Statistical Learning Project

Marlon Helbing, Nemanja Ilic, Daniele Virzì

2024-07-10

1. Introduction

A key factor in a thriving business is the retention of employees, evidenced by their long-term commitment to the company. But what strategies can a business implement to achieve this? What drives employees to leave? In this statistical analysis conducted by *Marlon Helbing*, *Nemanja Ilic*, *Daniele Virzì*, we will develop both Regression and Classification Models using a publicly available HR dataset from a company. We will optimize these models iteratively based on various metrics and present our findings, discussing the insights they reveal.

1.1 Dataset

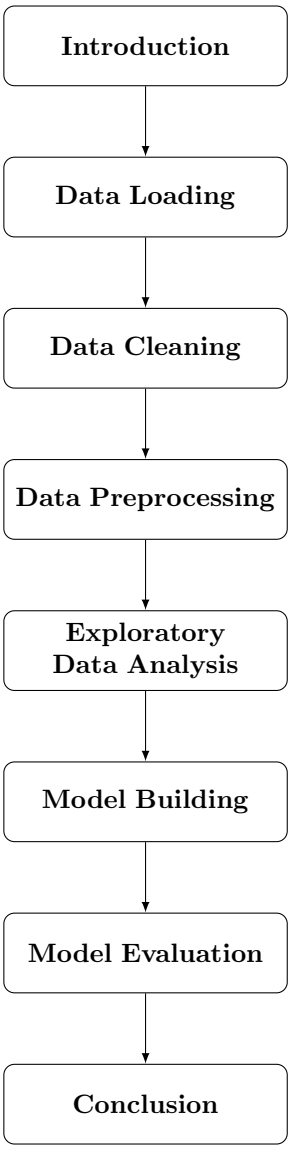
- **HR Analytics Case Study**: This dataset, sourced from *Kaggle*, contains data on around 4000 employees and includes numerous features. Since these features were spread across multiple files and not all were relevant to our study, we selected two files, `general_data` and `employee_survey_data`. The final dataset consists of 4410 observations and 27 variables. The variables are as follows:
  - `Age`: Age of the employee.
  - `Attrition`: Whether the employee has left the company or not.
  - `BusinessTravel`: Frequency of travel for the employee.
  - `Department`: Department of the employee.
  - `DistanceFromHome`: Distance of the employee's residence from the company.
  - `Education`: Education level of the employee.
  - `EducationField`: Field of education of the employee.
  - `EmployeeCount`: Employee count.
  - `EmployeeID`: Employee ID.
  - `Gender`: The gender of the employee.
  - `JobLevel`: Job level of the employee.
  - `JobRole`: Job role of the employee.
  - `MaritalStatus`: Marital status of the employee.
  - `MonthlyIncome`: Monthly income of the employee.
  - `NumCompaniesWorked`: Number of companies the employee has worked for.
  - `Over18`: Whether the employee is over 18 years old or not.
  - `PercentSalaryHike`: Percentage increase in salary.
  - `StandardHours`: Standard hours of work.
  - `StockOptionLevel`: Stock option level of the employee.
  - `TotalWorkingYears`: Total years the employee has worked.
  - `TrainingTimesLastYear`: Number of times the employee was trained last year.
  - `YearsAtCompany`: Number of years the employee has worked at the company.
  - `YearsSinceLastPromotion`: Number of years since the last promotion.
  - `YearsWithCurrManager`: Number of years the employee has worked with the current manager.
  - `EnvironmentSatisfaction`: Environment satisfaction level of the employee.
  - `JobSatisfaction`: Job satisfaction level of the employee.
  - `WorkLifeBalance`: Work-life balance level of the employee.

1.2 Project goals

Following an initial analysis of the dataset, we identified two features that stood out and were well-suited for an in-depth analysis using statistical tools:

- **Regression Model**: In order to understand the factors that influence the number of years an employee stays in the company, we analyze `YearsAtCompany` based on the available features. This analysis offers the company valuable insights into which variables most significantly impact the duration of an employee's tenure.
- **Classification Model**: In order to understand the factors that influence the attrition of employees in the company, we analyze `Attrition` based on the available features. This allows the company to identify which variables most significantly affect the employee attrition rate.

1.3 Methodology



2. Data Loading

We begin by loading the necessary libraries and the dataset into the R environment. The libraries used in this project are:

```
library(MASS) # For step, glm, lda, qda
library(e1071) # For naiveBayes
library(car) # For vif
library(corrplot) # For plotting correlation matrix
library(pROC) # For ROC curve
```

The data is obtained from `general_data.csv` and `employee_survey_data.csv`. We then merge them on the `EmployeeID` variable.

```
general_data <- read.csv("./data/general_data.csv")
employee_survey_data <- read.csv("./data/employee_survey_data.csv")
data <- merge(general_data, employee_survey_data, by = "EmployeeID")
```

3. Data Cleaning

3.1 Handling missing values

We found 111 missing values in our dataset. As these constitute at most 2.5% of the data, we opted to remove these rows entirely rather than use imputation methods, allowing us to work with an unaltered dataset.

```
missing_values <- sum(is.na(data))
missing_values
```

```
## [1] 111
```

```
data <- na.omit(data)
```

3.2 Handling duplicate rows

We found no duplicated rows in our dataset.

```
duplicates <- sum(duplicated(data))
duplicates
```

```
## [1] 0
```

3.3 Removing unnecessary features

We removed `EmployeeID`, because it is a unique identifier and does not provide any useful information for the analysis. Since `Over18`, `StandardHours`, and `EmployeeCount` had the same value for all employees, resulting in zero variance, we deleted these aswell.

```
data <- data[, !(names(data) %in% c("EmployeeID", "Over18", "StandardHours", "EmployeeCount"))]
```

4. Data Preprocessing

4.1 Encoding categorical variables

We identified the categorical variables and encoded them using the R *factor* data type.

```
data$Attrition <- factor(data$Attrition)
data$Gender <- factor(data$Gender)
```

```r
data$BusinessTravel <- factor(data$BusinessTravel)
data$JobRole <- factor(data$JobRole)
data$Department <- factor(data$Department)
data$EducationField <- factor(data$EducationField)
data$MaritalStatus <- factor(data$MaritalStatus)
data$StockOptionLevel <- factor(data$StockOptionLevel)
data$Education <- factor(data$Education)
data$JobLevel <- factor(data$JobLevel)
data$EnvironmentSatisfaction <- factor(data$EnvironmentSatisfaction)
data$JobSatisfaction <- factor(data$JobSatisfaction)
data$WorkLifeBalance <- factor(data$WorkLifeBalance)
```

4.2 Log transformation

We observed that MonthlyIncome is right-skewed and spans a wide range, including both very low and very high values. To normalize its distribution, we applied a log transformation.

```r
data$MonthlyIncome <- log(data$MonthlyIncome)
```

4.3 Final structure of the dataset

We showcase the dataset after cleaning and preprocessing.

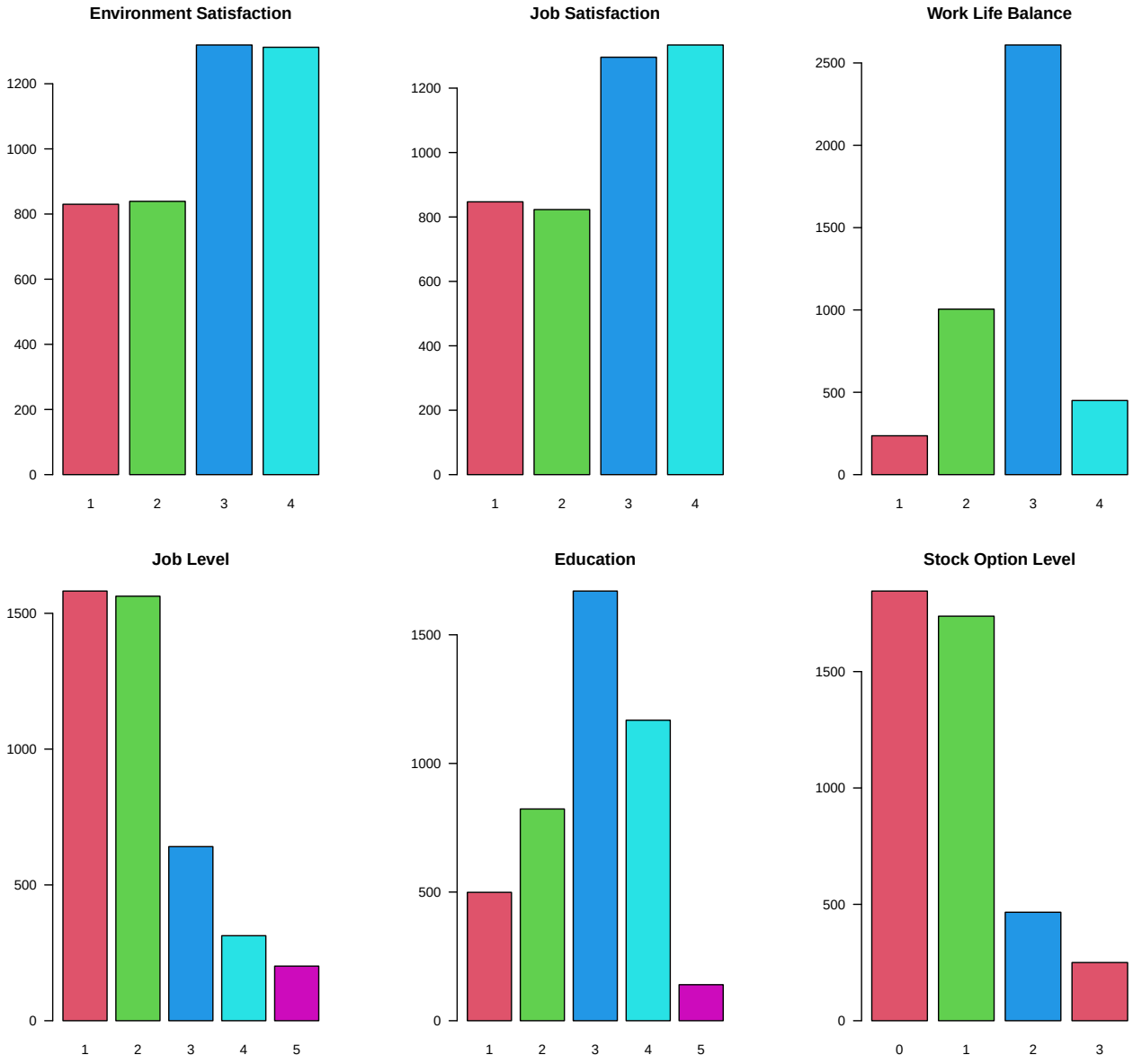
```r
str(data)
```

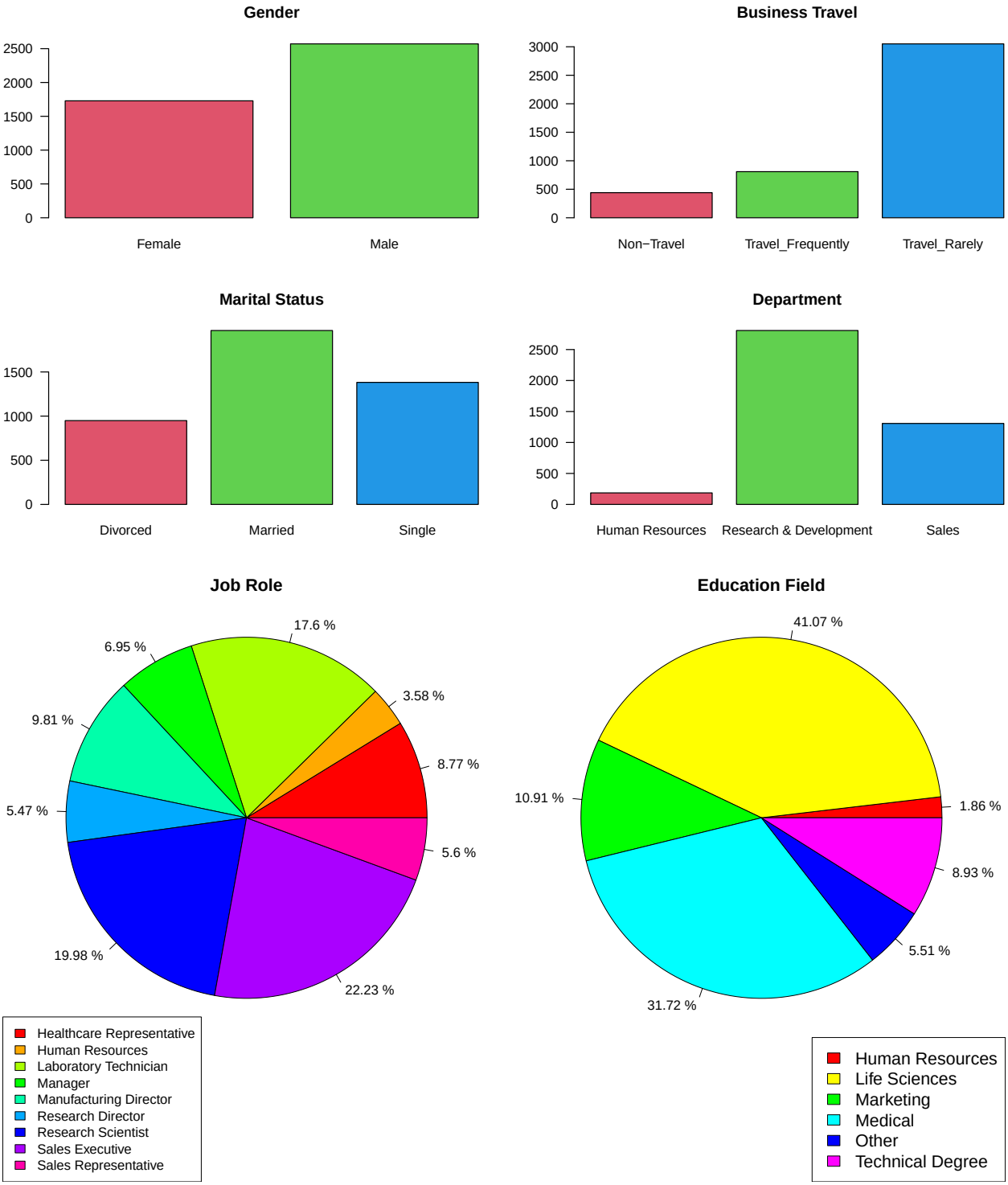
```
## 'data.frame':    4300 obs. of  23 variables:
##  $ Age                     : int  51 31 32 38 32 46 28 29 31 25 ...
##  $ Attrition               : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 2 1 1 1 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 2 2 1 3 3 3 3 3
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...
##  $ DistanceFromHome        : int  6 10 17 2 10 8 11 18 1 7 ...
##  $ Education               : Factor w/ 5 levels "1","2","3","4",..: 2 1 4 5 1 3 2 3 3 4 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 1 1 2 2 2 1 2 2 2 1 ...
##  $ JobLevel                : Factor w/ 5 levels "1","2","3","4",..: 1 1 4 3 1 4 2 2 3 4 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 1 7 8 2 8 6 8 8 3 3 .
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 2 3 2 2 3 2 3 2 2 1 ...
##  $ MonthlyIncome           : num  11.8 10.6 12.2 11.3 10.1 ...
##  $ NumCompaniesWorked      : int  1 0 1 3 4 3 2 2 0 1 ...
##  $ PercentSalaryHike       : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ StockOptionLevel        : Factor w/ 4 levels "0","1","2","3": 1 2 4 4 3 1 2 4 1 2 ...
##  $ TotalWorkingYears       : int  1 6 5 13 9 28 5 10 10 6 ...
##  $ TrainingTimesLastYear   : int  6 3 2 5 2 5 2 2 2 2 ...
##  $ YearsAtCompany          : int  1 5 5 8 6 7 0 0 9 6 ...
##  $ YearsSinceLastPromotion: int  0 1 0 7 0 7 0 0 7 1 ...
##  $ YearsWithCurrManager    : int  0 4 3 5 4 7 0 0 8 5 ...
##  $ EnvironmentSatisfaction: Factor w/ 4 levels "1","2","3","4": 3 3 2 4 4 3 1 1 2 2 ...
##  $ JobSatisfaction         : Factor w/ 4 levels "1","2","3","4": 4 2 2 4 1 2 3 2 4 1 ...
##  $ WorkLifeBalance         : Factor w/ 4 levels "1","2","3","4": 2 4 1 3 3 2 1 3 3 3 ...
```

5. Exploratory Data Analysis

To get a better overview over our categorical and numerical variables, we used various charts to visualize their distribution. Then we analyzed our two features of interest, `YearsAtCompany` and `Attrition` to understand their relationship with other variables in the dataset.

5.1 Categorical variables





We noticed a high similarity in the distributions of `JobSatisfaction` and `EnvironmentSatisfaction`. To confirm this observation, we computed the chi-squared statistic between these two variables, with the null hypothesis being that they are independent. The p-value was less than 0.1, leading us to reject the null hypothesis and conclude that the two variables are dependent. Consequently, we removed `EnvironmentSatisfaction` from the dataset.

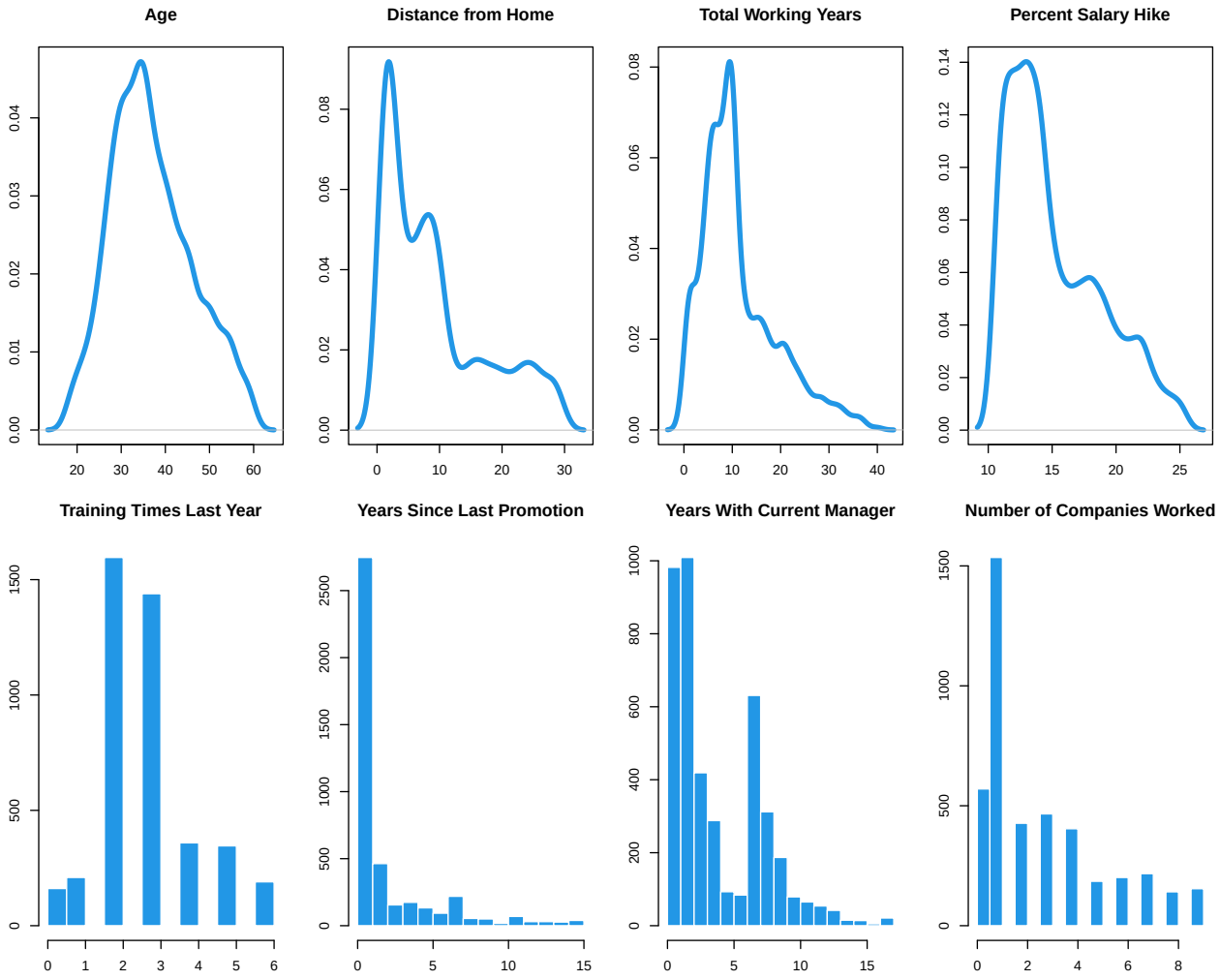
```
contingency_table <- table(data$EnvironmentSatisfaction, data$JobSatisfaction)
chi_squared <- chisq.test(contingency_table)
```

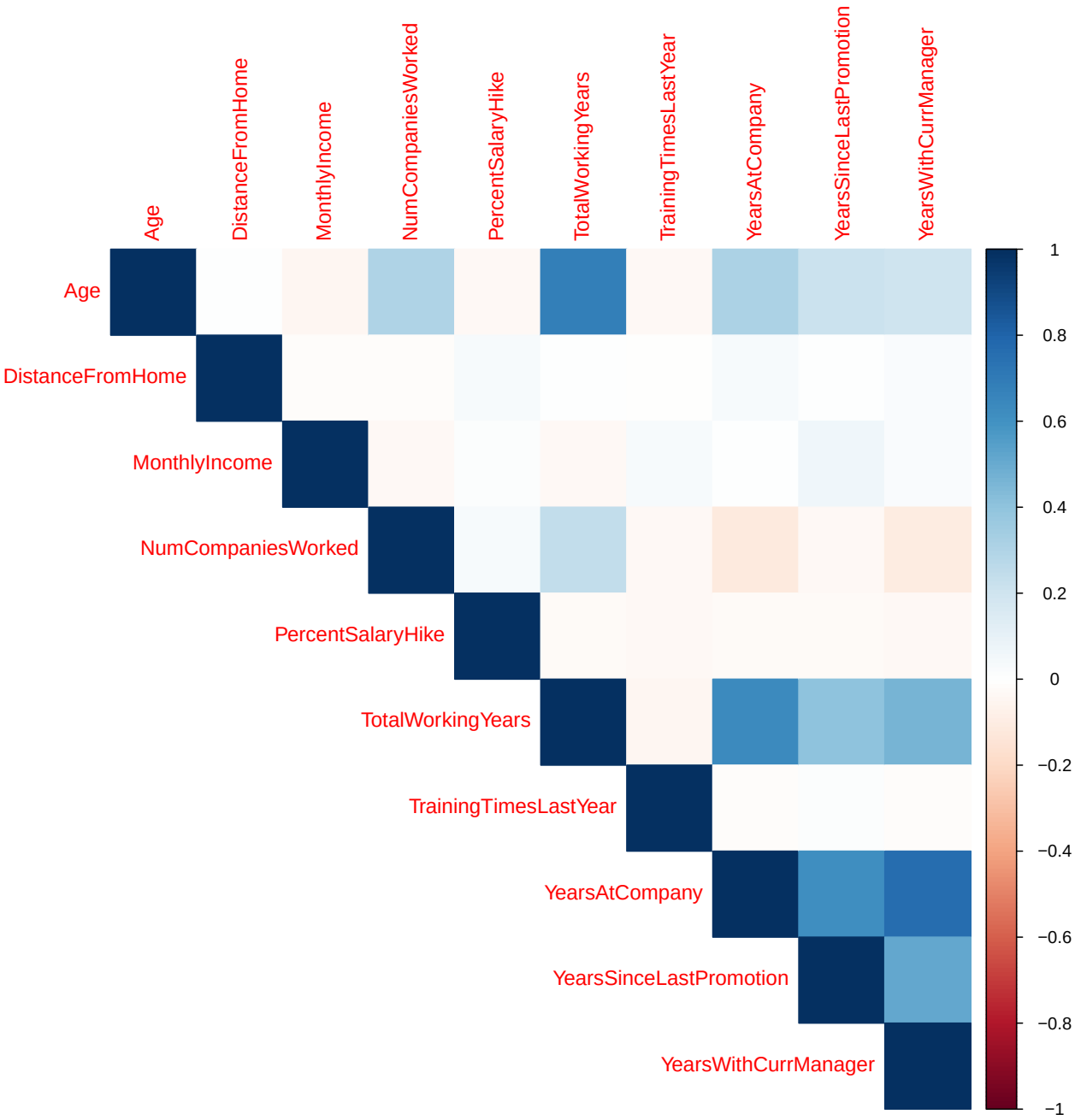
```
chi_squared
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 15.327, df = 9, p-value = 0.08235
```

```
data$EnvironmentSatisfaction <- NULL
```

5.2 Numerical variables

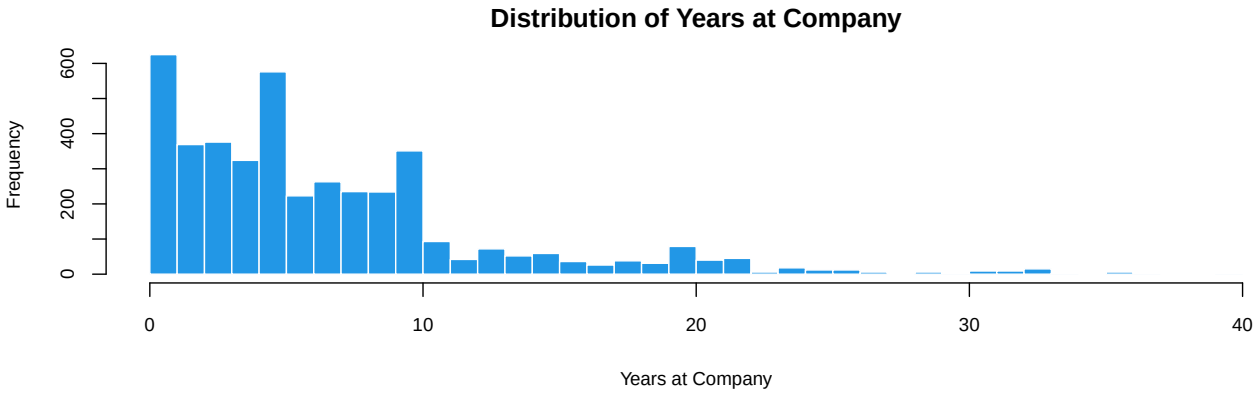




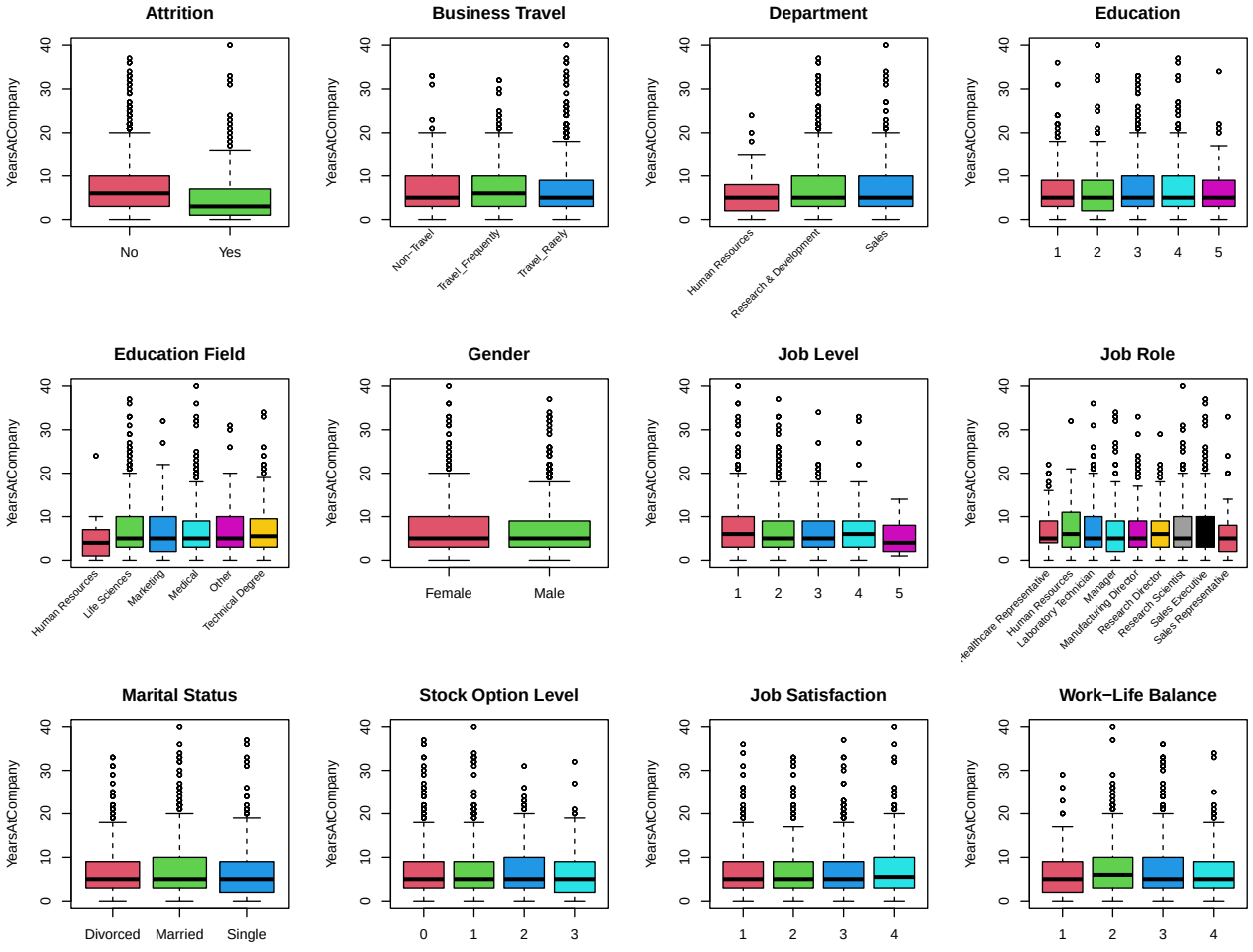
We notice that `YearsAtCompany` is highly correlated with `YearsWithCurrManager`. Therefore, we started building the regression model from this variable as we will see later.

5.3 Years At Company Analysis

First we visualized the distribution of `YearsAtCompany` to get an overview.

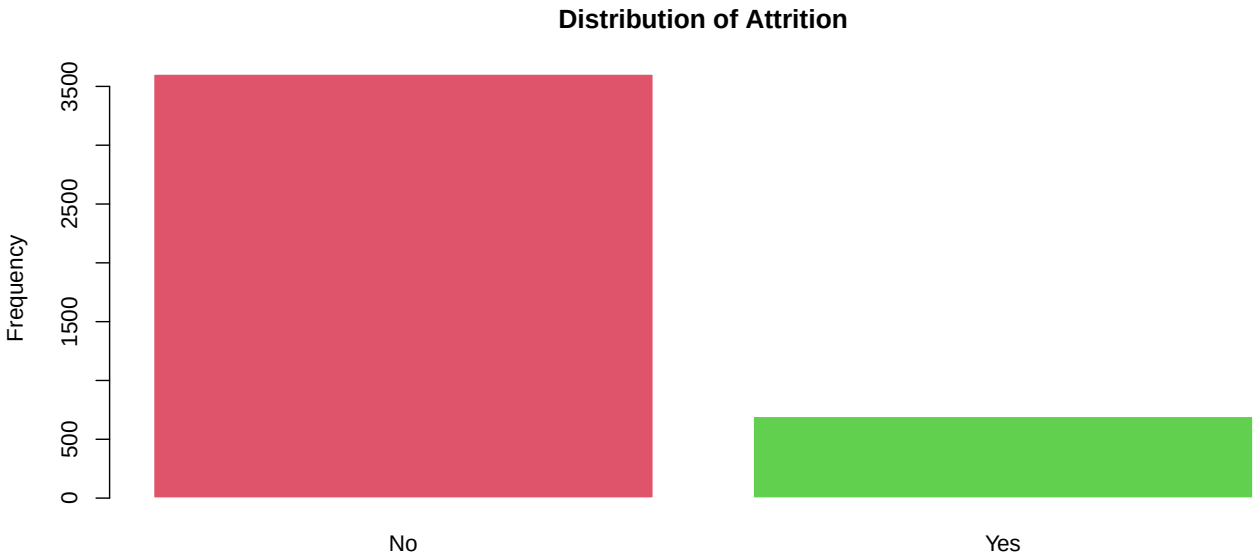


Then we analyzed the relationship of our variable of interest against all the categorical variables using boxplots.

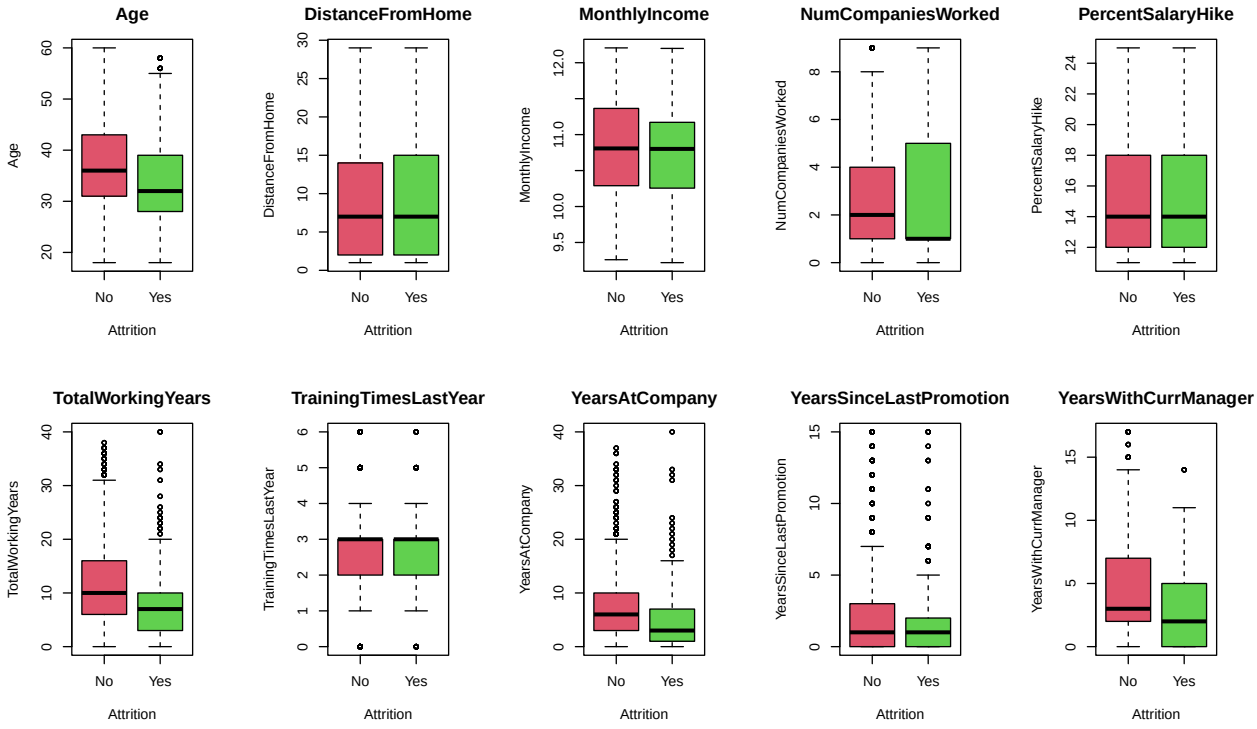


5.4 Attrition Analysis

We visualized the distribution of `Attrition`.



Afterwards, we plotted our variable of interest against all numerical variables.



6. Model Building

First, we split the dataset into a training and test set in a 80%/20% ratio.

```
set.seed(123)
n <- dim(data)[1]
test <- sample(1:n, n*0.2) # indexes of data in the test set
train <- setdiff(1:n, test) # indexes of data in training set
test.data <- data[test, ] # validation set
train.data <- data[train, ] # training set
```

```
## [1] "Number of observations in the training set:  3440"
```

```
## [1] "Number of observations in the test set:  860"
```

6.1 Regression Models

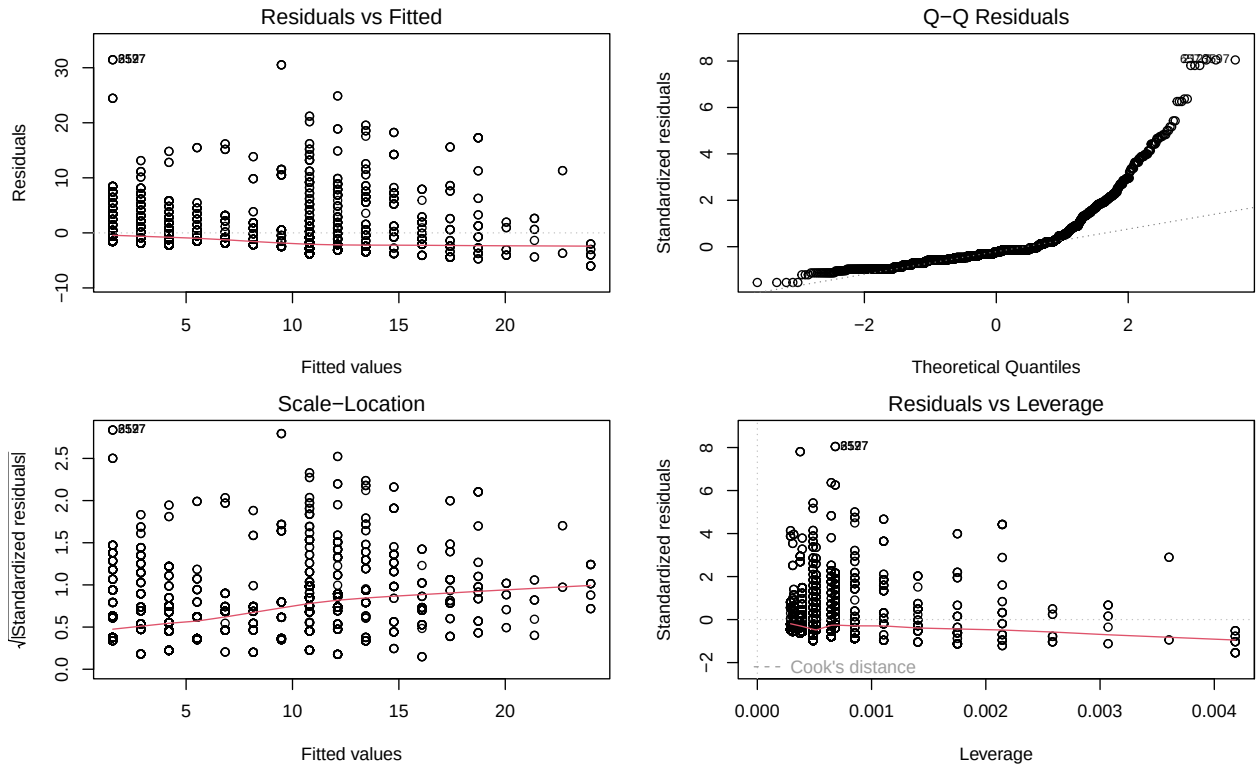
The goal was to develop a robust and accurate model that helps us understand which features influence the number of years an employee spends at the company and the nature of their impact. Our approach began with a simple regression model, which we incrementally enhanced using various statistical tools. The optimal model was identified based on the highest R-squared value, calculated using the test set for each model, and included the most significant variables for predicting `YearsAtCompany`.

6.1.1 Simple Linear Regression

We started with a simple regression model, which predicts `YearsAtCompany` using only `YearsWithCurrManager` as the independent variable.

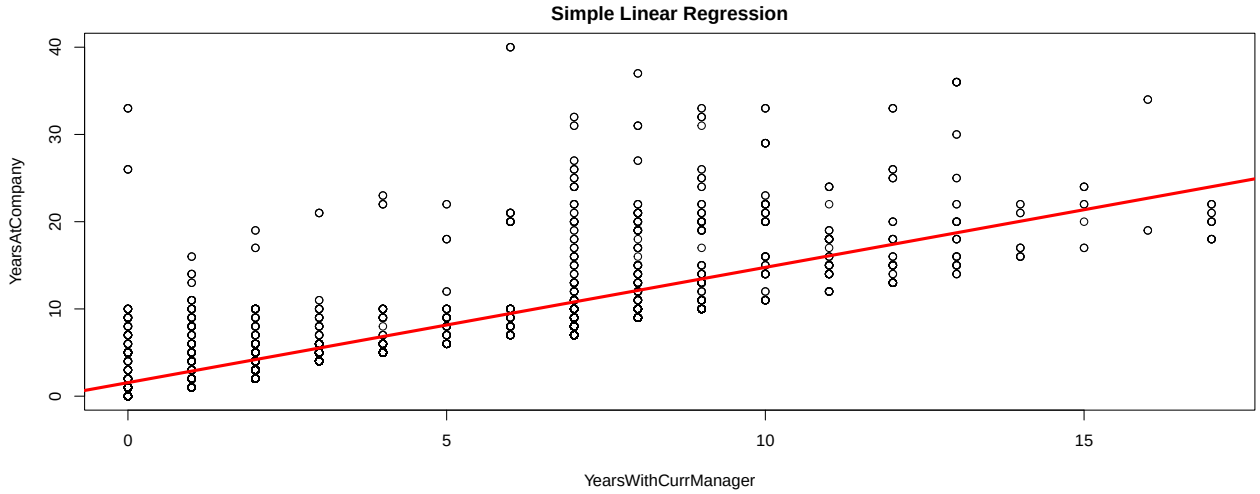
```
model_base <- lm(YearsAtCompany ~ YearsWithCurrManager, data = train.data)
R2_base <- summary(model_base)$r.squared
summary(model_base)
```

```
##
## Call:
## lm(formula = YearsAtCompany ~ YearsWithCurrManager, data = train.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.0139 -2.1219 -0.8546  0.4487 31.4487
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.5513     0.1021   15.19   <2e-16 ***
## YearsWithCurrManager  1.3213     0.0189   69.91   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.909 on 3438 degrees of freedom
## Multiple R-squared:  0.5871, Adjusted R-squared:  0.5869
## F-statistic:  4887 on 1 and 3438 DF,  p-value: < 2.2e-16
```



Since we used only one variable for this model, we plotted the regression line on a scatter plot to visualize how well the model fits the data.

```
par(mfrow = c(1, 1), mar = c(4, 4, 2, 2))
plot(train.data$YearsWithCurrManager, train.data$YearsAtCompany,
     xlab = "YearsWithCurrManager", ylab = "YearsAtCompany",
     main = "Simple Linear Regression")
abline(model_base, col = "red", lwd = 3)
```



Based on the low R-squared score of 0.59, this simple regression model lacks complexity and does not fit the data well.

6.1.2 Multiple Linear Regression

Seeing that a single predictor was insufficient, we built a multiple regression model using all the variables in the dataset, which resulted in an R-squared score of 0.75.

```
model_mlr1 <- lm(YearsAtCompany ~ ., data = train.data)
R2_mlr1 <- summary(model_mlr1)$r.squared
R2_mlr1
```

```
## [1] 0.7528953
```

6.1.3 Multiple Linear Regression without Multicollinearity

The multiple linear regression model demonstrates significant improvements over the simple linear regression model based on our comparison metric. However, we noticed that some variables were highly correlated in the current best model. Therefore, we decided to remove variables that were highly correlated (VIF > 5) and build our optimized model with the remaining variables.

```
vif_values <- vif(model_mlr1)
vif_values > 5
columns_to_remove <- c("Department", "EducationField")
data_reduced <- train.data[, !names(data) %in% columns_to_remove]
data_reduced_test <- test.data[, !names(data) %in% columns_to_remove]
```

```
model_mlr2 <- lm(YearsAtCompany ~ ., data = data_reduced)
R2_mlr2 <- summary(model_mlr2)$r.squared
R2_mlr2
```

```
## [1] 0.7515676
```

Based on the R-squared score, this model did not show any improvement compared to the previously best-performing model. Nevertheless, as multicollinearity potentially leads to unstable coefficients and reduced model interpretability, we omit variables `Department` and `EducationField` for further analysis.

6.1.4 Polynomial Regression

To account for more complex and non-linear interactions between predictor variables, we chose to use a polynomial regression model to possibly improve the fit of the model to the data. To achieve this, we will take the square of each of the currently selected features as predictors.

```
model_pr <- lm(YearsAtCompany ~
                poly(Age, 2, raw = TRUE) +
                poly(Attrition, 2, raw = TRUE) +
                poly(BusinessTravel, 2, raw = TRUE) +
                poly(DistanceFromHome, 2, raw = TRUE) +
                poly(Education, 2, raw = TRUE) +
                poly(Gender, 2, raw = TRUE) +
                poly(JobLevel, 2, raw = TRUE) +
                poly(JobRole, 2, raw = TRUE) +
                poly(MaritalStatus, 2, raw = TRUE) +
                poly(MonthlyIncome, 2, raw = TRUE) +
                poly(NumCompaniesWorked, 2, raw = TRUE) +
                poly(PercentSalaryHike, 2, raw = TRUE) +
                poly(StockOptionLevel, 2, raw = TRUE) +
                poly(TotalWorkingYears, 2, raw = TRUE) +
                poly(TrainingTimesLastYear, 2, raw = TRUE) +
                poly(YearsSinceLastPromotion, 2, raw = TRUE) +
                poly(YearsWithCurrManager, 2, raw = TRUE) +
```

```
                poly(JobSatisfaction, 2, raw = TRUE) +
                poly(WorkLifeBalance, 2, raw = TRUE),
                data = data_reduced)
R2_pr <- summary(model_pr)$r.squared
R2_pr
```

```
## [1] 0.7763953
```

We achieve an improvement of 0.02 in the R-squared value compared to the previous model.

6.1.5 Polynomial Regression with Feature Selection

To further improve the polynomial regression model, we used backward feature selection, which iteratively removes the variable that minimized the AIC the most.

```
backward_model <- step(model_pr, direction = "backward")
```

```
R2_backward <- summary(backward_model)$r.squared
R2_backward
```

```
## [1] 0.7750299
```

After backward elimination, the R-squared value of the model remains at 0.77, showing no improvement over the current best model. However, we have achieved a less complex model that predicts `YearsAtCompany` as effectively as the previous model. Therefore, we will continue working with the selected predictors.

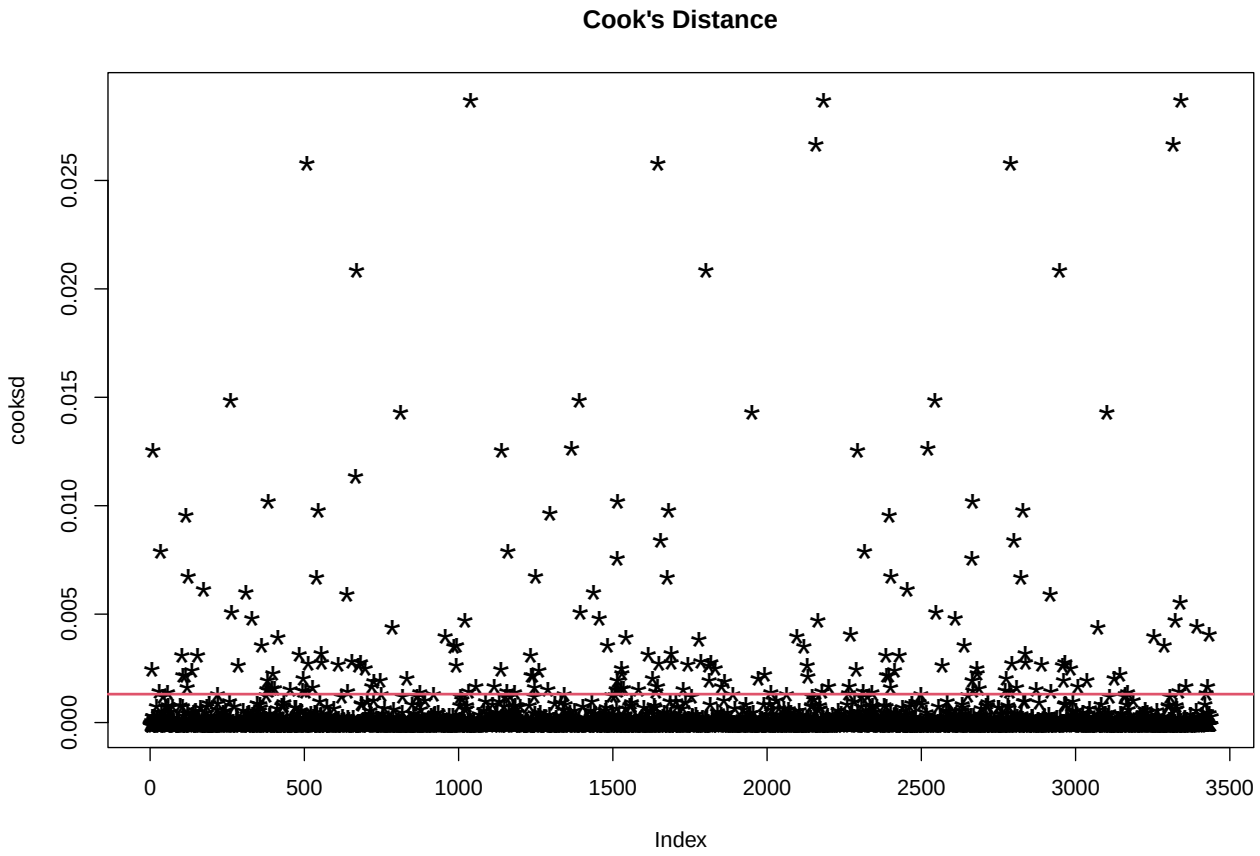
```
best_model <- lm(YearsAtCompany
                 ~ poly(Age, 2, raw = TRUE) +
                   poly(Education, 2, raw = TRUE) +
                   Gender +
                   poly(NumCompaniesWorked , 2, raw = TRUE) +
                   TotalWorkingYears +
                   poly(TrainingTimesLastYear, 2, raw = TRUE) +
                   poly(YearsSinceLastPromotion, 2, raw = TRUE) +
                   YearsWithCurrManager +
                   poly(JobSatisfaction, 2, raw = TRUE),

                 data = data_reduced)
R2_best <- summary(best_model)$r.squared
R2_best
```

```
## [1] 0.7724469
```

6.1.6 Final Model

As a final step, we removed influential points with a Cook's distance greater than three times the mean Cook's distance across all observations.

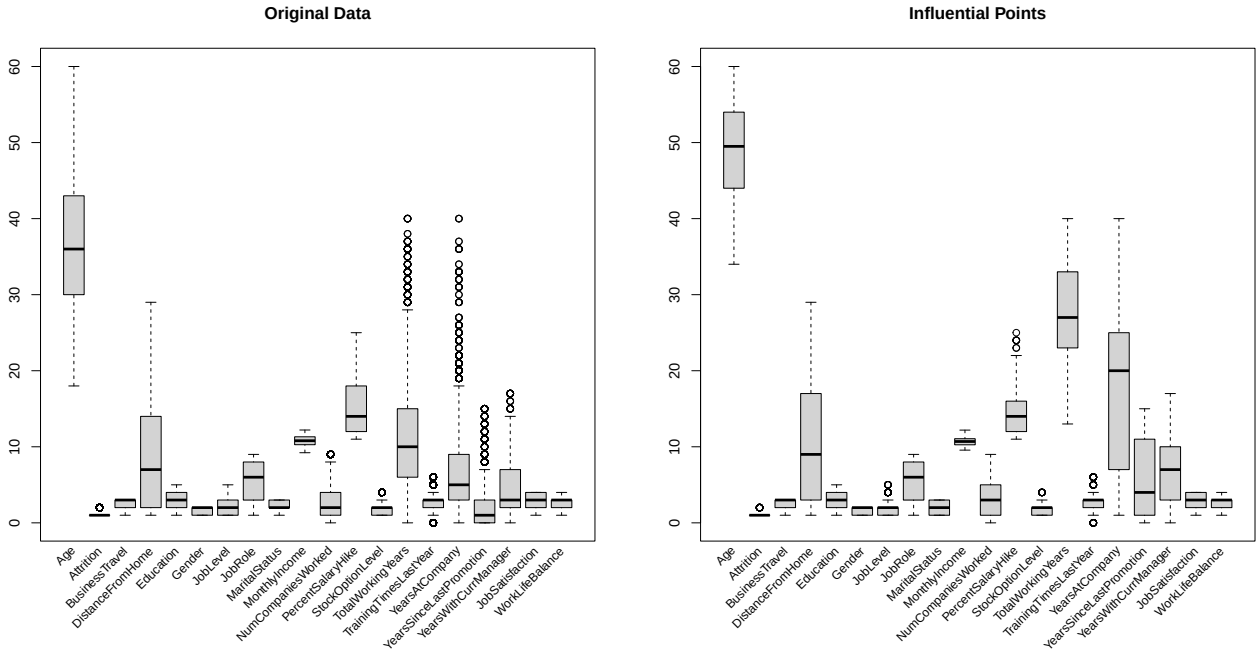
```
cooksd <- cooks.distance(best_model)
par(mfrow = c(1, 1))
plot(cooksd, pch = "*", cex = 2, main = "Cook's Distance")
abline(h = 3 *mean(cooksd, na.rm=TRUE),  col=c(2), lwd=2)
```




```
influential <- which(cooksd > 3*mean(cooksd, na.rm=TRUE))
length(influential)
```

```
## [1] 226
```

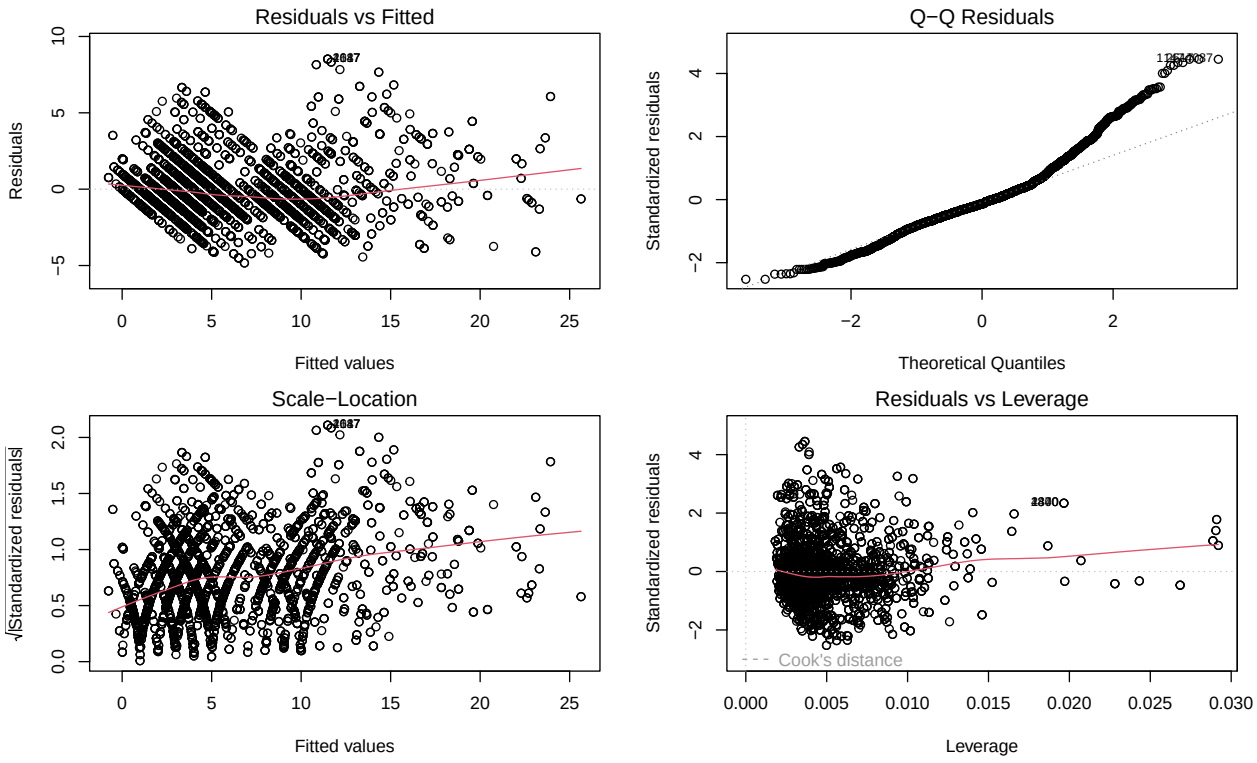
Comparing the original data to the points selected as influential, we can see that these points are mostly outliers. Thus, we decided to remove these points to potentially improve our model.



```
final_train.data <- data_reduced[-influential, ]
```

```
final_model <- lm(YearsAtCompany
                  ~ poly(Age, 2, raw = TRUE) +
                    poly(Education, 2, raw = TRUE) +
                    Gender +
                    poly(NumCompaniesWorked , 2, raw = TRUE) +
                    TotalWorkingYears +
                    poly(TrainingTimesLastYear, 2, raw = TRUE) +
                    poly(YearsSinceLastPromotion, 2, raw = TRUE) +
                    YearsWithCurrManager +
                    poly(JobSatisfaction, 2, raw = TRUE),

                  data = final_train.data)
R2_final <- summary(final_model)$r.squared
R2_final
```

```
## [1] 0.8417186
```



This final model achieved an R-squared value of 0.84, making it our best model overall. We conclude our regression analysis at this point and move on to the classification models.

6.2 Classification Models

The goal was to develop an accurate model that enables a company to understand which features affect employee attrition and how these features influence it. Given the binary response variable `Attrition`, our approach began with optimizing a Logistic Regression model and, using the selected predictors, we subsequently tested multiple well-known classification models in statistics. Our evaluation metric of choice was Accuracy, with the scores calculated based on the test set.

6.2.1 Logistic Regression

We fitted a simple logistic regression model that incorporated all variables as predictors. Here we reached an accuracy of 0.84, which is a good starting point.

```
initial_model_logit <- glm(Attrition ~ ., data = train.data, family = binomial)
```

6.2.2 Logistic Regression with Feature Selection

As we did earlier to enhance our polynomial regression model, we employed backward feature selection to identify and retain only the most significant variables for our task.

```
backward_model_logit <- step(initial_model_logit, direction = "backward")
summary_backward_model_logit <- summary(backward_model_logit)
summary_backward_model_logit
```

We observed a slight improvement of 0.01 in accuracy. For the same reasons outlined in the regression analysis, we will proceed with our analysis using the selected predictors.

```
final_model_logit <- glm(Attrition ~
                           Age +
                           BusinessTravel +
                           Department +
                           MaritalStatus +
                           NumCompaniesWorked +
                           TotalWorkingYears +
                           TrainingTimesLastYear +
                           YearsSinceLastPromotion +
                           YearsWithCurrManager +
                           JobSatisfaction +
                           WorkLifeBalance,
                         data = train.data, family = binomial)

summary_best_model_logit <- summary(final_model_logit)
```

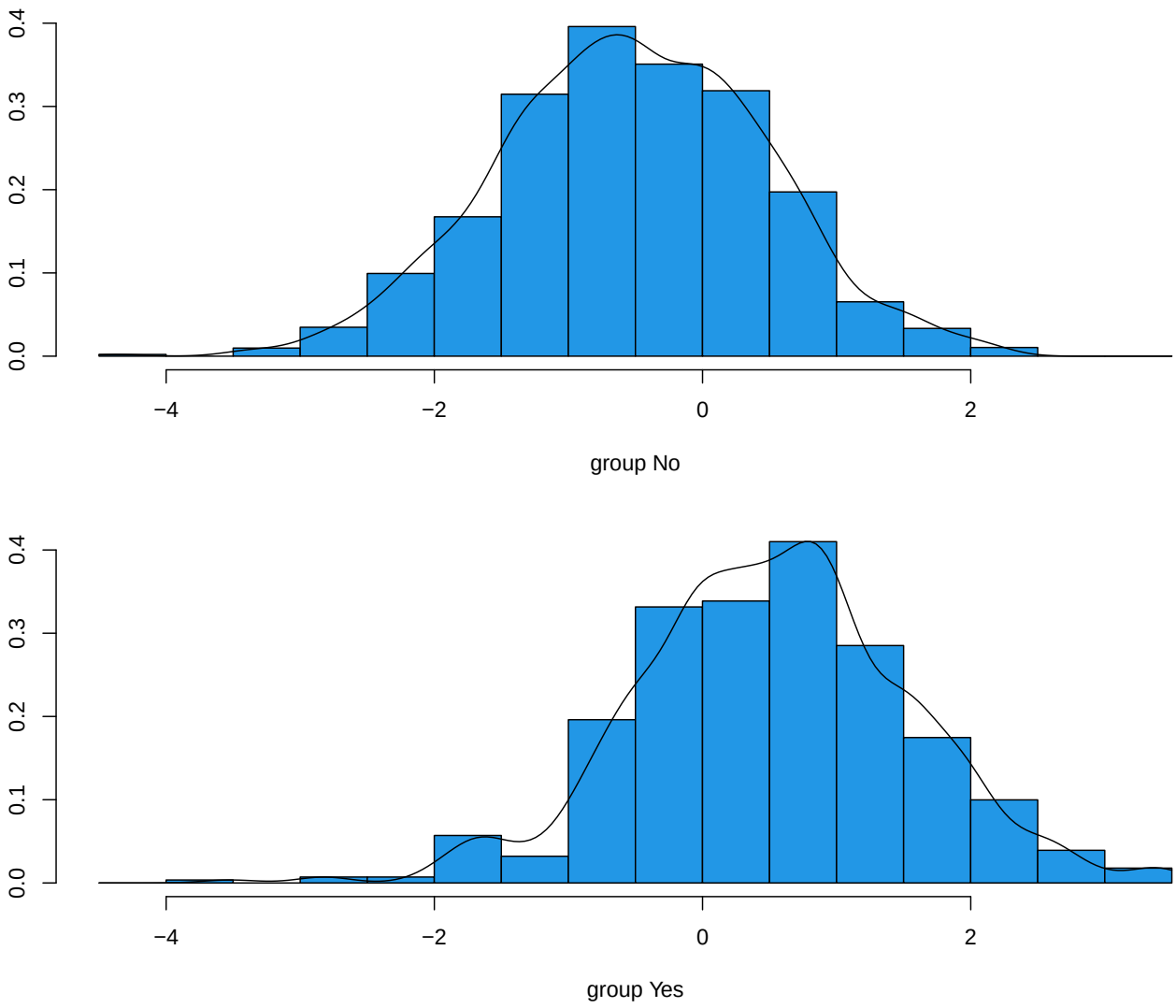
Next, we will concentrate on several other common classification models used in statistical analysis.

6.2.3 LDA (Linear Discriminant Analysis)

Using LDA, our goal was to identify the linear combinations of predictive features that most effectively differentiate between employees who remained with the company and those who left.

```
model_lda <- lda(Attrition ~
                   Age +
                   BusinessTravel +
                   Department +
                   MaritalStatus +
                   NumCompaniesWorked +
                   TotalWorkingYears +
                   TrainingTimesLastYear +
                   YearsSinceLastPromotion +
                   YearsWithCurrManager +
                   JobSatisfaction +
                   WorkLifeBalance,
                 data = train.data)
```

```r
par(mfrow = c(2, 1), mar = c(4, 4, 2, 2))
plot(model_lda, type="both", col = c(4))
```



We can see that the 2 graphs are not well separated and have significant overlap, which suggests that the discriminative power of the LDA model is limited.

Achieving an accuracy score of 0.84, the LDA model performed similarly to the logistic regression model.

6.2.4 QDA (Quadratic Discriminant Analysis)

QDA is a more flexible model than LDA, as it does not assume that the covariance of the predictors is the same in each class. It additionally is able to fit quadratic boundaries, leading to potentially better performance for complex datasets.

```r
model_qda <- qda(Attrition ~
                   Age +
                   BusinessTravel +
                   Department +
                   MaritalStatus +
                   NumCompaniesWorked +
```

```
                    TotalWorkingYears +
                    TrainingTimesLastYear +
                    YearsSinceLastPromotion +
                    YearsWithCurrManager +
                    JobSatisfaction +
                    WorkLifeBalance,
                  data = train.data)
```

In our analysis, the QDA model yielded a lower accuracy score compared to all other classification models we tested up to now.

6.2.5 Naive Bayes

Naive Bayes is a probabilistic classification algorithm that relies on Bayes' Theorem and assumes independence between features. Its main strengths are simplicity, efficiency, and scalability, making it effective for high-dimensional datasets. However, it struggles with highly correlated features and small datasets.

```
model_nb <- naiveBayes(Attrition ~
                         Age +
                         BusinessTravel +
                         Department +
                         MaritalStatus +
                         NumCompaniesWorked +
                         TotalWorkingYears +
                         TrainingTimesLastYear +
                         YearsSinceLastPromotion +
                         YearsWithCurrManager +
                         JobSatisfaction +
                         WorkLifeBalance,
                       data = train.data)
```

The Naive Bayes model achieved an accuracy score of 0.84, which is consistent with the performance of the other models we tested.

7 Evaluation

For the evaluation, we provide an overview of the performance of the models we developed. Specifically, we demonstrate the significant improvement of our final regression model compared to the initial one. Additionally, we deliver an in-depth analysis on the different classification models using a variety of different metrics. Given the imbalance in our dataset regarding the attrition variable, we conducted an additional analysis on the classification models using a balanced dataset, placing greater emphasis on the F1-Score. Finally, we highlight the most influential features for each model and provide our interpretation of their contextual meaning.

7.1 Regression models

Given that we found the final regression model to be the most accurate, we decided to compare the Mean Squared Error (MSE) on the test set between the initial and final model.

```
base.pred <- predict(model_base, newdata = test.data)
base.mse <- mean((test.data$YearsAtCompany - base.pred)^2)
print(paste("MSE of the initial model on test set:", base.mse))
```

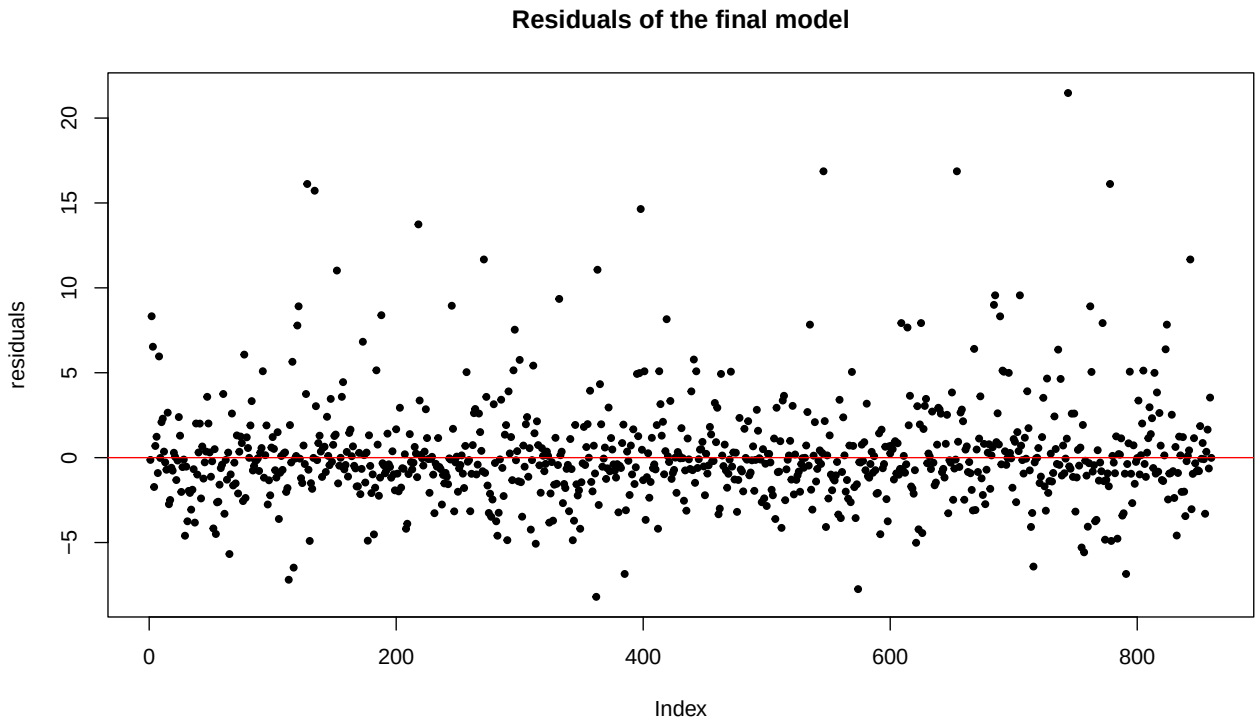
```
## [1] "MSE of the initial model on test set: 16.2086966065198"
```

```
final.pred <- predict(final_model, newdata = test.data)
final.mse <- mean((test.data$YearsAtCompany - final.pred)^2)
print(paste("MSE of the final model on test set:", final.mse))
```

```
## [1] "MSE of the final model on test set: 9.3755059331471"
```

Additionally, we showcase the Residuals of the final model.

```
residuals <- test.data$YearsAtCompany - final.pred
par(mfrow = c(1, 1))
plot(residuals, pch = 20, main = "Residuals of the final model")
abline(h = 0, col = "red")
```



Model	MSE	R-squared
Base Model	16.20	0.59
Final Model	9.37	0.84

The final model demonstrates a lower MSE and a higher R-squared value compared to the base model. Furthermore, due to the residuals being distributed randomly around 0, the superior model's predictions are unbiased.

7.2 Classification Models

7.2.1 Confusion matrices

Given similar performances across all classification models, we compared the models based on their confusion matrices.

```
perf.measure <- function(true.values, pred.values,  lab.pos = 1){
  conf.matrix <- table(pred.values, true.values)
```

```
  n <- sum(conf.matrix)
  lab.pos <- as.character(lab.pos)
  lab <- rownames(conf.matrix)
  lab.neg <- lab[lab != lab.pos]
  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
  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))
}
```

The confusion matrix of the initial logistic regression model.

```
conf_matrix_base_logit <- table(base.logit.pred, test.data$Attrition)
conf_matrix_base_logit
```

```
##
## base.logit.pred  No Yes
##             No  712 116
##             Yes  14  18
```

```
## [1] "Accuracy of the initial logistic regression model: 85 %"
```

The confusion matrix of the logistic regression model after feature selection.

```
conf_matrix_final_logit <- table(final.logit.pred, test.data$Attrition)
conf_matrix_final_logit
```

```
##
## final.logit.pred  No Yes
##              No  719 113
##              Yes   7  21
```

```
## [1] "Accuracy of the final logistic regression model: 86 %"
```

The confusion matrix of the LDA model.

```
lda.prob <- predict(model_lda, newdata = test.data)
lda.pred <- rep("No", 860)
lda.pred[lda.prob$posterior[,2]>= 0.5] <- "Yes"
conf_matrix_lda <- table(lda.pred, test.data$Attrition)
conf_matrix_lda
```

```
##
## lda.pred  No Yes
##      No  717 120
##      Yes   9  14
```

```
## [1] "Accuracy of the LDA model: 85 %"
```

The confusion matrix of the QDA model.

```
qda.prob <- predict(model_qda, newdata = test.data)
qda.pred <- rep("No", 860)
qda.pred[qda.prob$posterior[,2]>= 0.5] <- "Yes"
conf_matrix_qda <- table(qda.pred, test.data$Attrition)
conf_matrix_qda
```

```
##
## qda.pred  No Yes
##      No  637  83
##      Yes  89  51
```

```
## [1] "Accuracy of the QDA model: 80 %"
```

The confusion matrix of the Naive Bayes model.

```
nb.prob <- predict(model_nb, newdata = test.data, type="raw")
nb.pred <- predict(model_nb, newdata = test.data, type="class")
conf_matrix_nb <- table(nb.pred, test.data$Attrition)
conf_matrix_nb
```

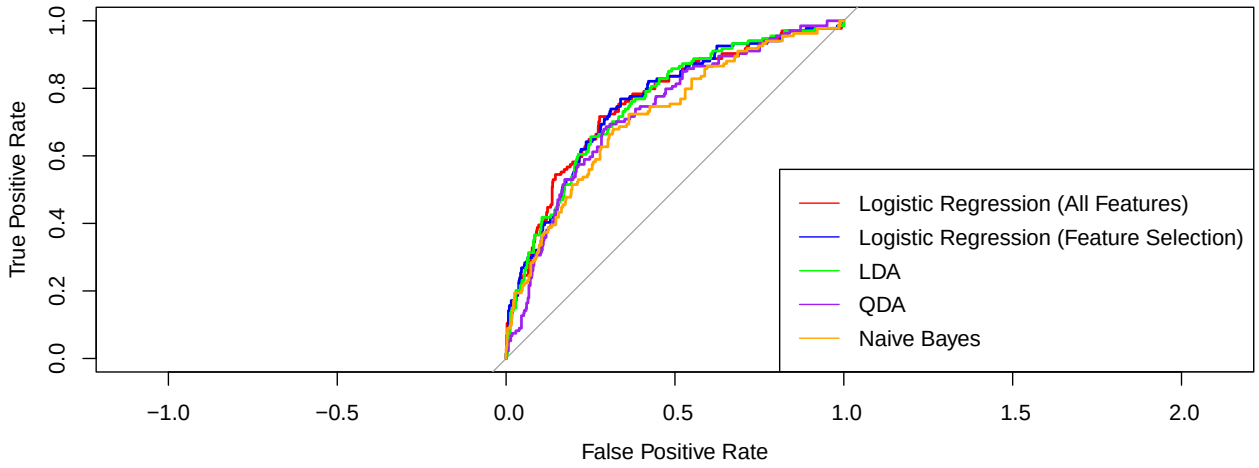
```
##
## nb.pred  No Yes
##     No  693 108
##     Yes  33  26
```

```
## [1] "Accuracy of the Naive Bayes model: 84 %"
```

7.2.2 ROC curves

Next, we plotted ROC curves for comparison.

```
roc.out_base <- roc(test.data$Attrition, base.logit.prob, levels=c("No", "Yes"))
roc.out_final <- roc(test.data$Attrition, final.logit.prob, levels=c("No", "Yes"))
roc.out_lda <- roc(test.data$Attrition, lda.prob$posterior[,2], levels=c("No", "Yes"))
roc.out_qda <- roc(test.data$Attrition, qda.prob$posterior[,2], levels=c("No", "Yes"))
roc.out_nb <- roc(test.data$Attrition, nb.prob[,2], levels=c("No", "Yes"))
plot(roc.out_base, col="red", legacy.axes=TRUE,
     xlab="False Positive Rate", ylab="True Positive Rate")
lines(roc.out_final, col="blue")
lines(roc.out_lda, col="green")
lines(roc.out_qda, col="purple")
lines(roc.out_nb, col="orange")
legend("bottomright", legend=c("Logistic Regression (All Features)",
                               "Logistic Regression (Feature Selection)",
                               "LDA", "QDA", "Naive Bayes"),
       col=c("red", "blue", "green", "purple", "orange"), lty=1)
```




```
auc_base <- round(auc(roc.out_base), 2)
auc_final <- round(auc(roc.out_final), 2)
auc_lda <- round(auc(roc.out_lda), 2)
auc_qda <- round(auc(roc.out_qda), 2)
auc_nb <- round(auc(roc.out_nb), 2)
```

7.2.3 Evaluation Summary

Using the obtained results, we compared the classification models based on various evaluation metrics.

Model	Accuracy	PPV	TPR	F1 Score	TNR	FPR	AUC
Logistic Regression (All Features)	0.85	0.56	0.13	0.21	0.98	0.02	0.76
Logistic Regression (Feature Selection)	0.86	0.75	0.16	0.26	0.99	0.01	0.76
LDA	0.85	0.60	0.10	0.18	0.98	0.01	0.75
QDA	0.80	0.36	0.38	0.37	0.87	0.12	0.73
Naive Bayes	0.84	0.44	0.20	0.27	0.95	0.05	0.72

The logistic regression model using all features shows moderate accuracy and AUC. The PPV and TNR are high, but the TPR is very low, indicating that the model is not capturing many of the actual positive instances (employees who leave). The logistic regression model after feature selection shows slight improvement in accuracy, TPR, and F1 Score and a strong increase in the PPV score. The improvement in PPV suggests better precision, possibly due to the removal of irrelevant features. The LDA model has similar accuracy to logistic regression but lower TPR and F1 Score. This indicates that LDA is also capturing fewer actual positives. The QDA model shows the highest F1-Score and TPR, suggesting it is better at identifying employees who will leave. However, this comes at the cost of lower TNR and higher FPR, indicating more false positives. The lower accuracy and AUC reflect this trade-off. The Naive Bayes model shows the lowest AUC score across all classification models.

We conclude that the logistic regression model with feature selection provides us with the strongest model in terms of accuracy and AUC score. The QDA model has the highest F1 and TPR score, which makes it useful if the primary goal is to identify as many employees who leave as possible.

7.2.4 Comparison of the Classification Models with Balanced Data

The low TPR score across most models indicates that they are not effectively identifying employees who leave. This is due to the heavy imbalance in our dataset, where only 16% of employees have an attrition value of

'yes'. To address this issue, we balanced the dataset by removing samples randomly from the majority class, resulting in a dataset where the number of employees staying is about twice the number of those leaving.

Model	Accuracy	PPV	TPR	F1 Score	TNR	FPR	AUC
Logistic Regression (All Features)	0.80	0.38	0.41	0.40	0.87	0.12	0.77
Logistic Regression (Feature Selection)	0.80	0.38	0.42	0.40	0.87	0.12	0.76
LDA	0.82	0.41	0.41	0.41	0.89	0.10	0.76
QDA	0.78	0.34	0.51	0.41	0.81	0.18	0.72
Naive Bayes	0.76	0.34	0.47	0.40	0.83	0.16	0.73

Here we find LDA to be the best model overall, outperforming the other models in nearly every metric. Thus, we propose to use it in cases where the dataset is more balanced with respect to the response variable of the classification model.

7.3 Feature Importance

In both models, we employed feature selection to identify the most important features. We will now analyze which variables had the greatest impact on the regression and classification models, and specifically how they influenced the outcomes.

7.3.1 Regression Models

We found that `YearsWithCurrManager` has the highest positive influence on `YearsAtCompany`. The longer an employee spends with their current manager, the more likely they are to stay at the company. Additionally, higher education levels and job satisfaction positively impact the number of years an employee spends at the company. Smaller positive influences include `TotalWorkingYears`, indicating that experience contributes positively to tenure, and `Age`, suggesting that older employees tend to stay longer. Conversely, `NumCompaniesWorked` has the strongest negative impact on `YearsAtCompany`, indicating that employees who have worked at more companies are more likely to leave, as they frequently change workplaces. Furthermore, `YearsSinceLastPromotion` negatively affects `YearsAtCompany`, suggesting that employees who have not been promoted in a long time are more likely to leave, possibly due to perceived lack of future opportunities at the company.

7.3.2 Classification Models

We found that both frequent and infrequent business travel have a significant positive influence on attrition. This indicates that employees who travel either frequently or rarely are more likely to leave the company compared to those who travel occasionally. Frequent travelers may experience stress from constant travel, while those who rarely travel may feel left out from company activities. Furthermore, we found that both single and married employees are likely to leave the company, although married employees are less likely to do so compared to single ones. We additionally find that `YearsSinceLastPromotion` and `NumCompaniesWorked` have small positive influence on attrition, indicating that employees who have not been promoted in a long time or have worked in many different companies are more likely to leave. As for the negative influences, we found that having a good `WorkLifeBalance` and higher `JobSatisfaction` are associated with no attrition. Lastly, employees working in Departments such as Research & Development and Sales are less likely to leave the company.

7.3.3 Comparison

When comparing the significant features in both the regression and classification models, the impact of `YearsSinceLastPromotion`, `YearsWithCurrentManager`, `JobSatisfaction` and `TotalWorkingYears` have a positive influence on `YearsAtCompany` and negative influence on `Attrition`. This means that these are

important factors when it comes to having employees work for a long time at a company without leaving. Nevertheless, we observe that the features with strong impact on one model do not necessarily appear among the features in the other model. This can be seen in `WorkLifeBalance`, `BusinessTravel` or `MaritalStatus` only appearing in the selected features of the classification models for `Attrition`, while `Education` was only relevant for `YearsAtCompany`. Thus, even though our two variables of interest share many significant features, they still need to be analyzed separately because each model has a number of unique features.