdata = test.data)$class
qda.model
```

```
Call:
qda(AttritionStayed ~ . - Job.RoleFinance - Job.RoleTechnology - 
    Job.RoleHealthcare - Monthly.Income - Education.LevelBachelor.s.Degree - 
    Education.LevelHigh.School - Education.LevelMaster.s.Degree - 
    Employee.Recognition, data = train.data)

Prior probabilities of groups:
        0         1 
0.4733213 0.5266787 

Group means:
        Age GenderMale Years.at.Company Job.RoleMedia Work.Life.Balance
0 0.4852905   0.497359        0.2793094     0.1595590         0.5121888
1 0.5144369   0.595782        0.3085520     0.1630189         0.4743169
  Job.Satisfaction Performance.Rating Number.of.Promotions OvertimeYes
0        0.3880795          0.2453354            0.1869616   0.3535397
1        0.3474890          0.2227213            0.2273886   0.3008697
  Distance.from.Home Education.LevelPhD Marital.StatusMarried
0          0.5287533         0.02552377             0.3808359
1          0.4736951         0.07359266             0.6108828
  Marital.StatusSingle Number.of.Dependents Job.Level Company.Size
0            0.4904818            0.2545287 0.2765252    0.5579071
1            0.2220205            0.2943335 0.5128548    0.5430246
  Remote.WorkYes Leadership.OpportunitiesYes Innovation.OpportunitiesYes
0     0.09901095                  0.04707717                   0.1548087
1     0.27130523                  0.05234318                   0.1699959
  Company.Reputation
0          0.6126768
1          0.5927979
```

```
evaluate.model.performance(qda.predict, y.test)
```

Accuracy	F1.Score	Precision	Recall
0.713	0.6968	0.7718	0.6351

Model Results

Performance Metrics and Confusion Matrix

```
# Function for confusion matrix and other performance metrics
perf.measure <- function(true.values, pred.values, lab.pos = 1) {
    # compute the confusion matrix and number of units
    conf.matrix <- table(pred.values, true.values)
    n <- sum(conf.matrix)
    # force the label of positives to be a character string
    lab.pos <- as.character(lab.pos)
    # obtain the label of negatives
    lab <- rownames(conf.matrix)
    lab.neg <- lab[lab != lab.pos]
    # extract relevant quantities from the confusion matrix
    TP <- conf.matrix[lab.pos, lab.pos]
    TN <- conf.matrix[lab.neg, lab.neg]
    FP <- conf.matrix[lab.pos, lab.neg]
    FN <- conf.matrix[lab.neg, lab.pos]
    P <- TP + FN
    N <- FP + TN
    P.ast <- TP + FP
    # compute the performance measures
    OER <- (FP + FN)/n
    PPV <- TP/P.ast
    TPR <- TP/P
    F1 <- 2 * PPV * TPR/(PPV + TPR)
    TNR <- TN/N
    FPR <- FP/N
    return(list(overall.ER = OER, PPV = PPV, TPR = TPR, F1 = F1, TNR = TNR,
        FPR = FPR))
}
```

```
# Confusion matrix for glm.ELNET

# Use best threshold for prediction (0 or 1)
Threshold <- 0.4
glm.ELNET.predict.factor <- as.factor(ifelse(glm.ELNET.predict >= Threshold,
    1, 0))


# PM <- perf.measure(test.data$AttritionStayed,
# glm.ELNET.predict.factor, lab.pos = 'Stayed') PM
```

# Conclusion