Predicting Employee Attrition Using Machine Learning

David

2025-05-05

Introduction

Employee attrition, commonly referred to as employee turnover, is the process through which employees leave an organization. Employee attrition can be due to many reasons, but the major causes include poor management, lack of career development opportunities, inadequate compensation relative to peer industries, work and life imbalances or limited recognition (Keserer, 2024). Employee attrition can have profound impacts on businesses or organizations in several ways e.g. losing top performers can create talent shortages that ripple through the organization, leading to disruptions in essential functions and increased recruitment costs. Attrition also diminishes productivity as experienced employees depart, often replaced by less skilled individuals who require time to reach peak performance levels. Frequent turnover can strain relationships with customers, suppliers and partners, thus damaging trust and potentially tarnishing the company's reputation in the marketplace t (Keserer, 2024 and Plum Insurance, 2024). Understanding and predicting employee attrition is therefore critical for organizations aiming to maintain a stable and productive workforce.

This analysis has the following two main objectives;

- To develop a reliable machine learning model that can accurately identify employees who are at high risk of leaving the organization.
- To determine the most significant factors contributing to employee attrition.

The dataset used in this analysis was obtained online from Kaggle.Link

```r
# Set working directory
setwd("~/R Programming/Projects/Classification/Employee Attrition 
Classification")

# Load packages
suppressPackageStartupMessages(
  {
    library(tidyverse)
    library(janitor)
    library(caret)
    library(mlr)
    library(pROC)
    library(yardstick)
    library(vip)
    library(corrplot)
    library(parallel)
    library(parallelMap)
  }
)

# Import data
HR_Employee_Attrition <- read_csv("HR-Employee-Attrition.csv")

## Rows: 1470 Columns: 35
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (9): Attrition, BusinessTravel, Department, EducationField, Gender,
Job...
## dbl (26): Age, DailyRate, DistanceFromHome, Education, EmployeeCount,
Employ...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.

# View the structure of the data set
HR_Employee_Attrition |> glimpse()

## Rows: 1,470
## Columns: 35
## $ Age                      <dbl> 41, 49, 37, 33, 27, 32, 59, 30, 38, 36,
35, 2…
## $ Attrition                <chr> "Yes", "No", "Yes", "No", "No", "No",
"No", "…
## $ BusinessTravel           <chr> "Travel_Rarely", "Travel_Frequently",
"Travel…
## $ DailyRate                <dbl> 1102, 279, 1373, 1392, 591, 1005, 1324,
1358,…
## $ Department               <chr> "Sales", "Research & Development",
"Research …
## $ DistanceFromHome         <dbl> 1, 8, 2, 3, 2, 2, 3, 24, 23, 27, 16, 15,
26, …
## $ Education                <dbl> 2, 1, 2, 4, 1, 2, 3, 1, 3, 3, 3, 2, 1, 2,
3, …
## $ EducationField           <chr> "Life Sciences", "Life Sciences",
"Other", "L…
## $ EmployeeCount            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, …
## $ EmployeeNumber           <dbl> 1, 2, 4, 5, 7, 8, 10, 11, 12, 13, 14, 15,
16,…
## $ EnvironmentSatisfaction  <dbl> 2, 3, 4, 4, 1, 4, 3, 4, 4, 3, 1, 4, 1, 2,
3, …
## $ Gender                   <chr> "Female", "Male", "Male", "Female",
"Male", "…
## $ HourlyRate               <dbl> 94, 61, 92, 56, 40, 79, 81, 67, 44, 94,
84, 4…
## $ JobInvolvement           <dbl> 3, 2, 2, 3, 3, 3, 4, 3, 2, 3, 4, 2, 3, 3,
2, …
## $ JobLevel                 <dbl> 2, 2, 1, 1, 1, 1, 1, 1, 3, 2, 1, 2, 1, 1,
1, …
## $ JobRole                  <chr> "Sales Executive", "Research Scientist",
"Lab…
## $ JobSatisfaction          <dbl> 4, 2, 3, 3, 2, 4, 1, 3, 3, 3, 2, 3, 3, 4,
3, …
## $ MaritalStatus            <chr> "Single", "Married", "Single", "Married",
"Ma…
## $ MonthlyIncome            <dbl> 5993, 5130, 2090, 2909, 3468, 3068, 2670,
269…
## $ MonthlyRate              <dbl> 19479, 24907, 2396, 23159, 16632, 11864,
9964…
## $ NumCompaniesWorked       <dbl> 8, 1, 6, 1, 9, 0, 4, 1, 0, 6, 0, 0, 1, 0,
5, …
## $ Over18                   <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y",
"Y", …
## $ OverTime                 <chr> "Yes", "No", "Yes", "Yes", "No", "No",
"Yes",…
## $ PercentSalaryHike        <dbl> 11, 23, 15, 11, 12, 13, 20, 22, 21, 13,
13, 1…
## $ PerformanceRating        <dbl> 3, 4, 3, 3, 3, 3, 4, 4, 4, 3, 3, 3, 3, 3,
3, …
## $ RelationshipSatisfaction <dbl> 1, 4, 2, 3, 4, 3, 1, 2, 2, 2, 3, 4, 4, 3,
2, …
## $ StandardHours            <dbl> 80, 80, 80, 80, 80, 80, 80, 80, 80, 80,
80, 8…
## $ StockOptionLevel         <dbl> 0, 1, 0, 0, 1, 0, 3, 1, 0, 2, 1, 0, 1, 1,
0, …
## $ TotalWorkingYears        <dbl> 8, 10, 7, 8, 6, 8, 12, 1, 10, 17, 6, 10,
5, 3…
## $ TrainingTimesLastYear    <dbl> 0, 3, 3, 3, 3, 2, 3, 2, 2, 3, 5, 3, 1, 2,
4, …
## $ WorkLifeBalance          <dbl> 1, 3, 3, 3, 3, 2, 2, 3, 3, 2, 3, 3, 2, 3,
3, …
## $ YearsAtCompany           <dbl> 6, 10, 0, 8, 2, 7, 1, 1, 9, 7, 5, 9, 5,
2, 4,…
## $ YearsInCurrentRole       <dbl> 4, 7, 0, 7, 2, 7, 0, 0, 7, 7, 4, 5, 2, 2,
2, …
## $ YearsSinceLastPromotion  <dbl> 0, 1, 0, 3, 2, 3, 0, 0, 1, 7, 0, 0, 4, 1,
0, …
## $ YearsWithCurrManager     <dbl> 5, 7, 0, 0, 2, 6, 0, 0, 8, 7, 3, 8, 3, 2,
3, …
```

The data has 1,470 observations of 35 variables. Attrition, Business Travel, Department, Education Field, Gender, Job Role, Marital Status, Over 18 and Over Time are character variables, while the rest of the variables are numeric (double).

```r
# View the first few observations
HR_Employee_Attrition |> head()
```

```
## # A tibble: 6 × 35
##     Age Attrition BusinessTravel DailyRate Department DistanceFromHome
Education
##   <dbl> <chr>     <chr>              <dbl> <chr>                 <dbl>
<dbl>
## 1    41 Yes       Travel_Rarely       1102 Sales                     1
2
## 2    49 No        Travel_Freque…       279 Research …                8
1
## 3    37 Yes       Travel_Rarely       1373 Research …                2
2
## 4    33 No        Travel_Freque…      1392 Research …                3
4
## 5    27 No        Travel_Rarely        591 Research …                2
1
## 6    32 No        Travel_Freque…      1005 Research …                2
2
## # ℹ 28 more variables: EducationField <chr>, EmployeeCount <dbl>,
## #   EmployeeNumber <dbl>, EnvironmentSatisfaction <dbl>, Gender <chr>,
## #   HourlyRate <dbl>, JobInvolvement <dbl>, JobLevel <dbl>, JobRole <chr>,
## #   JobSatisfaction <dbl>, MaritalStatus <chr>, MonthlyIncome <dbl>,
## #   MonthlyRate <dbl>, NumCompaniesWorked <dbl>, Over18 <chr>, OverTime
<chr>,
## #   PercentSalaryHike <dbl>, PerformanceRating <dbl>,
## #   RelationshipSatisfaction <dbl>, StandardHours <dbl>, …
```

Data Cleaning and Preprocessing

The process of data cleaning will involve assessing for, and handling data quality issues such as missing values, white spaces, duplicated observations and inconsistent values. The character variables will also be converted to factors.

```r
# Clean variable names
HR_Employee_Attrition <- clean_names(HR_Employee_Attrition)

# Check for missing values in each column
map_dbl(HR_Employee_Attrition, ~sum(is.na(.)))
```

```
##                        age                  attrition
##                          0                          0
##            business_travel                 daily_rate
##                          0                          0
##                 department         distance_from_home
##                          0                          0
##                  education            education_field
##                          0                          0
##             employee_count            employee_number
##                          0                          0
##   environment_satisfaction                     gender
##                          0                          0
##                hourly_rate            job_involvement
##                          0                          0
##                  job_level                   job_role
##                          0                          0
##           job_satisfaction             marital_status
##                          0                          0
##             monthly_income               monthly_rate
##                          0                          0
##       num_companies_worked                     over18
##                          0                          0
##                  over_time        percent_salary_hike
##                          0                          0
##         performance_rating  relationship_satisfaction
##                          0                          0
##             standard_hours         stock_option_level
##                          0                          0
##        total_working_years   training_times_last_year
##                          0                          0
##          work_life_balance           years_at_company
##                          0                          0
##      years_in_current_role years_since_last_promotion
##                          0                          0
##    years_with_curr_manager
##                          0
```

The data has no missing values.

```
# Check for duplicated observations
sum(duplicated(HR_Employee_Attrition))

## [1] 0
```

There are no duplicated observations in the data.

```
# Convert the character variables to factors

# Specify columns to factor
cols_to_factor <- c("business_travel", "department", "education_field",
"gender",
                    "over18", "job_role", "marital_status", "over_time")
# Convert the specified columns into factors
HR_Employee_Attrition <- HR_Employee_Attrition |>
  mutate_at(.vars = cols_to_factor, .fun = factor)
```

```r
# Factor the variable Education Level (education)
HR_Employee_Attrition$education <- factor(HR_Employee_Attrition$education,
                                          labels = c("Below College",
"College",
                                                     "Degree", "Masters",
"Doctor(PhD)"),
                                          levels = c(1,2,3,4,5))

# Convert the target "attrition" into a binary variable
HR_Employee_Attrition$attrition <- ifelse(HR_Employee_Attrition$attrition ==
"Yes", 1, 0)

# Factor the target variable
HR_Employee_Attrition$attrition <- factor(HR_Employee_Attrition$attrition,
                                          levels = c(0,1),
                                          labels = c("No", "Yes"))
```

Exploratory Data Analysis

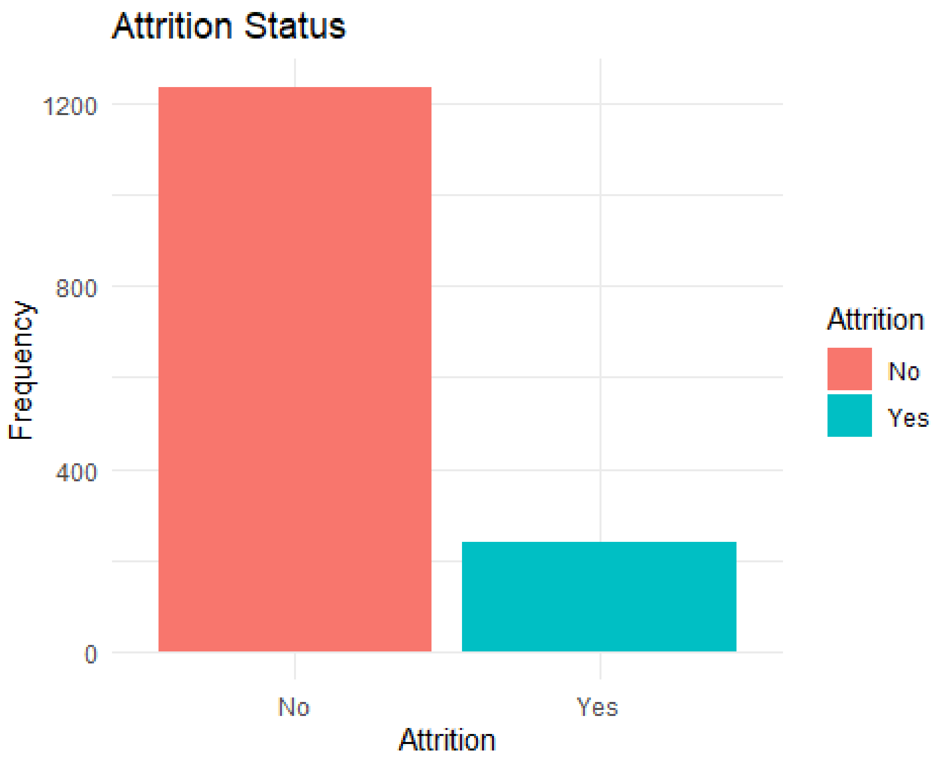
EDA is a crucial step before modeling because it helps uncover patterns, anomalies, and relationships in the data. This process will involve generating summary statistics for each column, and visualizing the data to help in understanding its main features, and uncover the underlying patterns.

```r
# Have a statistical summary of the data
summary(HR_Employee_Attrition)

##       age        attrition           business_travel   daily_rate
##  Min.   :18.00   No :1233   Non-Travel       : 150   Min.   : 102.0
##  1st Qu.:30.00   Yes: 237   Travel_Frequently: 277   1st Qu.: 465.0
##  Median :36.00              Travel_Rarely    :1043   Median : 802.0
##  Mean   :36.92                                       Mean   : 802.5
##  3rd Qu.:43.00                                       3rd Qu.:1157.0
##  Max.   :60.00                                       Max.   :1499.0
##
##                     department  distance_from_home          education
##  Human Resources        : 63   Min.   : 1.000     Below College:170
##  Research & Development:961   1st Qu.: 2.000     College      :282
##  Sales                 :446   Median : 7.000     Degree       :572
##                               Mean   : 9.193     Masters      :398
##                               3rd Qu.:14.000     Doctor(PhD)  : 48
##                               Max.   :29.000
##
##           education_field employee_count employee_number
##  Human Resources : 27     Min.   :1      Min.   :   1.0
##  Life Sciences   :606     1st Qu.:1      1st Qu.: 491.2
##  Marketing       :159     Median :1      Median :1020.5
##  Medical         :464    Mean   :1      Mean   :1024.9
##  Other           : 82    3rd Qu.:1      3rd Qu.:1555.8
##  Technical Degree:132    Max.   :1      Max.   :2068.0
##
##  environment_satisfaction    gender     hourly_rate     job_involvement
##  Min.   :1.000           Female:588   Min.   : 30.00   Min.   :1.00
##  1st Qu.:2.000           Male  :882   1st Qu.: 48.00   1st Qu.:2.00
##  Median :3.000                        Median : 66.00   Median :3.00
##  Mean   :2.722                        Mean   : 65.89   Mean   :2.73
##  3rd Qu.:4.000                        3rd Qu.: 83.75   3rd Qu.:3.00
##  Max.   :4.000                        Max.   :100.00   Max.   :4.00
##
##    job_level                          job_role   job_satisfaction
##  Min.   :1.000   Sales Executive          :326   Min.   :1.000
##  1st Qu.:1.000   Research Scientist       :292   1st Qu.:2.000
##  Median :2.000   Laboratory Technician    :259   Median :3.000
##  Mean   :2.064   Manufacturing Director   :145   Mean   :2.729
##  3rd Qu.:3.000   Healthcare Representative:131   3rd Qu.:4.000
##  Max.   :5.000   Manager                  :102   Max.   :4.000
##                  (Other)                  :215
##   marital_status monthly_income   monthly_rate   num_companies_worked
over18
##  Divorced:327    Min.   : 1009   Min.   : 2094   Min.   :0.000
Y:1470
##  Married :673    1st Qu.: 2911   1st Qu.: 8047   1st Qu.:1.000
##  Single  :470    Median : 4919   Median :14236   Median :2.000
##                  Mean   : 6503   Mean   :14313   Mean   :2.693
##                  3rd Qu.: 8379   3rd Qu.:20462   3rd Qu.:4.000
##                  Max.   :19999   Max.   :26999   Max.   :9.000
##
##  over_time  percent_salary_hike performance_rating
relationship_satisfaction
##  No :1054   Min.   :11.00       Min.   :3.000      Min.   :1.000
##  Yes: 416   1st Qu.:12.00       1st Qu.:3.000      1st Qu.:2.000
##             Median :14.00       Median :3.000      Median :3.000
##             Mean   :15.21       Mean   :3.154      Mean   :2.712
##             3rd Qu.:18.00       3rd Qu.:3.000      3rd Qu.:4.000
##             Max.   :25.00       Max.   :4.000      Max.   :4.000
##
##  standard_hours stock_option_level total_working_years
training_times_last_year
##  Min.   :80     Min.   :0.0000     Min.   : 0.00       Min.   :0.000
##  1st Qu.:80     1st Qu.:0.0000     1st Qu.: 6.00       1st Qu.:2.000
##  Median :80     Median :1.0000     Median :10.00       Median :3.000
##  Mean   :80     Mean   :0.7939     Mean   :11.28       Mean   :2.799
##  3rd Qu.:80     3rd Qu.:1.0000     3rd Qu.:15.00       3rd Qu.:3.000
##  Max.   :80     Max.   :3.0000     Max.   :40.00       Max.   :6.000
##
##  work_life_balance years_at_company years_in_current_role
##  Min.   :1.000     Min.   : 0.000   Min.   : 0.000
##  1st Qu.:2.000     1st Qu.: 3.000   1st Qu.: 2.000  
##  Median :3.000     Median : 5.000   Median : 3.000  
##  Mean   :2.761     Mean   : 7.008   Mean   : 4.229  
##  3rd Qu.:3.000     3rd Qu.: 9.000   3rd Qu.: 7.000  
##  Max.   :4.000     Max.   :40.000   Max.   :18.000  
##                                                     
##  years_since_last_promotion years_with_curr_manager
##  Min.   : 0.000             Min.   : 0.000         
##  1st Qu.: 0.000             1st Qu.: 2.000         
##  Median : 1.000             Median : 3.000         
##  Mean   : 2.188             Mean   : 4.123         
##  3rd Qu.: 3.000             3rd Qu.: 7.000         
##  Max.   :15.000             Max.   :17.000         
##
```

The mean age of employees was 36.92 years. 237 employees left the company while 1233 did not leave the company. Most of the employees rarely went for business travels. The average daily rate was 802.5 US dollars, while the average hourly rate was 65.89 US dollars. Most of the employees were from the department of Research and Development. The average monthly income and monthly rate were 6,503 and 14,313 US dollars respectively. Most of the employees were Sale Executives. Also, most employees did not work overtime. The mean percentage salary hike was 15.21%. Most of the employees had stock option levels 0 and 1.

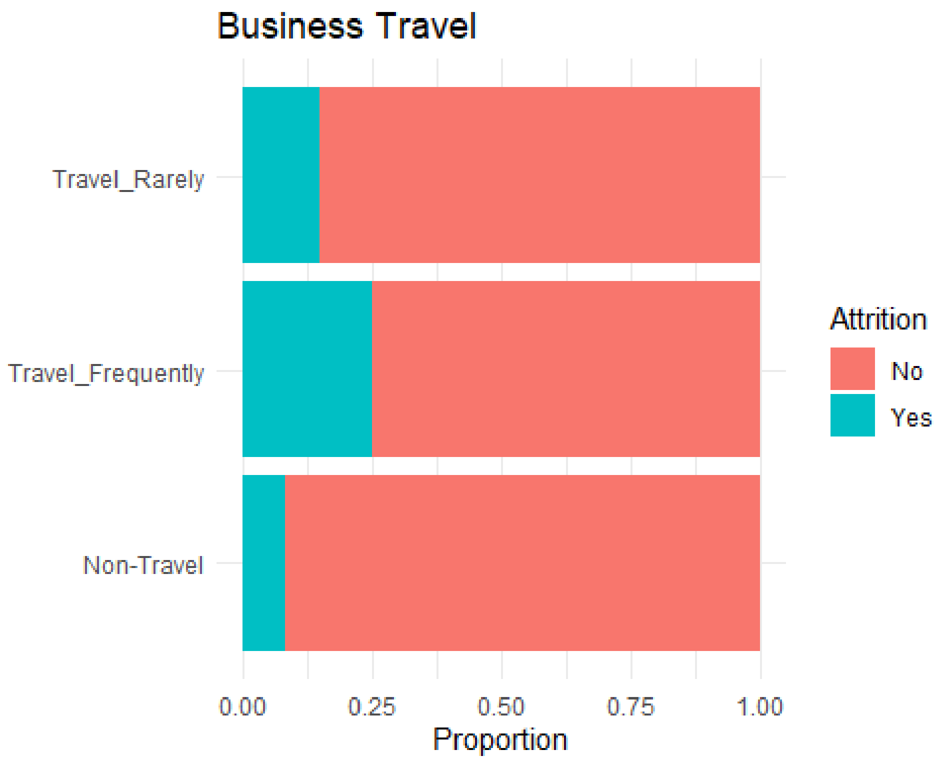
Visualize the Data

```
## Begin with Categorical Features

# The Distribution of the target variable Attrition
ggplot(HR_Employee_Attrition, aes(attrition, fill = attrition)) +
  geom_bar() +
  labs(title = "Attrition Status",
       x = "Attrition", y = "Frequency",
       fill = "Attrition") +
  theme_minimal()
```



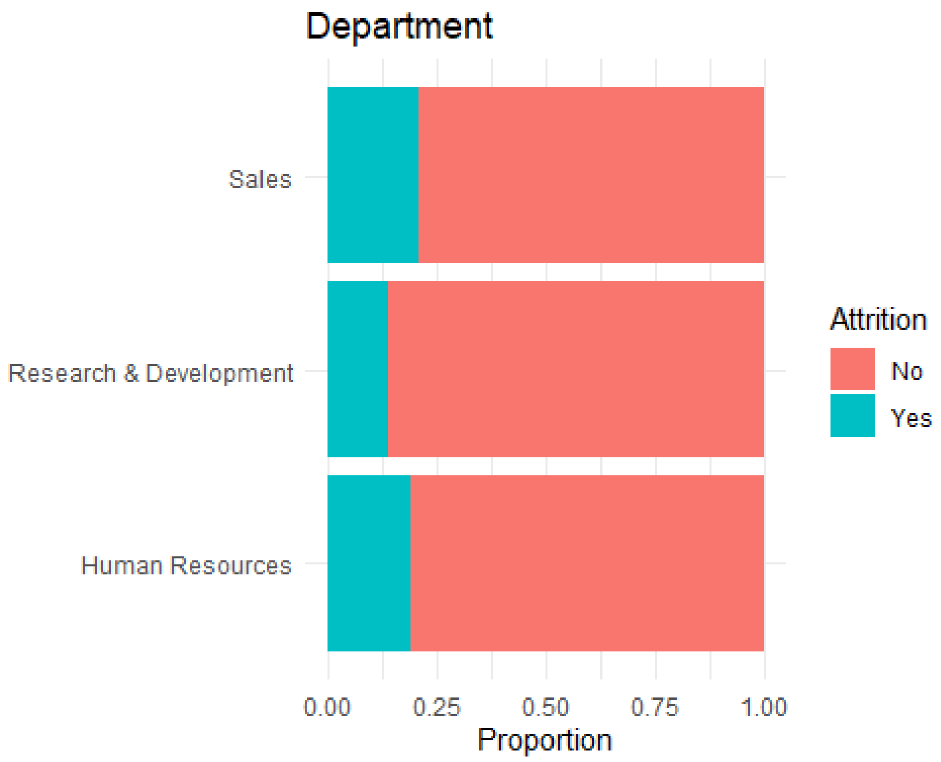
Attrition was very unlikely among the employees. More than 1200 employees did not leave the company. Attrition was about 5 times less likely.

```r
# Attrition by Business Travel
ggplot(HR_Employee_Attrition, aes(business_travel, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Business Travel",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



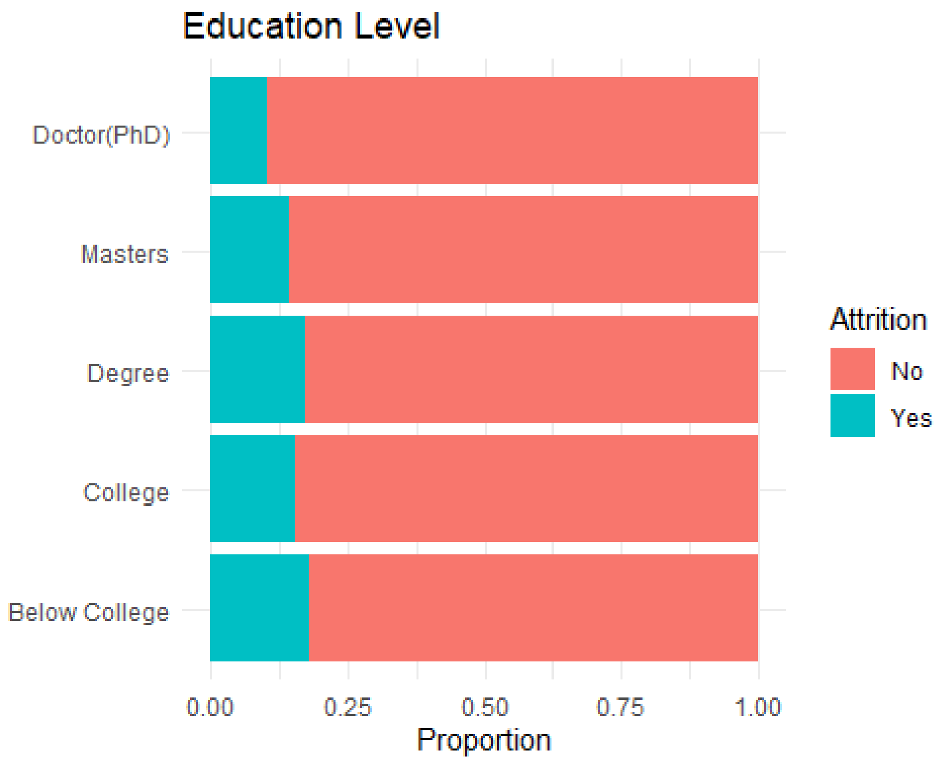
Attrition rate was highest among employees who travel frequently (25%), followed by those who travel rarely (about 14%). The rate of attrition was least among employees who never went for business travels.

```
# Attrition by Department
ggplot(HR_Employee_Attrition, aes(department, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Department",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



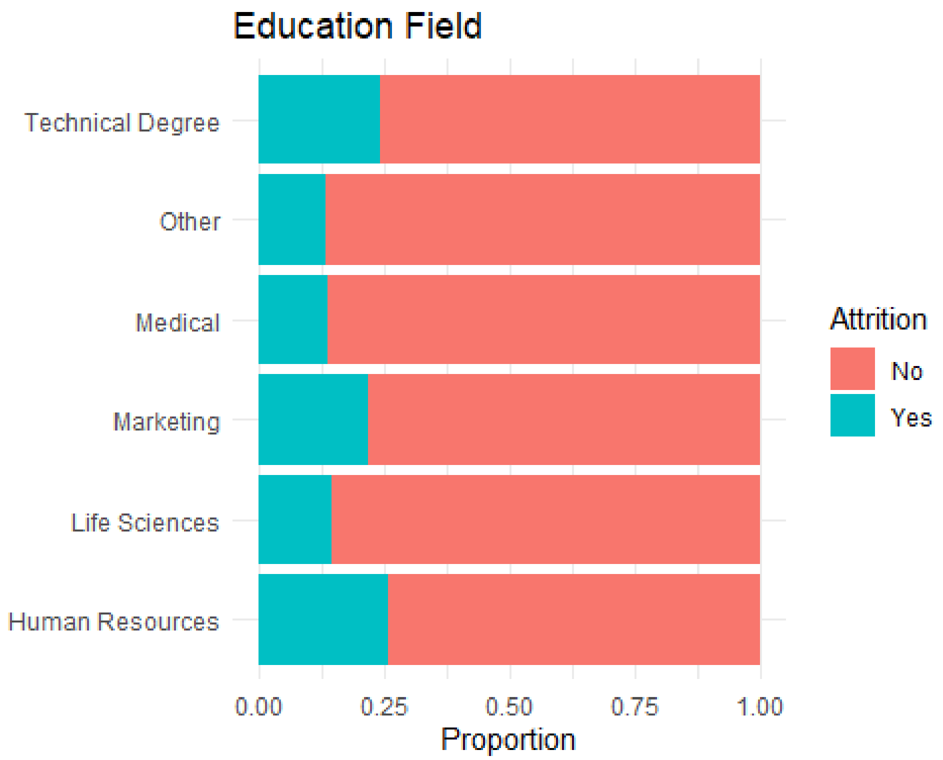
Attrition rate was highest in the Sales department closely followed by HR.

```r
# Attrition by Education Level
ggplot(HR_Employee_Attrition, aes(education, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Education Level",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



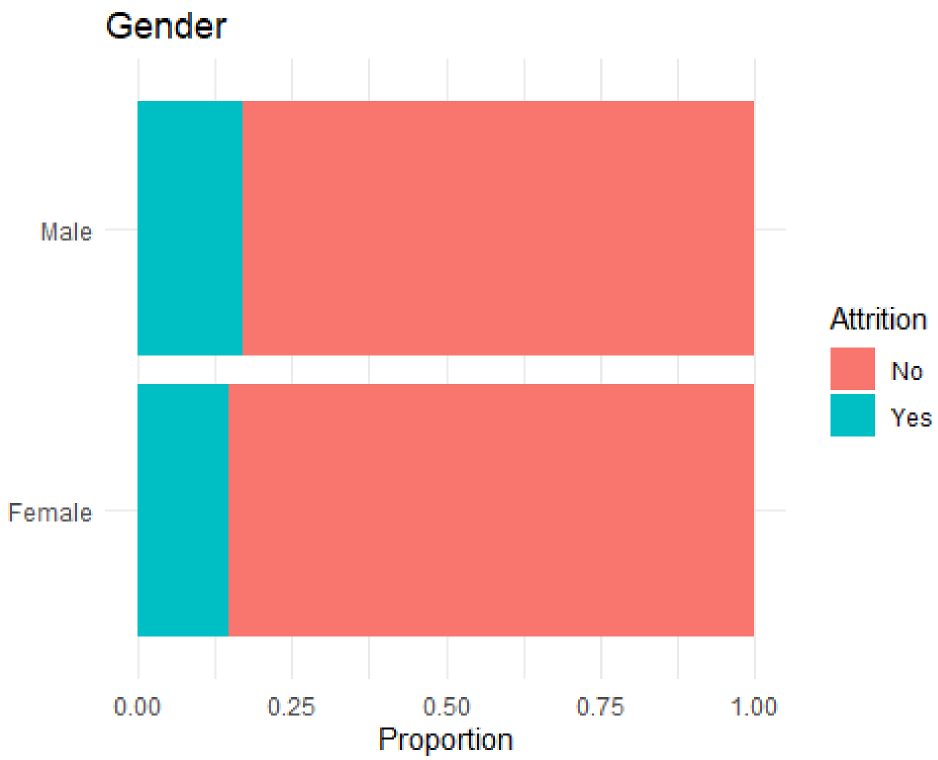
Attrition was highest among those who had basic education (below college), closely followed by those who had bachelor’s degrees, then college graduates and Master’s holders respectively.

```r
# Attrition by Education Field
ggplot(HR_Employee_Attrition, aes(education_field, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Education Field",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



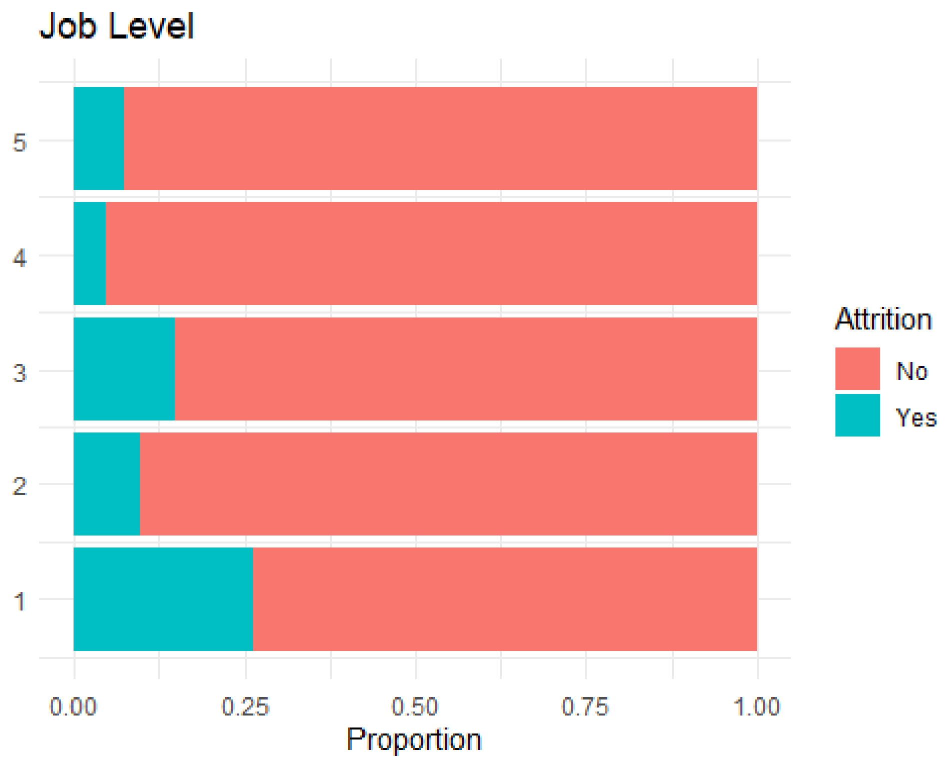
Attrition rate was highest among employees within HR education field (about 26%), followed by Technical Degree field (about 23%) and Marketing (about 22%) respectively. Medical field and "Other" nearly had the same rates of attrition.

```r
# Attrition by Gender
ggplot(HR_Employee_Attrition, aes(gender, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Gender",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



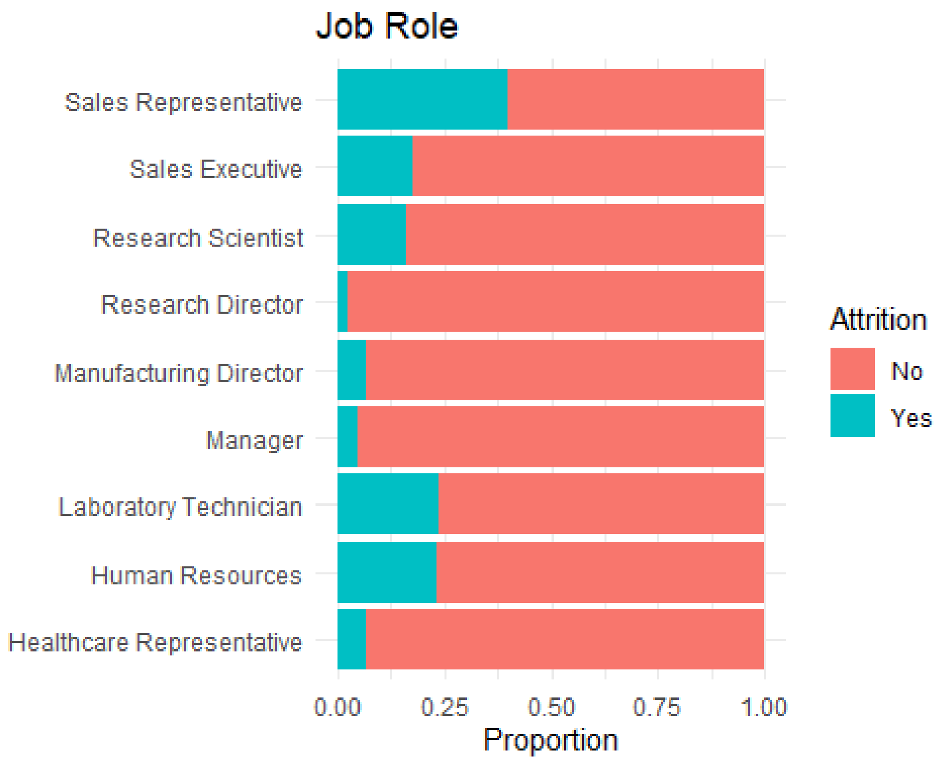
Male employees had a slightly higher rate of attrition than female employees.

```
# Attrition by Job Level
ggplot(HR_Employee_Attrition, aes(job_level, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Job Level",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```

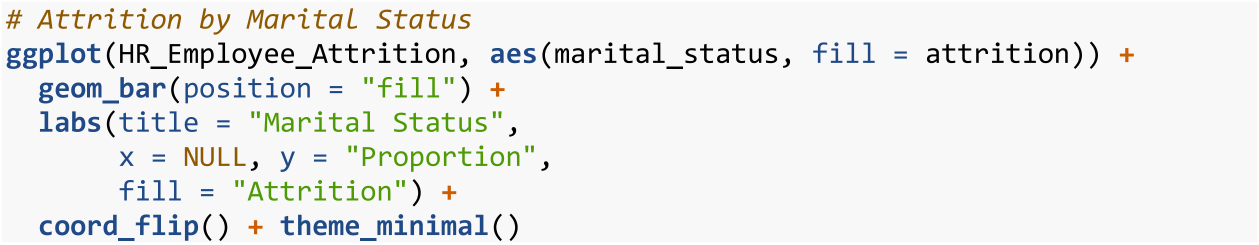


Attrition was highest among employees with job level 1 (about 27%), followed by job level 3 (about 15%).

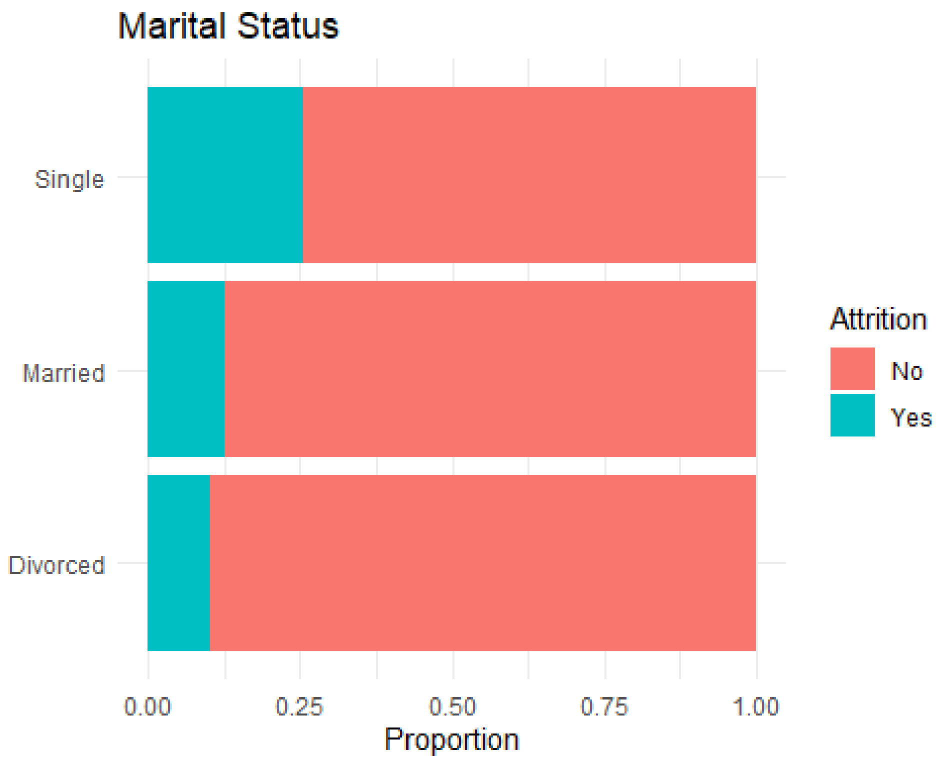
```
# Attrition by Job Role
ggplot(HR_Employee_Attrition, aes(job_role, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Job Role",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



Attrition rate was highest among Sales Representatives (about 40%), followed by lab technicians (about 23%) and HR (23%) respectively.

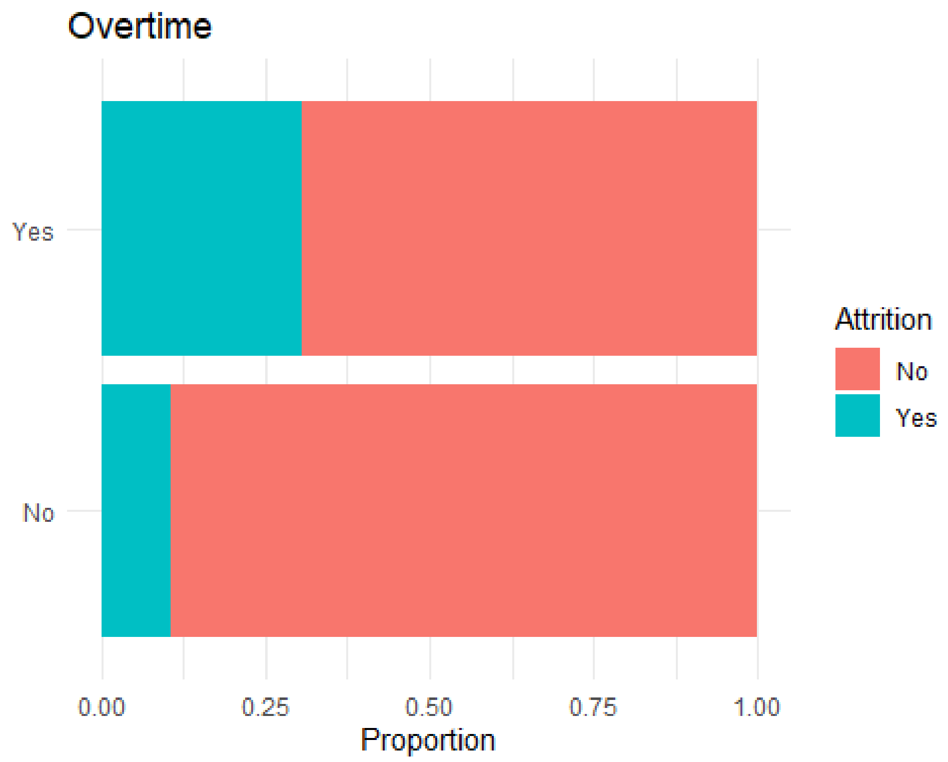


```
# Attrition by Marital Status
ggplot(HR_Employee_Attrition, aes(marital_status, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Marital Status",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



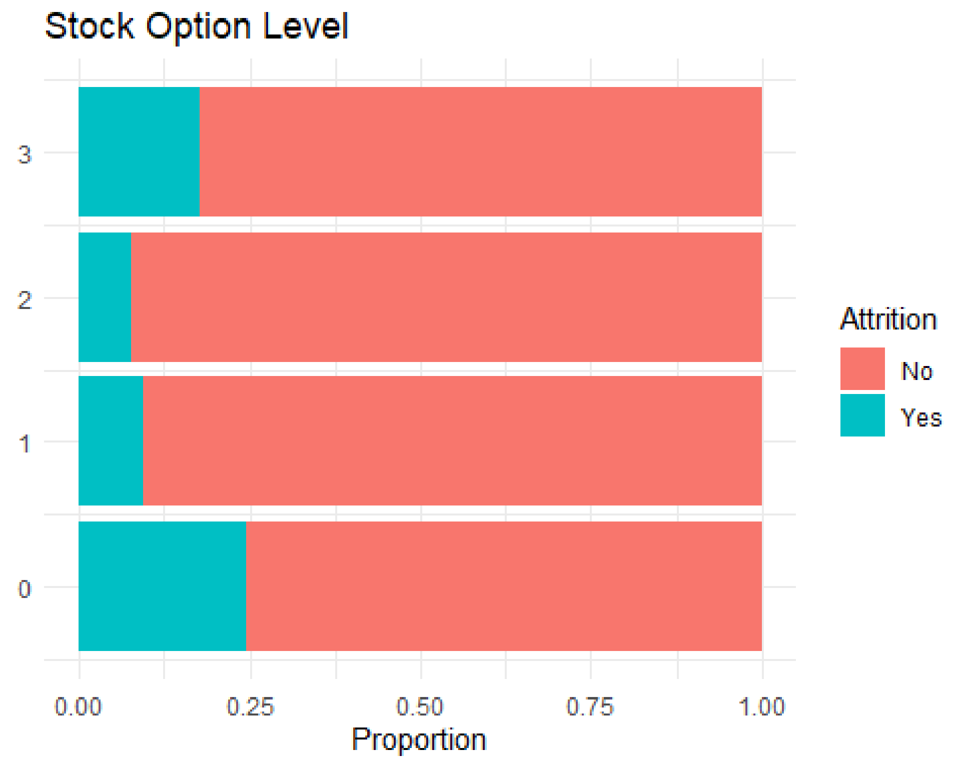
Attrition rate was highest among single employees (about 26%).

```
# Attrition by Overtime
ggplot(HR_Employee_Attrition, aes(over_time, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Overtime",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



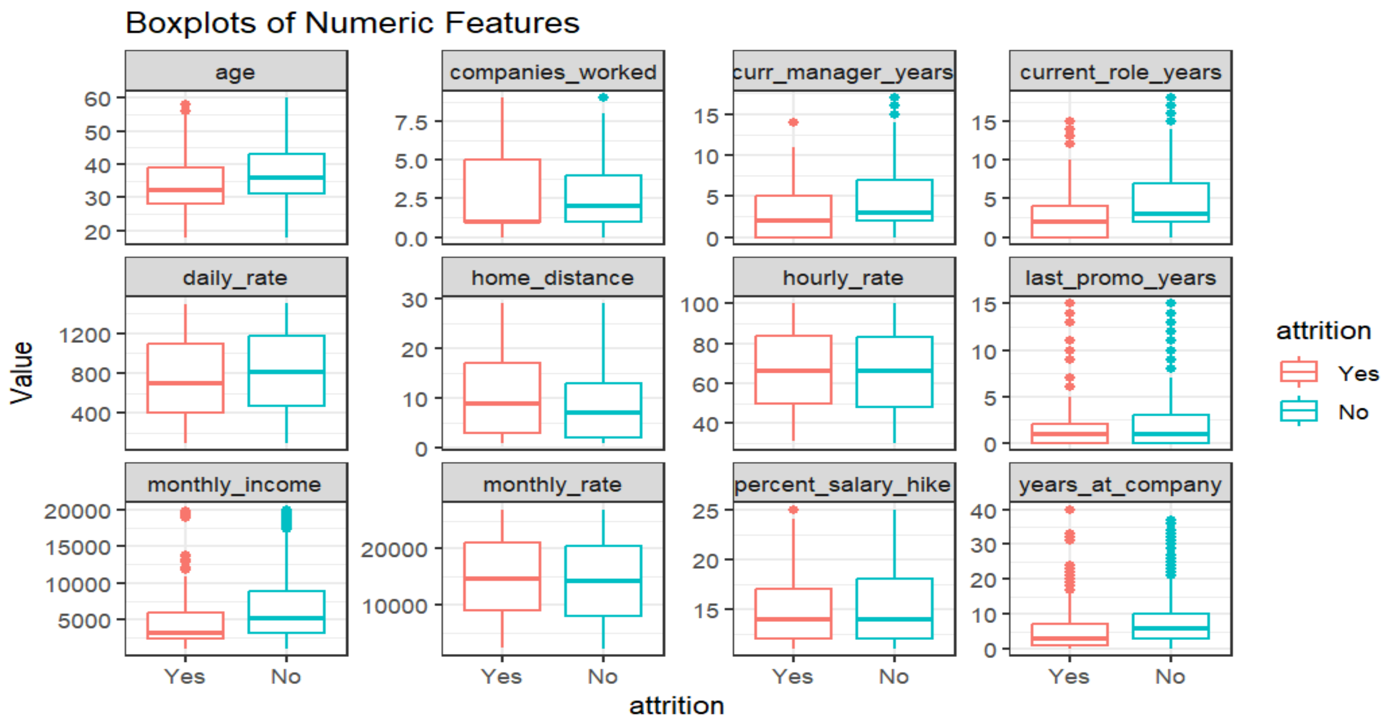
Attrition was highest among the employees who worked overtime (about 30%).

```r
# Attrition by Stock Option Level
ggplot(HR_Employee_Attrition, aes(stock_option_level, fill = attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Stock Option Level",
       x = NULL, y = "Proportion",
       fill = "Attrition") +
  coord_flip() + theme_minimal()
```



Attrition was highest among employees who had stock option level 0 (about 24%), followed by employees with stock option level 3 (about 17%).

```r
## Visualize the distribution of the numeric Features in a single plot

# This will need the numeric features data to be converted into long format

# Select the numeric features plus the response var and convert them to long
format
Num_Untidy <- HR_Employee_Attrition |>
  select(age, daily_rate, home_distance = distance_from_home, hourly_rate,
         monthly_income, monthly_rate, companies_worked =
num_companies_worked,
         percent_salary_hike, years_at_company,
         current_role_years = years_in_current_role,
         last_promo_years = years_since_last_promotion,
         curr_manager_years = years_with_curr_manager, attrition) |>
  gather(key = "Variable", value = "Value", -attrition)

# Create box plots for the numeric features characterized by Attrition
ggplot(Num_Untidy, aes(attrition, as.numeric(Value), colour = attrition)) +
  facet_wrap(~Variable, scales = "free_y") +
  geom_boxplot() +
  labs(title = "Boxplots of Numeric Features",
       y = "Value") +
  theme_bw()
```

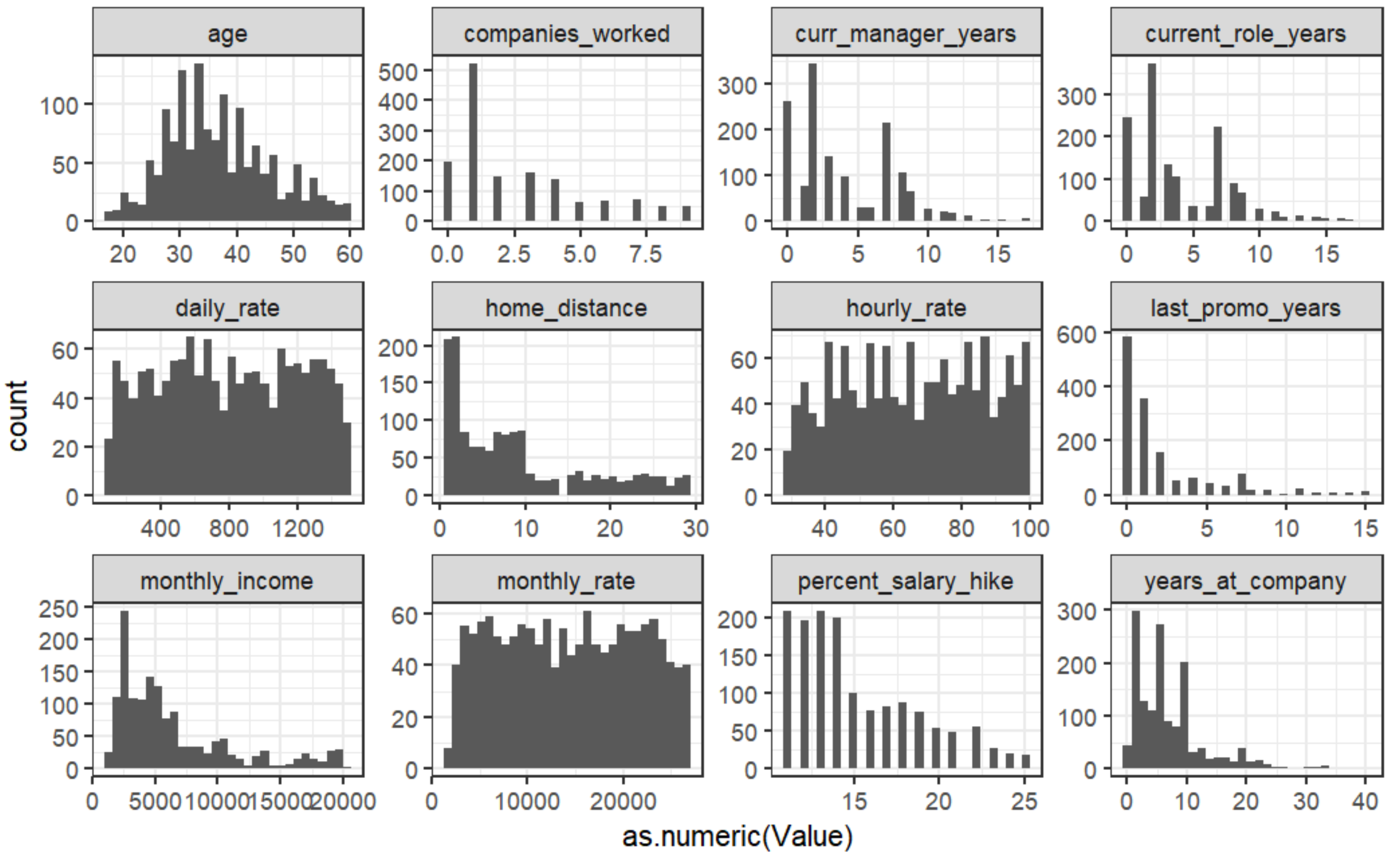


On average;

- Young employees were more likely to leave their jobs for other companies as compared to older employees. Among the employees who left the company, two were much older than the others.
- Attrition rate was high among employees who have worked for more companies, and also among those who have less years at current role.
- Attrition rate was slightly higher among employees who receive lower daily rates.
- Attrition rate was also higher among employees who travel for long distances from home to work.
- Attrition rates were higher among employees who haven't received promotion for the last few years, employees who have been at the company for a few years, employees who have been with their current managers for a few years and employees who get less monthly income.
- Employees who received little percentage of salary hike were also more likely to leave the company.
- Attrition was about 50/50 for Hourly rate and monthly rate.

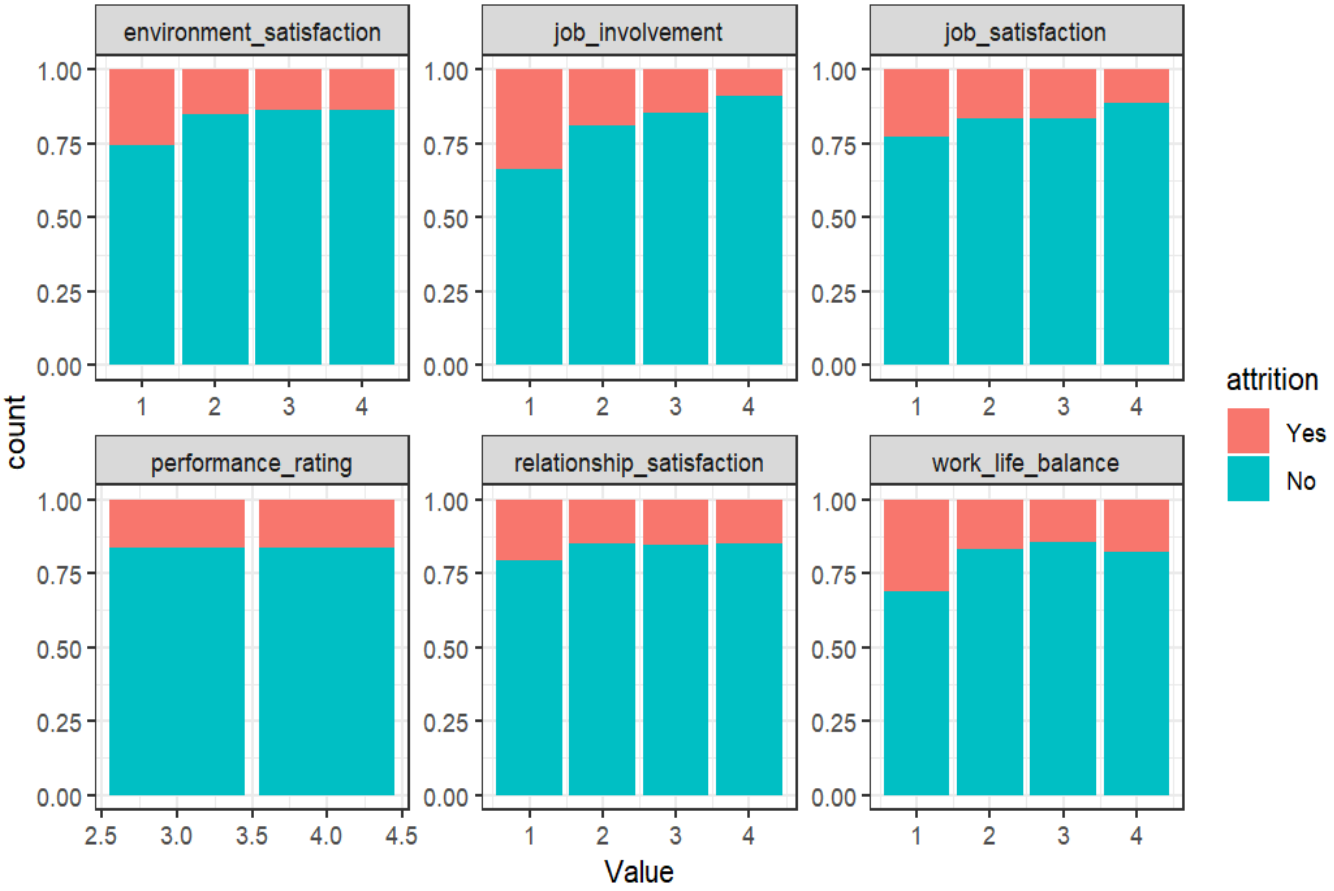
```
# Plot histograms of the numeric features
ggplot(Num_Untidy, aes(as.numeric(Value))) +
  facet_wrap(~Variable, scales = "free") +
  geom_histogram() + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Age nearly follows a normal distribution, while number of companies worked at, years in current role, years since last promotion, years with current manager, monthly income, percentage salary hike and years at company are all right-skewed. Daily rate, hourly rate and monthly rate seem to be multi-modal.

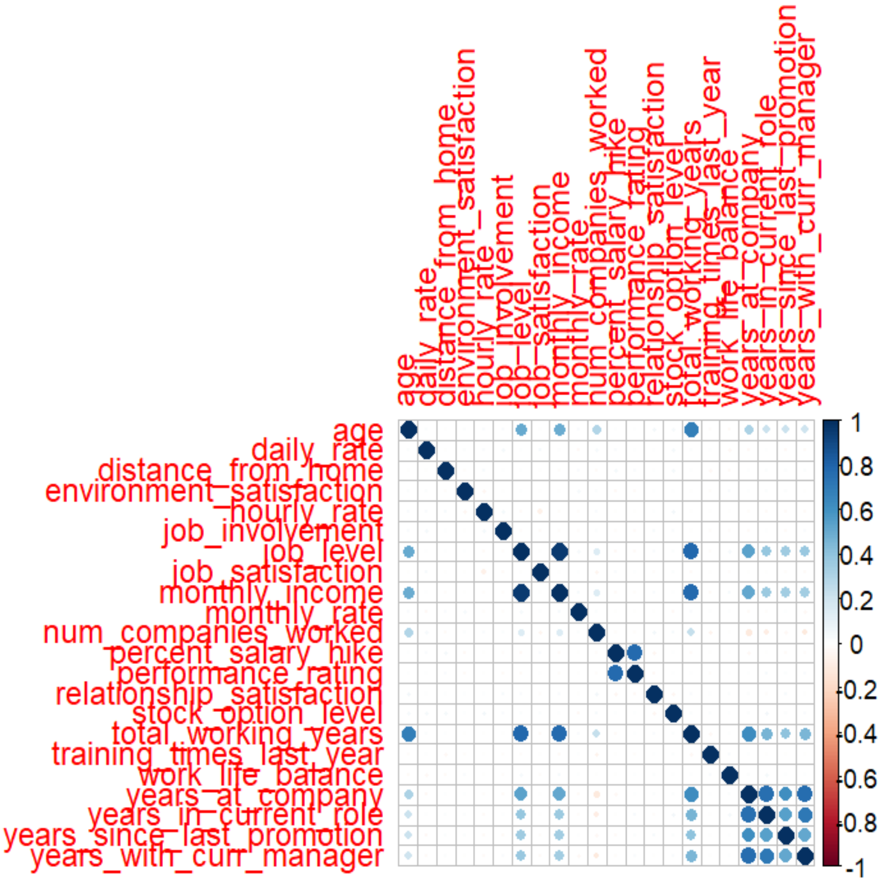
```r
# Job and work-related features
JobWork_Untidy <- HR_Employee_Attrition |>
  select(environment_satisfaction, job_involvement, job_satisfaction,
         performance_rating, relationship_satisfaction, work_life_balance,
         attrition) |>
  gather(key = "Variable", value = "Value", -attrition)

# Create bar plots for the Job & Work related features
ggplot(JobWork_Untidy, aes(Value, fill = attrition)) +
  facet_wrap(~Variable, scales = "free") +
  geom_bar(position = "fill") + theme_bw()
```



Attrition rates were highest among employees who were least satisfied by their job, work environment, work-life balance & relationship at work, and also among employees who had the least job involvement. Attrition rates were nearly the same for Performance ratings 3 and 4.

Correlation Analysis

```
## Check the correlations between numeric features
Num <- HR_Employee_Attrition |> select(where(~is.numeric(.)))
corrplot(cor(Num |> select(-employee_count, -employee_number, -
standard_hours)))
```



There are strong positive correlations between job level & monthly income, percentage salary hike & performance rating, monthly income and total working years, years in current role & years at the company, and years in current role and years with the current manager. There are also moderate positive correlations between age & job level, age & total years of work and years at the company & total years of work.

Feature Engineering

These is a step that precedes model training. It involves preparing the data for Machine Learning models by scaling numeric features and encoding categorical predictors. Feature encoding is important because some algorithms like KNN, SVM and XGBoost cannot handle categorical predictors. Scaling of numeric predictors is also important because some algorithms are sensitive to the magnitude of feature values, and large values tend to dominate various computations, leading to biased parameter estimates or inefficient optimization. I'll begin by partitioning the data into training and test sets, then prepare the two sets separately to prevent information leakage. I will then use the training set to train my models with cross-validation, and validate the models on the independent test set.

```
# First drop features which don't contain much information with regards to
attrition
# (e.g employee count, employee number, over 18 and standard hrs)
```

```
data <- HR_Employee_Attrition |>
  select(-employee_count, -employee_number, -standard_hours, -over18)

# Partition the data into training and validation sets

# Set seed for reproducibility
set.seed(123)

# Split the data (use 80/20 split)
train_index <- createDataPartition(data$attrition, p = 0.80, list = FALSE)
# Assign 80% to training set
training_data <- data[train_index, ]
# Assign the remaining 20% to test set
test_data <- data[-train_index, ]
```

The training set contains 1,177 observations while test set contains 293 observations.

```
## Prepare training data

## Scale the numeric features in training data
training_scaled <- training_data |> mutate_if(is.numeric, ~
as.vector(scale(.)))

## Label-encode the categorical features

# Encode business_travel
training_scaled[["business_travel"]] <-
factor(training_scaled[["business_travel"]],
                                    labels = c(1,2,3),
                                    levels = c("Non-
Travel","Travel_Frequently",
                                               "Travel_Rarely"))

# Encode department
training_scaled[["department"]] <- factor(training_scaled[["department"]],
                                        labels = c(1,2,3),
                                        levels = c("Human Resources",
                                                   "Research & Development",
                                                   "Sales"))

# Encode education field
training_scaled[["education_field"]] <-
factor(training_scaled[["education_field"]],
                                             labels = c(1,2,3,4,5,6),
                                             levels = c("Human Resources",
                                                        "Life Sciences",
                                                        "Marketing",
"Medical",
                                                        "Other",
```

```r
                                                        "Technical Degree"))

# Encode gender into a binary variable (male = 1, female = 0)
training_scaled$gender <- ifelse(training_scaled$gender == "Male", 1, 0)

# Encode job_role
training_scaled[["job_role"]] <- factor(training_scaled[["job_role"]],
                                      labels = c(1,2,3,4,5,6,7,8,9),
                                      levels = c("Healthcare Representative",
                                                 "Human Resources",
                                                 "Laboratory Technician",
                                                 "Manager",
                                                 "Manufacturing Director",
                                                 "Research Director",
                                                 "Research Scientist",
                                                 "Sales Executive",
                                                 "Sales Representative"))

# Encode marital status
training_scaled[["marital_status"]] <-
factor(training_scaled[["marital_status"]],
                                            labels = c(1,2,3),
                                            levels = c("Divorced", "Married",
                                                       "Single"))

# Encode overtime
training_scaled$over_time <- ifelse(training_scaled$over_time == "Yes", 1, 0)

# Convert the encoded factor variables to numeric type
predictors <- training_scaled |> dplyr::select(-attrition) |>
  mutate_if(is.factor, ~ as.numeric(.))

# Add a column with the target variable
training_set <- predictors |> mutate(attrition = training_data$attrition)

## Prepare test data

# Scale the numeric features in test data
test_scaled <- test_data |> mutate_if(is.numeric, ~ as.vector(scale(.)))

# Label encode the categorical features

# Encode business_travel
test_scaled[["business_travel"]] <- factor(test_scaled[["business_travel"]],
                                    labels = c(1,2,3),
                                    levels = c("Non-Travel",
"Travel_Frequently",
                                               "Travel_Rarely"))
```

```r
# Encode department
test_scaled[["department"]] <- factor(test_scaled[["department"]],
                                        labels = c(1,2,3),
                                        levels = c("Human Resources",
                                                   "Research & Development",
                                                   "Sales"))

# Encode education field
test_scaled[["education_field"]] <- factor(test_scaled[["education_field"]],
                                             labels = c(1,2,3,4,5,6),
                                             levels = c("Human Resources",
                                                        "Life Sciences",
                                                        "Marketing",
                                                        "Medical", "Other",
                                                        "Technical Degree"))

# Encode gender into a binary variable (male = 1, female = 0)
test_scaled$gender <- ifelse(test_scaled$gender == "Male", 1, 0)

# Encode job_role
test_scaled[["job_role"]] <- factor(test_scaled[["job_role"]],
                                      labels = c(1,2,3,4,5,6,7,8,9),
                                      levels = c("Healthcare Representative",
                                                 "Human Resources",
                                                 "Laboratory Technician",
                                                 "Manager",
                                                 "Manufacturing Director",
                                                 "Research Director",
                                                 "Research Scientist",
                                                 "Sales Executive",
                                                 "Sales Representative"))

# Encode marital status
test_scaled[["marital_status"]] <- factor(test_scaled[["marital_status"]],
                                            labels = c(1,2,3),
                                            levels = c("Divorced", "Married",
                                                       "Single"))

# Encode overtime into a binary variable
test_scaled$over_time <- ifelse(test_scaled$over_time == "Yes", 1, 0)

# Convert the encoded factor variables to numeric type
predictors <- test_scaled |> select(-attrition) |>
  mutate_if(is.factor, ~ as.numeric(.))

# Add a column with the target variable
test_set <- predictors |> mutate(attrition = test_data$attrition)
```

Feature Selection

Logistic Regression model will be used to perform stepwise feature selection.

```
# Fit a logistic Regression model
model <- glm(attrition ~., family = binomial(logit), data = training_set)
# Have a model summary to check for the predictors which are significant
summary(model)
```

```
##
## Call:
## glm(formula = attrition ~ ., family = binomial(logit), data =
training_set)
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -6.27342    0.86576  -7.246 4.29e-13 ***
## age                        -0.23428    0.13382  -1.751 0.079990 .
## business_travel             0.05086    0.14993   0.339 0.734437
## daily_rate                 -0.16947    0.09370  -1.809 0.070492 .
## department                  0.79828    0.28235   2.827 0.004695 **
## distance_from_home          0.28943    0.09305   3.111 0.001867 **
## education                  -0.02259    0.09318  -0.242 0.808467
## education_field             0.03965    0.07187   0.552 0.581149
## environment_satisfaction   -0.40423    0.09536  -4.239 2.24e-05 ***
## gender                      0.45986    0.19875   2.314 0.020680 *
## hourly_rate                -0.06530    0.09485  -0.688 0.491147
## job_involvement            -0.36946    0.09357  -3.948 7.86e-05 ***
## job_level                  -0.22454    0.34115  -0.658 0.510419
## job_role                   -0.07863    0.05600  -1.404 0.160295
## job_satisfaction           -0.45047    0.09539  -4.722 2.33e-06 ***
## marital_status              0.68634    0.18721   3.666 0.000246 ***
## monthly_income             -0.20674    0.34751  -0.595 0.551894
## monthly_rate                0.01711    0.09560   0.179 0.857958
## num_companies_worked        0.43621    0.10107   4.316 1.59e-05 ***
## over_time                   1.85456    0.20379   9.101  < 2e-16 ***
## percent_salary_hike        -0.14783    0.15274  -0.968 0.333113
## performance_rating          0.07078    0.15453   0.458 0.646915
## relationship_satisfaction  -0.23085    0.09527  -2.423 0.015387 *
## stock_option_level         -0.05352    0.13380  -0.400 0.689147
## total_working_years        -0.59015    0.24497  -2.409 0.015993 *
## training_times_last_year   -0.08118    0.09728  -0.835 0.403989
## work_life_balance          -0.23981    0.09179  -2.613 0.008985 **
## years_at_company            0.70085    0.25999   2.696 0.007024 **
## years_in_current_role      -0.59501    0.17490  -3.402 0.000669 ***
## years_since_last_promotion  0.48827    0.14136   3.454 0.000552 ***
## years_with_curr_manager    -0.46872    0.17990  -2.605 0.009175 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1040.54  on 1176  degrees of freedom
## Residual deviance:  748.05  on 1146  degrees of freedom
## AIC: 810.05
##
## Number of Fisher Scoring iterations: 6
```

The significant predictors are age, department, home distance, environmental satisfaction, gender, job involvement, job satisfaction, marital status, number of companies worked for, working overtime, relationship satisfaction, work-life balance, years at the company, years in current role, years since last promotion and years with the current manager (p < .05).

```
# Perform stepwise regression for feature selection
stepwise_model <- step(model)
```

```
## Start:  AIC=810.05
## attrition ~ age + business_travel + daily_rate + department +
##     distance_from_home + education + education_field +
environment_satisfaction +
##     gender + hourly_rate + job_involvement + job_level + job_role +
##     job_satisfaction + marital_status + monthly_income + monthly_rate +
##     num_companies_worked + over_time + percent_salary_hike +
##     performance_rating + relationship_satisfaction + stock_option_level +
##     total_working_years + training_times_last_year + work_life_balance +
##     years_at_company + years_in_current_role + years_since_last_promotion
+
##     years_with_curr_manager
##
##                              Df Deviance    AIC
## - monthly_rate                1   748.08 808.08
## - education                   1   748.11 808.11
## - business_travel             1   748.16 808.16
## - stock_option_level          1   748.21 808.21
## - performance_rating          1   748.26 808.26
## - education_field             1   748.35 808.35
## - monthly_income              1   748.40 808.40
## - job_level                   1   748.48 808.48
## - hourly_rate                 1   748.52 808.52
## - training_times_last_year    1   748.75 808.75
## - percent_salary_hike         1   748.99 808.99
## - job_role                    1   750.01 810.01
## <none>                            748.05 810.05
## - age                         1   751.21 811.21
## - daily_rate                  1   751.34 811.34
## - gender                      1   753.53 813.53
## - relationship_satisfaction   1   753.95 813.95
## - total_working_years         1   754.20 814.20
## - years_with_curr_manager     1   754.74 814.74
## - work_life_balance          1   754.85 814.85
## - years_at_company           1   755.10 815.10
## - department                 1   756.41 816.41
## - distance_from_home         1   757.58 817.58
## - years_in_current_role      1   759.73 819.73
## - years_since_last_promotion  1   760.32 820.32
## - marital_status             1   762.31 822.31
## - job_involvement            1   763.87 823.87
## - num_companies_worked       1   766.28 826.28
## - environment_satisfaction   1   766.40 826.40
## - job_satisfaction           1   771.03 831.03
## - over_time                  1   837.98 897.98
## 
## Step:  AIC=808.08
## attrition ~ age + business_travel + daily_rate + department + 
##     distance_from_home + education + education_field + 
environment_satisfaction + 
##     gender + hourly_rate + job_involvement + job_level + job_role + 
##     job_satisfaction + marital_status + monthly_income + 
num_companies_worked + 
##     over_time + percent_salary_hike + performance_rating + 
relationship_satisfaction + 
##     stock_option_level + total_working_years + training_times_last_year + 
##     work_life_balance + years_at_company + years_in_current_role + 
##     years_since_last_promotion + years_with_curr_manager
## 
##                              Df Deviance    AIC
## - education                  1   748.14 806.14
## - business_travel            1   748.20 806.20
## - stock_option_level         1   748.25 806.25
## - performance_rating         1   748.28 806.28
## - education_field            1   748.38 806.38
## - monthly_income             1   748.44 806.44
## - job_level                  1   748.50 806.50
## - hourly_rate                1   748.56 806.56
## - training_times_last_year   1   748.79 806.79
## - percent_salary_hike        1   749.01 807.01
## - job_role                   1   750.04 808.04
## <none>                           748.08 808.08
## - age                        1   751.25 809.25
## - daily_rate                 1   751.42 809.42
## - gender                     1   753.54 811.54
## - relationship_satisfaction  1   753.97 811.97
## - total_working_years        1   754.23 812.23
## - years_with_curr_manager    1   754.79 812.79
## - work_life_balance          1   754.88 812.88
## - years_at_company           1   755.12 813.12
## - department                 1   756.43 814.43
## - distance_from_home         1   757.72 815.72
## - years_in_current_role      1   759.77 817.77
## - years_since_last_promotion  1   760.40 818.40
## - marital_status              1   762.31 820.31
## - job_involvement             1   763.88 821.88
## - num_companies_worked        1   766.29 824.29
## - environment_satisfaction    1   766.43 824.43
## - job_satisfaction            1   771.03 829.03
## - over_time                   1   838.03 896.03
## 
## Step:  AIC=806.14
## attrition ~ age + business_travel + daily_rate + department + 
##     distance_from_home + education_field + environment_satisfaction + 
##     gender + hourly_rate + job_involvement + job_level + job_role + 
##     job_satisfaction + marital_status + monthly_income + 
num_companies_worked + 
##     over_time + percent_salary_hike + performance_rating + 
relationship_satisfaction + 
##     stock_option_level + total_working_years + training_times_last_year + 
##     work_life_balance + years_at_company + years_in_current_role + 
##     years_since_last_promotion + years_with_curr_manager
## 
##                              Df Deviance    AIC
## - business_travel             1   748.27 804.27
## - stock_option_level          1   748.32 804.32
## - performance_rating          1   748.35 804.35
## - education_field             1   748.45 804.45
## - monthly_income              1   748.52 804.52
## - job_level                   1   748.55 804.55
## - hourly_rate                 1   748.61 804.61
## - training_times_last_year    1   748.85 804.85
## - percent_salary_hike         1   749.09 805.09
## - job_role                    1   750.12 806.12
## <none>                            748.14 806.14
## - daily_rate                  1   751.49 807.49
## - age                         1   751.53 807.53
## - gender                      1   753.64 809.64
## - relationship_satisfaction   1   754.01 810.01
## - total_working_years         1   754.35 810.35
## - years_with_curr_manager     1   754.89 810.89
## - work_life_balance           1   754.93 810.93
## - years_at_company            1   755.21 811.21
## - department                  1   756.52 812.52
## - distance_from_home          1   757.72 813.72
## - years_in_current_role       1   759.82 815.82
## - years_since_last_promotion  1   760.41 816.41
## - marital_status              1   762.32 818.32
## - job_involvement             1   764.07 820.07
## - num_companies_worked        1   766.30 822.30
## - environment_satisfaction    1   766.45 822.45
## - job_satisfaction            1   771.04 827.04
## - over_time                   1   838.15 894.15
```

```
##
## Step:  AIC=804.27
## attrition ~ age + daily_rate + department + distance_from_home +
##     education_field + environment_satisfaction + gender + hourly_rate +
##     job_involvement + job_level + job_role + job_satisfaction +
##     marital_status + monthly_income + num_companies_worked +
##     over_time + percent_salary_hike + performance_rating +
relationship_satisfaction +
##     stock_option_level + total_working_years + training_times_last_year +
##     work_life_balance + years_at_company + years_in_current_role +
##     years_since_last_promotion + years_with_curr_manager
##
##                              Df Deviance    AIC
## - stock_option_level          1   748.44 802.44
## - performance_rating          1   748.49 802.49
## - education_field             1   748.57 802.57
## - monthly_income              1   748.64 802.64
## - job_level                   1   748.69 802.69
## - hourly_rate                 1   748.72 802.72
## - training_times_last_year    1   748.97 802.97
## - percent_salary_hike         1   749.26 803.26
## - job_role                    1   750.23 804.23
## <none>                            748.27 804.27
## - daily_rate                  1   751.63 805.63
## - age                         1   751.68 805.68
## - gender                      1   753.74 807.74
## - relationship_satisfaction   1   754.20 808.20
## - total_working_years         1   754.37 808.37
## - years_with_curr_manager     1   755.15 809.15
## - work_life_balance           1   755.18 809.18
## - years_at_company            1   755.28 809.28
## - department                  1   756.63 810.63
## - distance_from_home          1   757.75 811.75
## - years_in_current_role       1   759.83 813.83
## - years_since_last_promotion  1   760.42 814.42
## - marital_status              1   762.47 816.47
## - job_involvement             1   764.13 818.13
## - num_companies_worked        1   766.39 820.39
## - environment_satisfaction    1   766.47 820.47
## - job_satisfaction            1   771.31 825.31
## - over_time                   1   838.40 892.40
##
## Step:  AIC=802.44
## attrition ~ age + daily_rate + department + distance_from_home +
##     education_field + environment_satisfaction + gender + hourly_rate +
##     job_involvement + job_level + job_role + job_satisfaction +
##     marital_status + monthly_income + num_companies_worked +
##     over_time + percent_salary_hike + performance_rating +
relationship_satisfaction +
##     total_working_years + training_times_last_year + work_life_balance +
##     years_at_company + years_in_current_role + years_since_last_promotion
+
##     years_with_curr_manager
##
##                              Df Deviance    AIC
## - performance_rating          1   748.68 800.68
## - education_field             1   748.76 800.76
## - monthly_income              1   748.80 800.80
## - job_level                   1   748.87 800.87
## - hourly_rate                 1   748.91 800.91
## - training_times_last_year    1   749.16 801.16
## - percent_salary_hike         1   749.47 801.47
## - job_role                    1   750.38 802.38
## <none>                            748.44 802.44
## - daily_rate                  1   751.83 803.83
## - age                         1   751.89 803.89
## - gender                      1   753.82 805.82
## - relationship_satisfaction   1   754.26 806.26
## - total_working_years         1   754.50 806.50
## - work_life_balance           1   755.36 807.36
## - years_with_curr_manager     1   755.39 807.39
## - years_at_company            1   755.48 807.48
## - department                  1   756.72 808.72
## - distance_from_home          1   757.82 809.82
## - years_in_current_role       1   759.99 811.99
## - years_since_last_promotion  1   760.59 812.59
## - job_involvement             1   764.24 816.24
## - num_companies_worked        1   766.46 818.46
## - environment_satisfaction    1   766.62 818.62
## - job_satisfaction            1   771.83 823.83
## - marital_status              1   779.05 831.05
## - over_time                   1   838.42 890.42
##
## Step:  AIC=800.68
## attrition ~ age + daily_rate + department + distance_from_home +
##     education_field + environment_satisfaction + gender + hourly_rate +
##     job_involvement + job_level + job_role + job_satisfaction +
##     marital_status + monthly_income + num_companies_worked +
##     over_time + percent_salary_hike + relationship_satisfaction +
##     total_working_years + training_times_last_year + work_life_balance +
##     years_at_company + years_in_current_role + years_since_last_promotion
+
##     years_with_curr_manager
##
##                              Df Deviance    AIC
## - education_field             1   748.99 798.99
## - monthly_income              1   749.07 799.07
## - job_level                   1   749.10 799.10
## - hourly_rate                 1   749.16 799.16
## - training_times_last_year    1   749.39 799.39
## - percent_salary_hike         1   749.69 799.69
## - job_role                    1   750.65 800.65
## <none>                            748.68 800.68
## - daily_rate                  1   752.03 802.03
## - age                         1   752.22 802.22
## - gender                      1   754.02 804.02
## - relationship_satisfaction   1   754.46 804.46
## - total_working_years         1   754.62 804.62
## - work_life_balance           1   755.62 805.62
## - years_with_curr_manager     1   755.64 805.64
## - years_at_company            1   755.67 805.67
## - department                  1   757.00 807.00
## - distance_from_home          1   757.92 807.92
## - years_in_current_role       1   760.14 810.14
## - years_since_last_promotion  1   760.96 810.96
## - job_involvement             1   764.44 814.44
## - num_companies_worked        1   766.58 816.58
## - environment_satisfaction    1   766.94 816.94
## - job_satisfaction            1   772.05 822.05
## - marital_status              1   779.21 829.21
## - over_time                   1   839.01 889.01
##
## Step:  AIC=798.99
## attrition ~ age + daily_rate + department + distance_from_home +
##     environment_satisfaction + gender + hourly_rate + job_involvement +
##     job_level + job_role + job_satisfaction + marital_status +
##     monthly_income + num_companies_worked + over_time +
percent_salary_hike +
##     relationship_satisfaction + total_working_years +
training_times_last_year +
##     work_life_balance + years_at_company + years_in_current_role +
##     years_since_last_promotion + years_with_curr_manager
##
##                              Df Deviance    AIC
## - monthly_income              1   749.37 797.37
## - job_level                   1   749.41 797.41
## - hourly_rate                 1   749.48 797.48
## - training_times_last_year    1   749.66 797.66
## - percent_salary_hike         1   749.97 797.97
## <none>                            748.99 798.99
## - job_role                    1   750.99 798.99
## - daily_rate                  1   752.25 800.25
## - age                         1   752.62 800.62
## - gender                      1   754.35 802.35
## - relationship_satisfaction   1   754.75 802.75
## - total_working_years         1   754.88 802.88
## - years_with_curr_manager     1   755.83 803.83
## - work_life_balance           1   755.87 803.87
## - years_at_company            1   755.92 803.92
## - department                  1   757.37 805.37
## - distance_from_home          1   758.17 806.17
## - years_in_current_role       1   760.39 808.39
## - years_since_last_promotion  1   761.29 809.29
## - job_involvement             1   764.80 812.80
## - num_companies_worked        1   766.98 814.98
## - environment_satisfaction    1   767.00 815.00
## - job_satisfaction            1   772.64 820.64
## - marital_status              1   779.58 827.58
## - over_time                   1   839.22 887.22
##
## Step:  AIC=797.37
## attrition ~ age + daily_rate + department + distance_from_home +
##     environment_satisfaction + gender + hourly_rate + job_involvement +
##     job_level + job_role + job_satisfaction + marital_status +
##     num_companies_worked + over_time + percent_salary_hike +
##     relationship_satisfaction + total_working_years +
training_times_last_year +
##     work_life_balance + years_at_company + years_in_current_role +
##     years_since_last_promotion + years_with_curr_manager
##
##                              Df Deviance    AIC
## - hourly_rate                 1   749.91 795.91
## - training_times_last_year    1   750.05 796.05
## - percent_salary_hike         1   750.39 796.39
## <none>                            749.37 797.37
## - job_role                    1   751.47 797.47
## - daily_rate                  1   752.65 798.65
## - age                         1   753.00 799.00
## - job_level                   1   753.78 799.78
## - gender                      1   754.66 800.66
## - relationship_satisfaction   1   755.14 801.14
## - total_working_years         1   755.73 801.73
## - years_with_curr_manager     1   755.95 801.95
## - years_at_company            1   756.15 802.15
## - work_life_balance           1   756.20 802.20
## - department                  1   758.13 804.13
## - distance_from_home          1   758.92 804.92
## - years_in_current_role       1   760.61 806.61
## - years_since_last_promotion  1   761.48 807.48
## - job_involvement             1   765.25 811.25
## - environment_satisfaction    1   767.26 813.26
## - num_companies_worked        1   767.30 813.30
## - job_satisfaction            1   772.78 818.78
## - marital_status              1   780.18 826.18
## - over_time                   1   839.37 885.37
##
## Step:  AIC=795.91
## attrition ~ age + daily_rate + department + distance_from_home +
##     environment_satisfaction + gender + job_involvement + job_level +
##     job_role + job_satisfaction + marital_status + num_companies_worked +
##     over_time + percent_salary_hike + relationship_satisfaction + 
##     total_working_years + training_times_last_year + work_life_balance + 
##     years_at_company + years_in_current_role + years_since_last_promotion 
+ 
##     years_with_curr_manager
## 
##                              Df Deviance    AIC
## - training_times_last_year    1   750.60 794.60
## - percent_salary_hike         1   750.92 794.92
## - job_role                    1   751.91 795.91
## <none>                            749.91 795.91
## - daily_rate                  1   753.19 797.19
## - age                         1   753.64 797.64
## - job_level                   1   754.31 798.31
## - gender                      1   755.36 799.36
## - relationship_satisfaction   1   755.58 799.58
## - total_working_years         1   756.30 800.30
## - years_with_curr_manager     1   756.47 800.47
## - years_at_company            1   756.60 800.60
## - work_life_balance           1   756.74 800.74
## - department                  1   758.51 802.51
## - distance_from_home          1   759.37 803.37
## - years_in_current_role       1   761.11 805.11
## - years_since_last_promotion  1   762.23 806.23
## - job_involvement             1   766.21 810.21
## - environment_satisfaction    1   767.70 811.70
## - num_companies_worked        1   767.95 811.95
## - job_satisfaction            1   773.00 817.00
## - marital_status              1   780.77 824.77
## - over_time                   1   839.53 883.53
## 
## Step:  AIC=794.6
## attrition ~ age + daily_rate + department + distance_from_home + 
##     environment_satisfaction + gender + job_involvement + job_level + 
##     job_role + job_satisfaction + marital_status + num_companies_worked + 
##     over_time + percent_salary_hike + relationship_satisfaction + 
##     total_working_years + work_life_balance + years_at_company + 
##     years_in_current_role + years_since_last_promotion + 
years_with_curr_manager
## 
##                              Df Deviance    AIC
## - percent_salary_hike         1   751.63 793.63
## - job_role                    1   752.41 794.41
## <none>                            750.60 794.60
## - daily_rate                  1   753.96 795.96
## - age                         1   754.43 796.43
## - job_level                   1   754.86 796.86
## - relationship_satisfaction   1   756.28 798.28
## - gender                      1   756.29 798.29
## - years_with_curr_manager     1   756.91 798.91
## - total_working_years         1   757.12 799.12
## - years_at_company            1   757.23 799.23
## - work_life_balance           1   757.57 799.57
## - department                  1   758.88 800.88
## - distance_from_home          1   760.07 802.07
## - years_in_current_role       1   761.78 803.78
## - years_since_last_promotion  1   762.93 804.93
## - job_involvement             1   766.88 808.88
## - environment_satisfaction    1   768.50 810.50
## - num_companies_worked        1   769.19 811.19
## - job_satisfaction            1   773.41 815.41
## - marital_status              1   781.07 823.07
## - over_time                   1   841.62 883.62
## 
## Step:  AIC=793.63
## attrition ~ age + daily_rate + department + distance_from_home + 
##     environment_satisfaction + gender + job_involvement + job_level + 
##     job_role + job_satisfaction + marital_status + num_companies_worked + 
##     over_time + relationship_satisfaction + total_working_years + 
##     work_life_balance + years_at_company + years_in_current_role + 
##     years_since_last_promotion + years_with_curr_manager
## 
##                              Df Deviance    AIC
## - job_role                    1   753.44 793.44
## <none>                            751.63 793.63
## - daily_rate                  1   755.18 795.18
## - age                         1   755.50 795.50
## - job_level                   1   755.74 795.74
## - relationship_satisfaction   1   757.24 797.24
## - gender                      1   757.50 797.50
## - years_with_curr_manager     1   757.93 797.93
## - years_at_company            1   758.24 798.24
## - total_working_years         1   758.27 798.27
## - work_life_balance           1   758.48 798.48
## - department                  1   759.79 799.79
## - distance_from_home          1   760.92 800.92
## - years_in_current_role       1   762.82 802.82
## - years_since_last_promotion  1   764.17 804.17
## - job_involvement             1   767.72 807.72
## - environment_satisfaction    1   769.31 809.31
## - num_companies_worked        1   770.31 810.31
## - job_satisfaction            1   774.26 814.26
## - marital_status              1   782.17 822.17
## - over_time                   1   842.49 882.49
## 
## Step:  AIC=793.44
## attrition ~ age + daily_rate + department + distance_from_home + 
##     environment_satisfaction + gender + job_involvement + job_level + 
##     job_satisfaction + marital_status + num_companies_worked + 
##     over_time + relationship_satisfaction + total_working_years + 
##     work_life_balance + years_at_company + years_in_current_role +
##     years_since_last_promotion + years_with_curr_manager
##
##                               Df Deviance    AIC
## <none>                            753.44 793.44
## - job_level                   1   756.61 794.61
## - daily_rate                  1   756.82 794.82
## - age                         1   757.40 795.40
## - relationship_satisfaction   1   759.23 797.23
## - years_with_curr_manager     1   759.63 797.63
## - gender                      1   759.65 797.65
## - years_at_company            1   759.77 797.77
## - total_working_years         1   759.90 797.90
## - work_life_balance           1   760.59 798.59
## - department                  1   760.94 798.94
## - distance_from_home          1   762.57 800.57
## - years_in_current_role       1   764.43 802.43
## - years_since_last_promotion  1   765.92 803.92
## - job_involvement             1   770.43 808.43
## - environment_satisfaction    1   771.05 809.05
## - num_companies_worked        1   772.18 810.18
## - job_satisfaction            1   776.28 814.28
## - marital_status              1   784.00 822.00
## - over_time                   1   842.76 880.76
```

```
summary(stepwise_model)
```

```
##
## Call:
## glm(formula = attrition ~ age + daily_rate + department +
distance_from_home +
##     environment_satisfaction + gender + job_involvement + job_level +
##     job_satisfaction + marital_status + num_companies_worked +
##     over_time + relationship_satisfaction + total_working_years +
##     work_life_balance + years_at_company + years_in_current_role +
##     years_since_last_promotion + years_with_curr_manager, family =
binomial(logit),
##     data = training_set)
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -5.91434    0.58263 -10.151  < 2e-16 ***
## age                         -0.25864    0.13229  -1.955 0.050577 .
## daily_rate                  -0.17062    0.09316  -1.831 0.067034 .
## department                   0.50571    0.18655   2.711 0.006711 **
## distance_from_home           0.27905    0.09160   3.046 0.002316 **
## environment_satisfaction    -0.39185    0.09428  -4.156 3.24e-05 ***
## gender                       0.48234    0.19619   2.459 0.013951 *
## job_involvement             -0.37905    0.09259  -4.094 4.24e-05 ***
## job_level                   -0.32696    0.18526  -1.765 0.077588 .
## job_satisfaction           -0.44309    0.09403  -4.712 2.45e-06 ***
## marital_status              0.73413    0.13799   5.320 1.04e-07 ***
## num_companies_worked        0.44040    0.10038   4.387 1.15e-05 ***
## over_time                   1.82812    0.20102   9.094  < 2e-16 ***
## relationship_satisfaction  -0.22665    0.09438  -2.402 0.016328 *
## total_working_years        -0.59602    0.24183  -2.465 0.013716 *
## work_life_balance          -0.24488    0.09135  -2.681 0.007348 **
## years_at_company            0.66222    0.25993   2.548 0.010845 *
## years_in_current_role      -0.57618    0.17475  -3.297 0.000977 ***
## years_since_last_promotion  0.49154    0.14095   3.487 0.000488 ***
## years_with_curr_manager    -0.44824    0.17893  -2.505 0.012244 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1040.54  on 1176  degrees of freedom
## Residual deviance:  753.44  on 1157  degrees of freedom
## AIC: 793.44
##
## Number of Fisher Scoring iterations: 6
```

The predictors selected by the stepwise regression model are age, department, distance_from_home, environment_satisfaction, gender, job_involvement, job_level, job_satisfaction, marital_status, number_of_companies_worked_at, over_time, percent_salary_hike, relationship_satisfaction, stock_option_level, training_times_last_year, work_life_balance, years_at_company, years_in_current_role, years_since_last_promotion, and years_with_current_manager.

I'll only use the selected features to train the models.

```
# Perform feature selection based on stepwise regression results
training_set <- training_set |> select(age, department, distance_from_home,
                            environment_satisfaction, gender,
job_involvement,
                            job_level, job_satisfaction, marital_status,
                            num_companies_worked, over_time,
percent_salary_hike,
                            relationship_satisfaction, stock_option_level,
                            training_times_last_year, work_life_balance,
                            years_at_company, years_in_current_role,
                            years_since_last_promotion,
years_with_curr_manager,
                            attrition)

test_set <- test_set |> select(age, department, distance_from_home,
                            environment_satisfaction, gender,
job_involvement,
                            job_level, job_satisfaction, marital_status,
                            num_companies_worked, over_time,
percent_salary_hike,
                            relationship_satisfaction, stock_option_level,
                            training_times_last_year, work_life_balance,
                            years_at_company, years_in_current_role,
                            years_since_last_promotion,
years_with_curr_manager,
                            attrition)
```

Model Training

I’ll try four different algorithms i.e. Logistic Regression, Random Forest, SVM and XGBoost. When training the models, I’ll perform hyperparameter tuning with cross-validation in order to obtain optimal solutions. Cross-validation helps to evaluate how the models would generalize on new data, and also helps to reduce overfitting. When performing cross-validation, I’ll use same seed number for each cross-validation process to ensure that the model results can directly be compared.

```
# Define classification task
AttritionTask <- makeClassifTask(data = training_set, target = "attrition")

## Warning in makeTask(type = type, data = data, weights = weights, blocking
=
## blocking, : Provided data is not a pure data.frame but from class tbl_df,
hence
## it will be converted.
```

Logistic Regression model

```
# Define learner
logReg <- makeLearner("classif.logreg", predict.type = "prob")

# Train the model
logRegModel <- train(logReg, AttritionTask)

# Cross-validate the model training process

# Set seed for reproducibility
set.seed(123)

# Define a 6-fold resampling description with 50 iterations
kFold <- makeResampleDesc(method = "RepCV", folds = 6,
                          reps = 40, stratify = TRUE)

# Cross-validate
logRegCV <- resample(learner = logReg, task = AttritionTask,
                     resampling = kFold,
                     measures = list(mmce, acc, fpr, fnr),
                     show.info = FALSE)
```

```
# View cross_validation results
logRegCV$aggr

## mmce.test.mean  acc.test.mean  fpr.test.mean  fnr.test.mean
##     0.13003921     0.86996079     0.64372480     0.03114837
```

The Logistic Regression model generalizes well. It has an accuracy of 86.69% and a False Negative Rate of 3.11%. However, FPR is very high (64.37%).

Random Forest

```
# Define learner
rf_learner <- makeLearner("classif.randomForest", predict.type = "prob")

# Define hyperparameter space for tuning
rf_ParamSpace <- makeParamSet(makeIntegerParam("ntree", lower = 300,
                                                  upper = 500),
                                 makeIntegerParam("mtry", lower = 5,
                                                  upper = 15),
                                 makeIntegerParam("nodesize", lower = 2,
                                                  upper = 7),
                                 makeIntegerParam("maxnodes", lower = 10,
                                                  upper = 50))

# Define search strategy to use random search with 200 iterations
randSearch <- makeTuneControlRandom(maxit = 200)

# Define a 6-fold resampling description
cvForTuning <- makeResampleDesc("CV", iters = 6, stratify = TRUE)
```

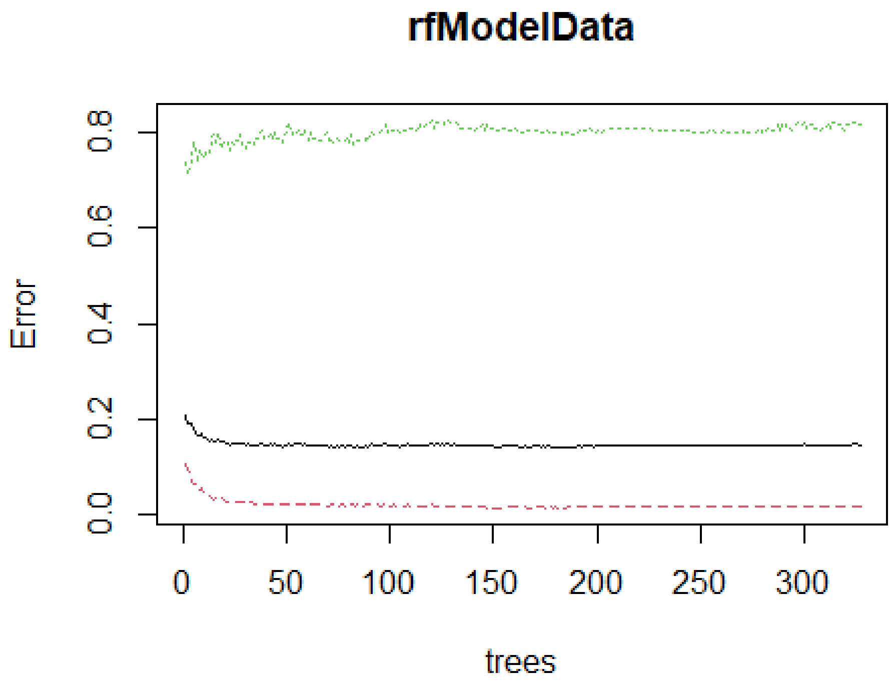
I’ll use parallelization to speed up the process because a large number of hyperparameter combinations will be tried.

```
# Begin parallelization
parallelStartSocket(cpus = detectCores())

## Starting parallelization in mode=socket with cpus=8.

# Set random seed for reproducibility
set.seed(123)

# Perform hyperparameter tuning
tuned_rf_Pars <- tuneParams(learner = rf_learner, task = AttritionTask,
                            resampling = cvForTuning,
                            par.set = rf_ParamSpace,
                            control = randSearch,
                            measures = list(mmce, acc, fpr, fnr),
                            show.info = FALSE)
```

```
## Exporting objects to slaves for mode socket: .mlr.slave.options

## Mapping in parallel: mode = socket; level = mlr.tuneParams; cpus = 8;
elements = 200.

# Stop parallelization
parallelStop()

## Stopped parallelization. All cleaned up.

# View cross-validation results
tuned_rf_Pars

## Tune result:
## Op. pars: ntree=327; mtry=13; nodesize=3; maxnodes=50
##
mmce.test.mean=0.1401735,acc.test.mean=0.8598265,fpr.test.mean=0.7837702,fnr.
test.mean=0.0161924
```

The RF classifier has a training accuracy of 85.98% which is good. However, this model has a very high FPR (78.38%) which is worse. The optimal hyperparameters with the least MMCE value are `ntree = 327`, `mtry = 13`, `nodesize = 3` and `maxnodes = 50`.

```
# Set the optimal hyperparameters for the final model
tuned_rf <- setHyperPars(rf_learner, par.vals = tuned_rf_Pars$x)

# Train the final model using the optimal hyperparameters
tuned_rf_Model <- train(tuned_rf, AttritionTask)

# Check if there are enough trees in the Random Forest model

# First extract model information
rfModelData <- getLearnerModel(tuned_rf_Model)

# Plot MMCE vs number of trees
plot(rfModelData)
```



The mean out-of-bag error stabilizes too early, implying that there are enough trees in the Forest. The Positive class has a very high mean out-of-bag error rate which doesn't look good.

SVM model

```
# Define learner
svmLearner <- makeLearner("classif.svm", predict.type = "prob")

# Define hyperparameter space for tuning the model
kernels <- c("polynomial", "radial")
svmParamSpace <- makeParamSet(makeDiscreteParam("kernel", values = kernels),
                              makeIntegerParam("degree", lower = 1, upper =
5),
                              makeNumericParam("cost", lower = 0.1, upper =
12),
                              makeNumericParam("gamma", lower = 0.05, 5))

# Define search strategy to use random search with 200 iterations
# (SVM is computationally expensive)
randSearch <- makeTuneControlRandom(maxit = 200)

# Define CV strategy
cvForTuning <- makeResampleDesc("CV", iters = 6, stratify = TRUE)
```

```
# set random seed for reproducibility
set.seed(123)

# Begin parallelization
parallelStartSocket(cpus = detectCores())

## Starting parallelization in mode=socket with cpus=8.

# Perform hyperparameter tuning with cross-validation
tunedSvmPars <- tuneParams(learner =  svmLearner, task = AttritionTask,
                           resampling = cvForTuning,
                           par.set = svmParamSpace,
                           control = randSearch,
                           measures = list(mmce, acc, fpr, fnr),
                           show.info = FALSE)

## Exporting objects to slaves for mode socket: .mlr.slave.options

## Mapping in parallel: mode = socket; level = mlr.tuneParams; cpus = 8;
elements = 200.

# Stop parallelization
parallelStop()

## Stopped parallelization. All cleaned up.

# View tuning results
tunedSvmPars

## Tune result:
## Op. pars: kernel=polynomial; degree=1; cost=1.7; gamma=1.13
##
mmce.test.mean=0.1410413,acc.test.mean=0.8589587,fpr.test.mean=0.8104839,fnr.
test.mean=0.0121582
```

The SVM model has a training accuracy of 86.83%, and also has a very high FPR (FPR = 74.76%). The optimal hyperparameters are a polynomial kernel with a degree of 1, cost value of 1.7 and a gamma value of 1.13.

```
# Use the optimal hyperparameters to train the final model

# Set the optimal hyperparameters for the final model
tunedSvm <- setHyperPars(learner =  svmLearner, par.vals = tunedSvmPars$x)

# Train the final model
tunedSvmModel <- train(tunedSvm, AttritionTask)
```

XGBoost

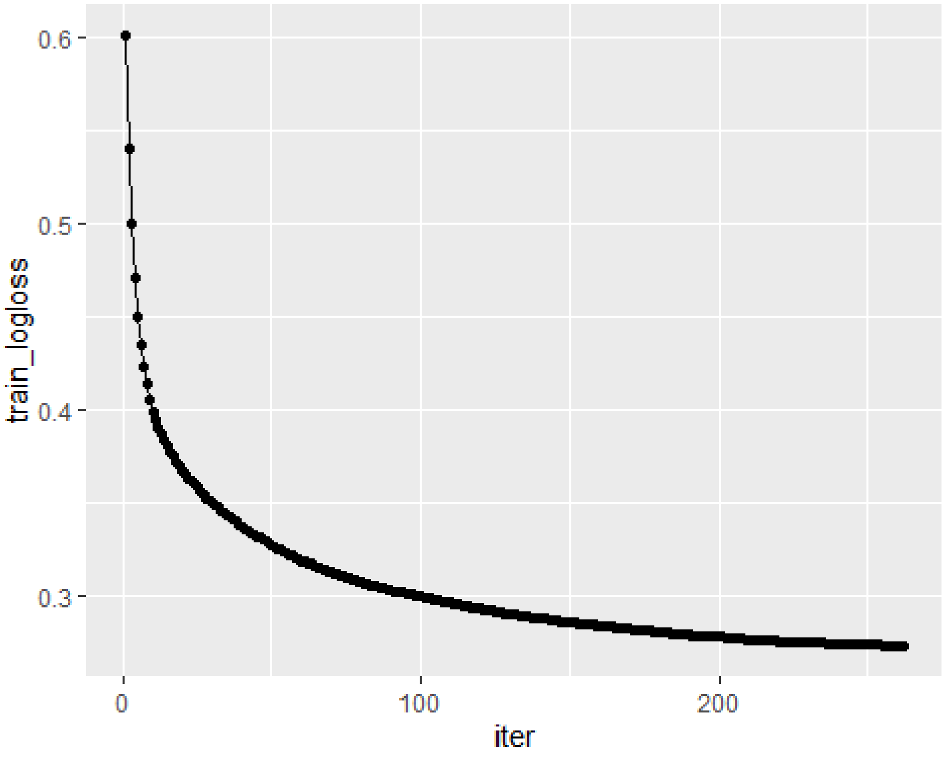
```
# Define learner
XGB <- makeLearner("classif.xgboost", predict.type = "prob")
```

```r
# Define hyperparameter space for tuning the model
xgbParamSpace <- makeParamSet(
makeNumericParam("eta", lower = 0.01, upper = 0.8),
makeNumericParam("gamma", lower = 0.001, upper = 7),
makeIntegerParam("max_depth", lower = 1, upper = 10),
makeNumericParam("min_child_weight", lower = 1, upper = 10),
makeNumericParam("subsample", lower = 0.5, upper = 1),
makeNumericParam("colsample_bytree", lower = 0.5, upper = 1),
makeIntegerParam("nrounds", lower = 20, upper = 300))

# Define search strategy to use random search
randSearch <- makeTuneControlRandom(maxit = 500)

# Make resampling description for CV
cvForTuning <- makeResampleDesc("CV", iters = 6, stratify = TRUE)

# Set random seed for reproducibility
set.seed(123)

# Tune the model with cross-validation
tunedXgbPars <- tuneParams(learner = XGB, task = AttritionTask,
                           resampling = cvForTuning,
                           par.set = xgbParamSpace,
                           control = randSearch,
                           measures = list(mmce, acc, fpr, fnr),
                           show.info = FALSE)

# Check performance
tunedXgbPars$y

## mmce.test.mean  acc.test.mean  fpr.test.mean  fnr.test.mean
##      0.1197737      0.8802263      0.6157594      0.0242917
```

XGBoost classifier has a training accuracy (88.02%) higher than the Logistic Regression, RF and SVM classifiers. However, it also has a high FPR (61.58%), though a bit lower than those by the previous models.

```r
# Train the final model using optimal hyperparameters

# Set the optimal hyperparameters for the final model
tunedXgb <- setHyperPars(XGB, par.vals = tunedXgbPars$x)

# Train the final model
tunedXgbModel <- train(tunedXgb, AttritionTask)

# Check if there are enough trees for the model

# Extract model information
xgbModelData <- getLearnerModel(tunedXgbModel)
```

```r
# Plot
ggplot(xgbModelData$evaluation_log, aes(iter, train_logloss)) +
  geom_line() + geom_point()
```



The training log loss stabilizes after about 200th iteration. This implies that I used enough trees.

Model Validation

I'll use the best two performing models (SVM and XGBoost) to make predictions on test data, and evaluate how they perform on unseen data. I'll use the caret's `confusionMatrix()` function to create a confusion matrix table, from which various evaluation metrics (like Accuracy, Sensitivity, Precision, Specificity) are calculated.

```r
# Use the SVM model to make predictions on test data
SvmPreds <- predict(tunedSvmModel, newdata = test_set)

## Warning in predict.WrappedModel(tunedSvmModel, newdata = test_set):
Provided
## data for prediction is not a pure data.frame but from class tbl_df, hence
it
## will be converted.

# Collect prediction
SvmPreds_data <- SvmPreds$data
```

```r
# Generate a confusion matrix
confusionMatrix(table(SvmPreds_data$response, SvmPreds_data$truth), positive
= "Yes")
```

```
## Confusion Matrix and Statistics
##
##
##        No Yes
##   No  243  32
##   Yes   3  15
##
##                Accuracy : 0.8805
##                  95% CI : (0.8378, 0.9154)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 0.03008
##
##                   Kappa : 0.409
##
##  Mcnemar's Test P-Value : 2.214e-06
##
##             Sensitivity : 0.31915
##             Specificity : 0.98780
##          Pos Pred Value : 0.83333
##          Neg Pred Value : 0.88364
##              Prevalence : 0.16041
##          Detection Rate : 0.05119
##    Detection Prevalence : 0.06143
##       Balanced Accuracy : 0.65348
##
##        'Positive' Class : Yes
##
```

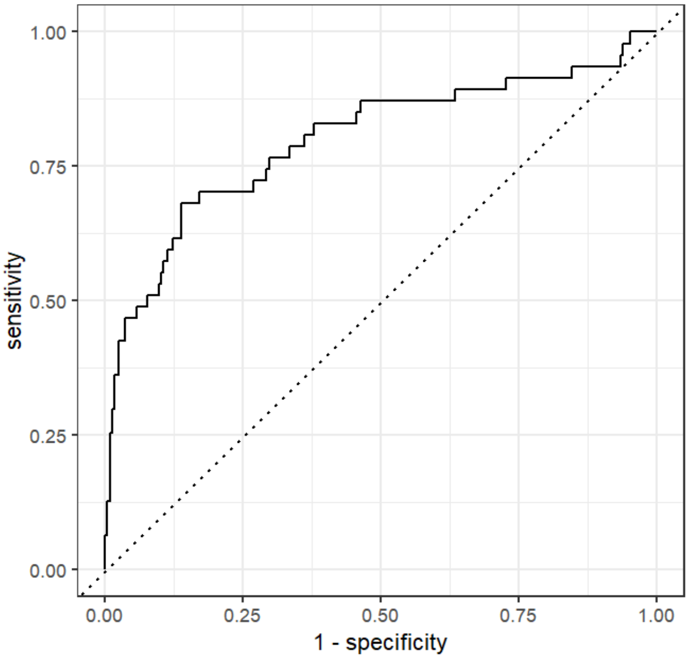
SVM classifier has a validation accuracy of 88.05%, which is good. However, the SVM model has a poor sensitivity for the positive class (31.92%), but the model's precision is good. When this model predicts a positive case, it is correct 83.33% of the time, and when it predicts a negative case, it is correct 88.36% of the time.

```r
# Calculate ROC AUC
SvmPreds_data |> roc_auc(truth = truth, prob.Yes)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.799
```

SVM has a ROC_AUC value 0.79, which isn't bad.

```r
# Plot ROC curve
SvmPreds_data |> roc_curve(truth = truth, prob.Yes) |> autoplot()
```



Note: Unfortunately, the `vip()` and `varImp()` functions don't have an implementation of feature importance for SVM model.

```
# Use the XGBoost model to make predictions on test data
xgbPreds <- predict(tunedXgbModel, newdata = test_set)

## Warning in predict.WrappedModel(tunedXgbModel, newdata = test_set):
Provided
## data for prediction is not a pure data.frame but from class tbl_df, hence
it
## will be converted.

# Collect prediction
xgbPreds_data <- xgbPreds$data

# Calculate confusion matrix
confusionMatrix(table(xgbPreds_data$response, xgbPreds_data$truth), positive
= "Yes")

## Confusion Matrix and Statistics
##
##
##          No Yes
##    No   241  25
##    Yes    5  22
##
##                   Accuracy : 0.8976
##                     95% CI : (0.8571, 0.9298)
##        No Information Rate : 0.8396
##        P-Value [Acc > NIR] : 0.0029273
##
```

```
##                   Kappa : 0.5408
##
##  Mcnemar's Test P-Value : 0.0005226
##
##             Sensitivity : 0.46809
##             Specificity : 0.97967
##          Pos Pred Value : 0.81481
##          Neg Pred Value : 0.90602
##              Prevalence : 0.16041
##          Detection Rate : 0.07509
##    Detection Prevalence : 0.09215
##       Balanced Accuracy : 0.72388
##
##        'Positive' Class : Yes
##
```

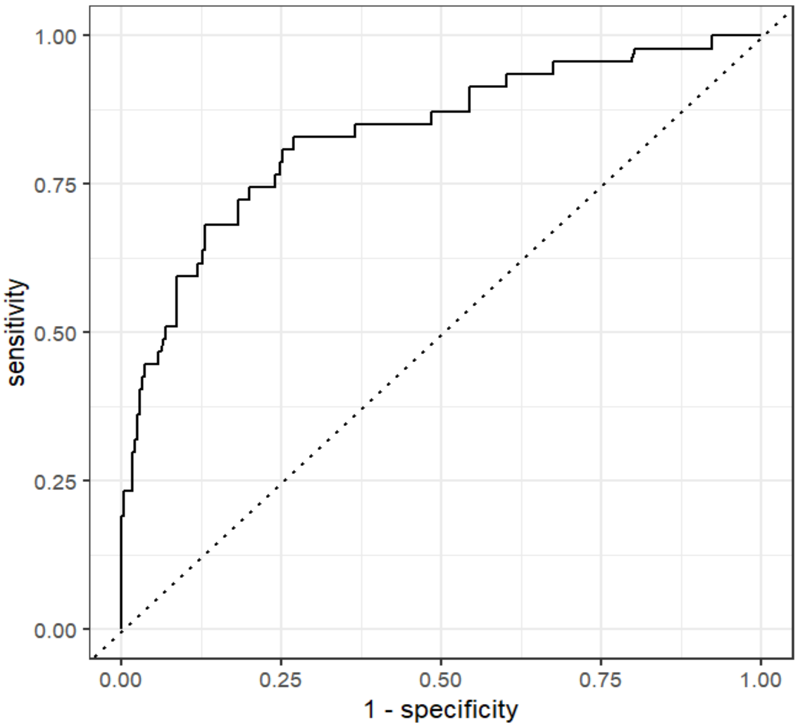
XGBoost classifier has a validation accuracy of 89.76%. It has a better Sensitivity than SVM (46.81%), and a Precision of 81.48%. When this model predicts a positive case, it is correct 81.48% of the time, and when it predicts a negative case, it is correct 90.6% of the time.

```
# Calculate ROC AUC
xgbPreds_data |> roc_auc(truth = truth, prob.Yes)
```

```
## # A tibble: 1 × 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 roc_auc binary         0.834
```

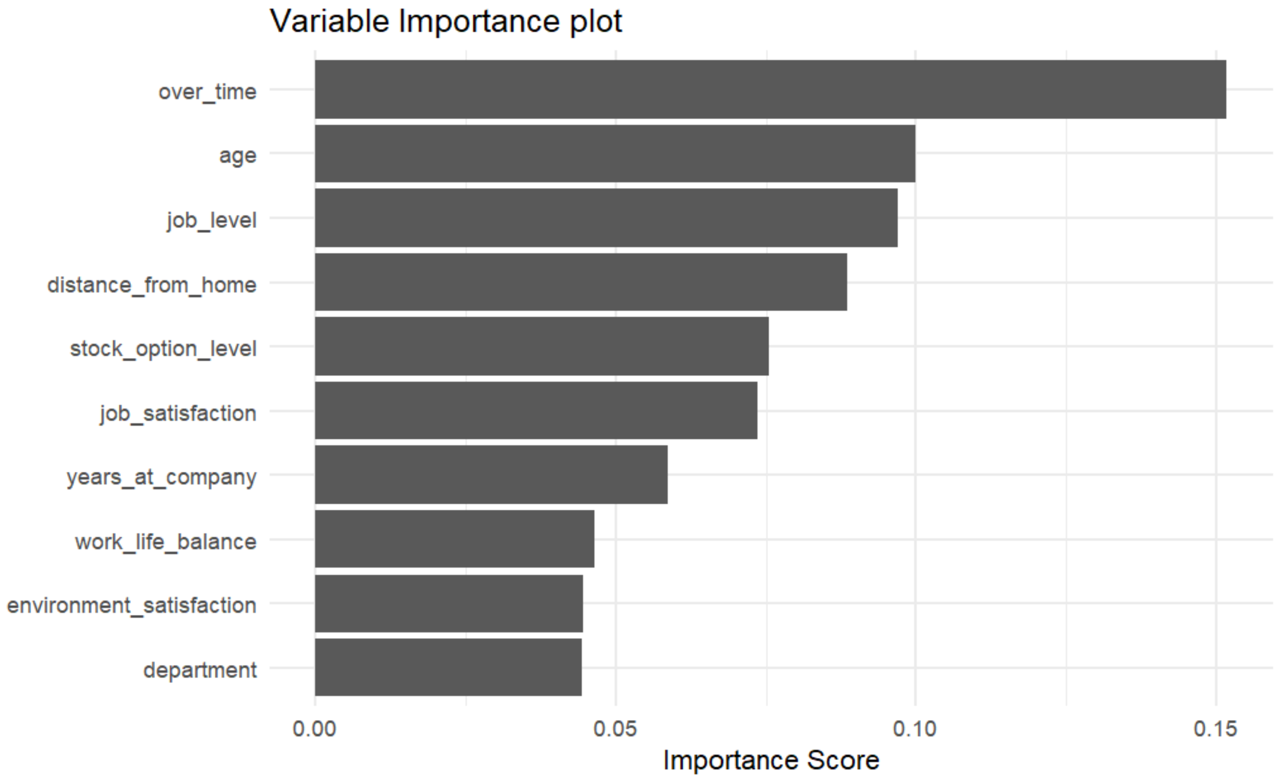
XGBoost has the highest ROC AUC value (0.83).

```
# Plot ROC curve
xgbPreds_data |> roc_curve(truth = truth, prob.Yes) |> autoplot()
```



ROC curve for the XGBoost classifier looks good. XGBoost is better at distinguishing between the two classes than SVM.

```r
# Create the variable importance plot
vip(tunedXgbModel) +
  labs(title = "Variable Importance plot",
       y = "Importance Score") +
  theme_minimal()
```



According to the XGBoost model, the most important predictors of attrition are overtime, age, job level, distance from home, stock option level, job satisfaction, years at the current company, work-life balance, environment satisfaction and department.

Conclusion

I'll pick the XGBoost model because it has a higher accuracy (89.76%), and a better precision for the positive class (81.48%).

Limitations of this Analysis

- The issue of class imbalance was not handled, leading to lower sensitivity by the model.
- Only the random search strategy was used during hyperparameter tuning, which might not have given the optimal hyperparameter combinations. The model can therefore still be improved upon.

Recommendations

- The high class imbalance should be handled using SMOTE technique, or using cost-sensitive classifiers.
- SHAP analysis should be conducted to help in understanding how individual features contribute to the predicted probabilities of attrition.

- What-if analysis should also be done using the `counterfactuals` package to determine the changes in individual features that can lead to a reduction in the probability (likelihood) of attrition. This can help in formulating potential retention strategies.