Employee Turnover Analysis

2024-12-10

The exploratory data analysis and machine learning model implementation done on this dataset is to inform employers regarding employee turnover factors.

This exploration is a useful tool for hiring organizations, staffing agencies, and management due to the linear and non-linear exploration of features that might influence many to leave their employers.

The csv file used for this research was pulled from kaggle.com, as it was deemed to be the most reliable and abundant dataset present at the time of this exploration.

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(ggplot2)
library(pheatmap)
library(reshape2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(e1071)
library(car)
```

```
## Loading required package: carData
```

```
library(randomForest)
```

```
## randomForest 4.7-1.2
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(caret)

# Upload csv file into RStudio and view first few rows
setwd("/Users/betuldemir/DTSC3010/")
turnoverData <- read.csv('~/DTSC3010/HR_comma_sep.csv')
head(turnoverData)
```

```
##   satisfaction_level last_evaluation number_project average_montly_hours
## 1               0.38            0.53              2                  157
## 2               0.80            0.86              5                  262
## 3               0.11            0.88              7                  272
## 4               0.72            0.87              5                  223
## 5               0.37            0.52              2                  159
## 6               0.41            0.50              2                  153
##   time_spend_company Work_accident left promotion_last_5years Department salary
## 1                  3             0    1                     0      sales    low
## 2                  6             0    1                     0      sales medium
## 3                  4             0    1                     0      sales medium
## 4                  5             0    1                     0      sales    low
## 5                  3             0    1                     0      sales    low
## 6                  3             0    1                     0      sales    low
```

```
# Convert categorical columns to factors
turnoverData$salary <- as.factor(turnoverData$salary)
turnoverData$Department <- as.factor(turnoverData$Department)

# Confirm the data structure again
str(turnoverData)
```

```
## 'data.frame':    14999 obs. of  10 variables:
##  $ satisfaction_level   : num  0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
##  $ last_evaluation      : num  0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...
##  $ number_project       : int  2 5 7 5 2 2 6 5 5 2 ...
##  $ average_montly_hours : int  157 262 272 223 159 153 247 259 224 142 ...
##  $ time_spend_company   : int  3 6 4 5 3 3 4 5 5 3 ...
##  $ Work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Department           : Factor w/ 10 levels "accounting","hr",..: 8 8 8 8 8 8 8
8 8 8 ...
##  $ salary               : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2
2 2 2 ...
```

Statistical Summary of Dataset

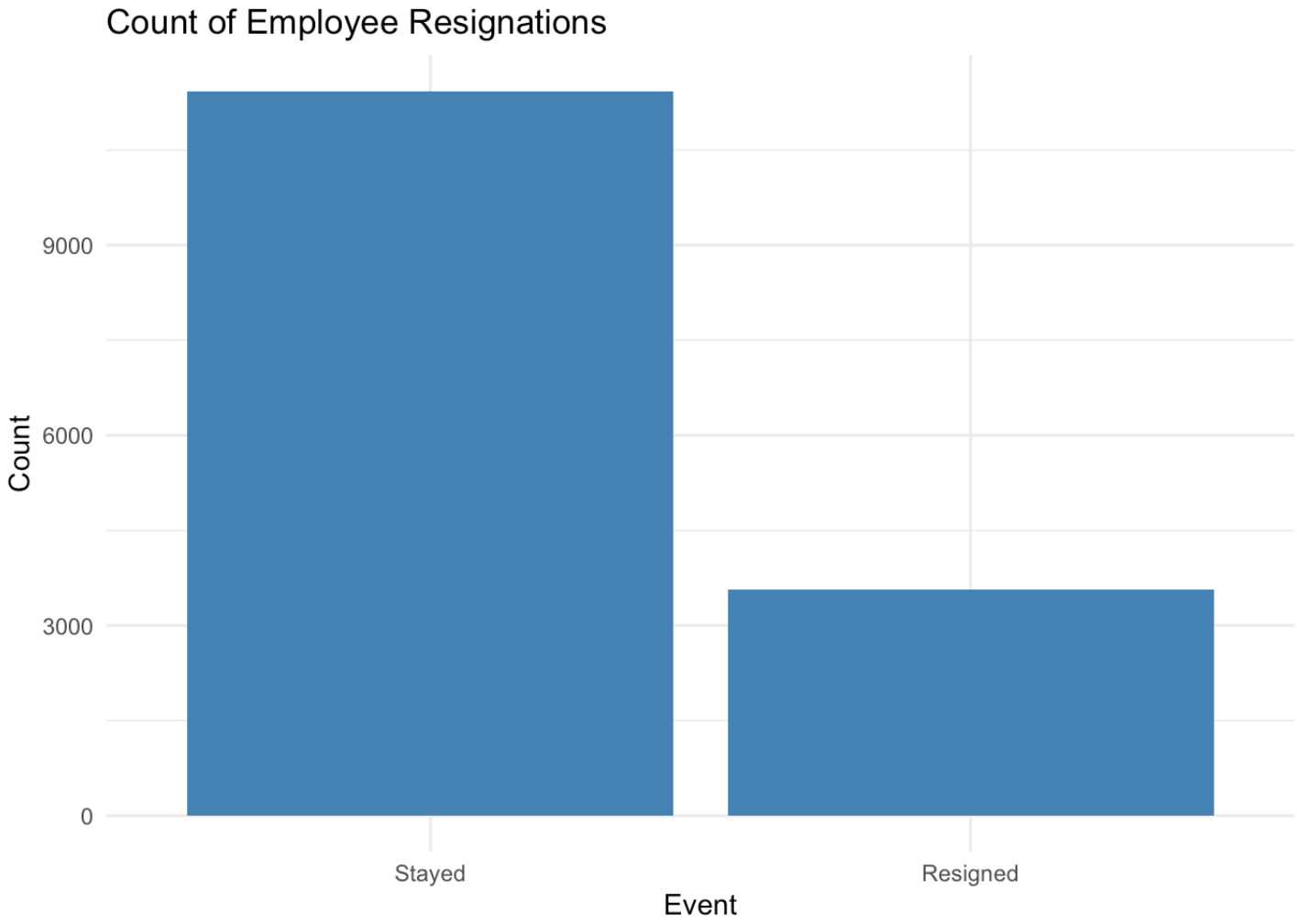
```
# Obtain a statistical summary of the dataset
summary(turnoverData)
```

```
##  satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900     Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400     Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128     Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200     3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000     Max.   :1.0000   Max.   :7.000   Max.   :310.0
##
##  time_spend_company Work_accident         left        promotion_last_5years
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median : 3.000     Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.2381   Mean   :0.02127
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##
##        Department      salary
##  sales      :4140   high  :1237
##  technical  :2720   low   :7316
##  support    :2229   medium:6446
##  IT         :1227
##  product_mng: 902
##  marketing  : 858
##  (Other)    :2923
```

Visualization Plots

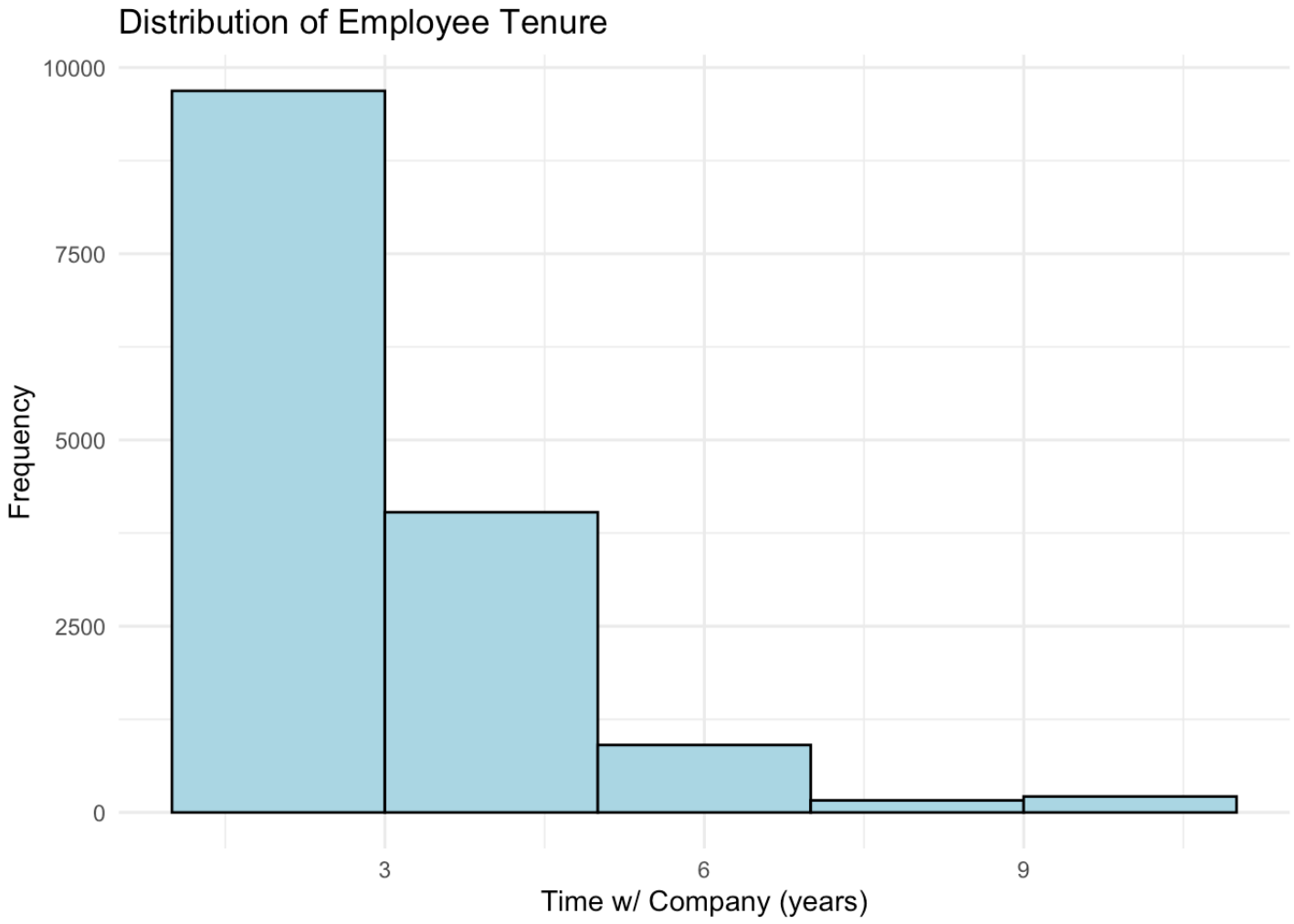
Bar Chart: Count of Employee Resignition

```
# Create a bar chart to visualize the count of employee resignations
ggplot(turnoverData, aes(x = factor(left, labels = c("Stayed", "Resigned")))) +
  geom_bar(fill = "steelblue") +
  labs(
    title = "Count of Employee Resignations",
    x = "Event",
    y = "Count"
  ) +
  theme_minimal()
```



Histogram: Employee's Duration w/ Company

```
# Create a histogram to visualize employees' duration at the company
ggplot(turnoverData, aes(x = time_spend_company)) +
  geom_histogram(binwidth = 2, fill = "lightblue", color = "black") +
  labs(title = "Distribution of Employee Tenure", x = "Time w/ Company (years)", y =
"Frequency") +
  theme_minimal()
```

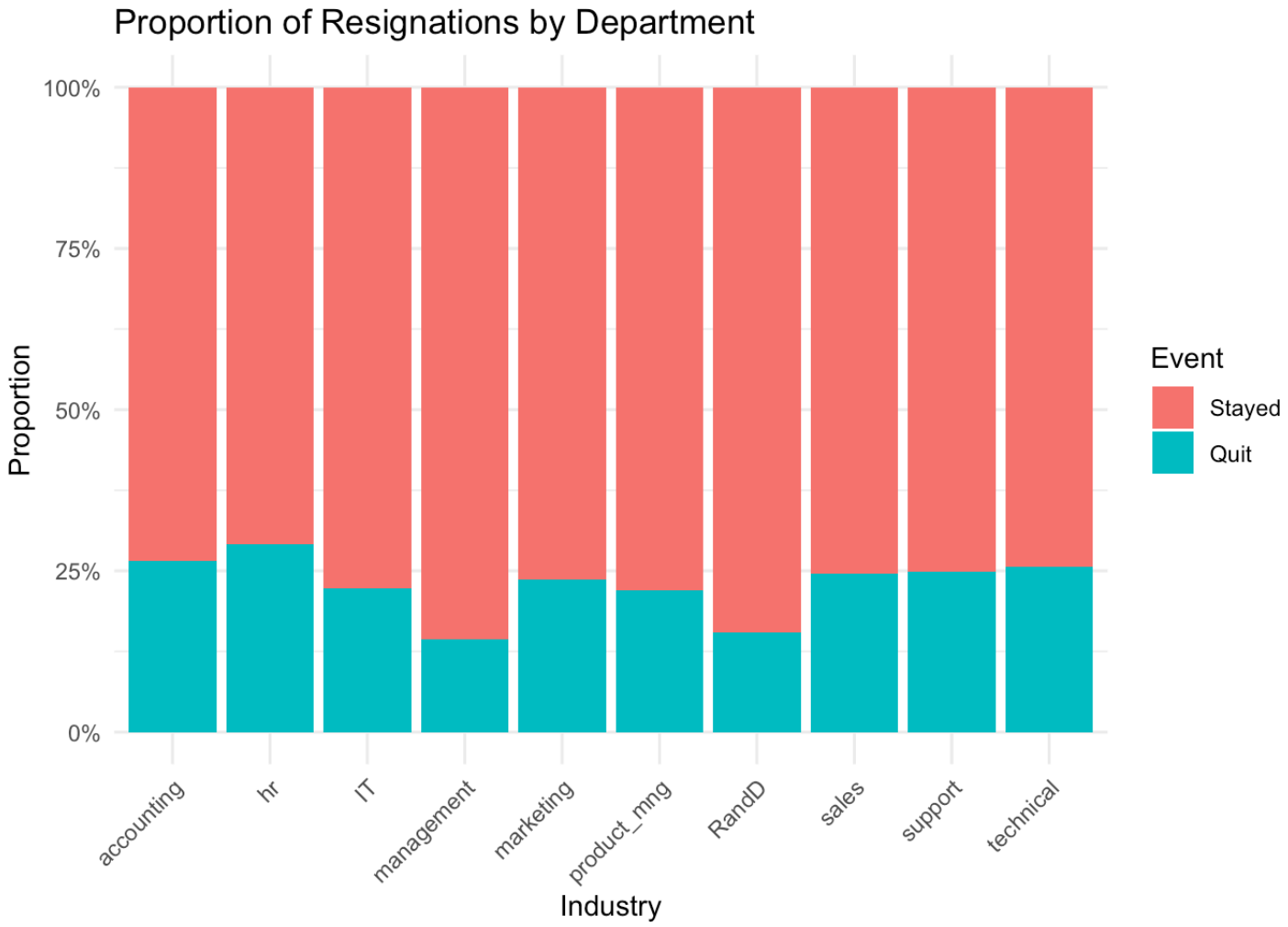


Scatter Plot: Proportion of Resignations by Promotion

```
# Create a bar chart to display the proportion of employees that stayed/quit based on work accidents
ggplot(turnoverData, aes(x = factor(promotion_last_5years, labels = c("No Promotion", "Promotion")), fill = factor(left, labels = c("Stayed", "Quit")))) +
  geom_bar(position = "fill") +
  labs(
    title = "Proportion of Resignations by Promotion",
    x = "Promotions within the Last 5 Years",
    y = "Proportion",
    fill = "Event"
  ) +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

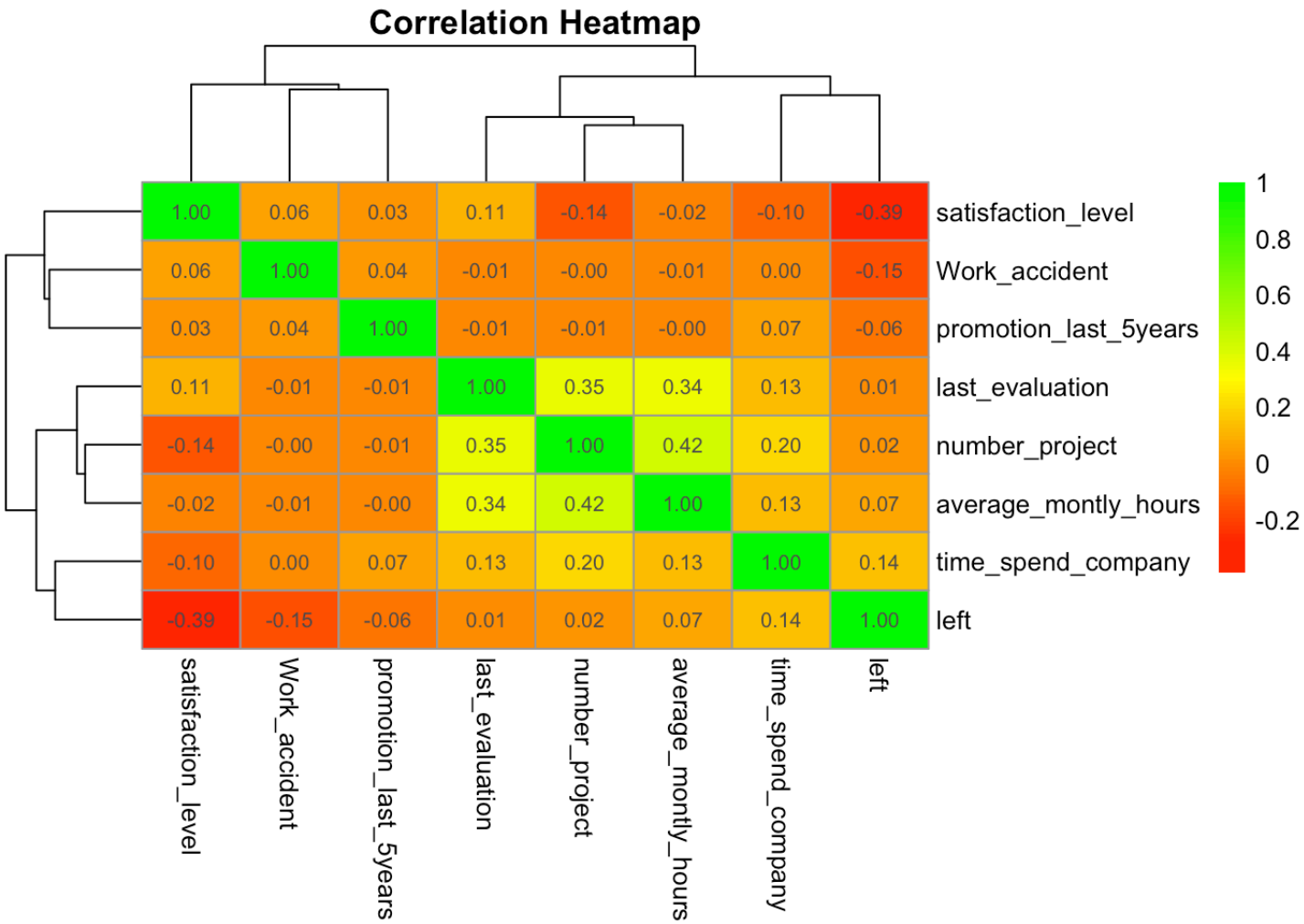
Stacked Bar Plot: Proportion of Resignations by Department

```
# Create a bar chart to display the proportion of employees that stayed/ quit
ggplot(turnoverData, aes(x = Department, fill = factor(left, labels = c("Stayed", "Qu
it")))) +
  geom_bar(position = "fill") +
  labs(
    title = "Proportion of Resignations by Department",
    x = "Industry",
    y = "Proportion",
    fill = "Event"
  ) +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Heatmap: Correlation Matrix of Numeric Values

```
# Create a subset to store columns with numeric cells
numeric_data <- turnoverData[, sapply(turnoverData, is.numeric)]

# Calculate the correlation matrix for the subset
cor_matrix <- cor(numeric_data, use = "complete.obs")

# Create a heatmap to visualize the correlation matrix
pheatmap(cor_matrix,
         clustering_distance_rows = "euclidean",
         clustering_distance_cols = "euclidean",
         clustering_method = "complete",
         display_numbers = TRUE,
         color = colorRampPalette(c("red", "yellow", "green"))(100),
         main = "Correlation Heatmap")
```



Statistical Tests

Pearson's T-Test

```
# Select numeric features (excluding target feature: left)
numeric_features <- names(turnoverData)[sapply(turnoverData, is.numeric) & names(turn
overData) != "left"]

# Perform t-tests for each numeric feature
t_test_results <- lapply(numeric_features, function(feature) {
  group_0 <- turnoverData[[feature]][turnoverData$left == 0]
  group_1 <- turnoverData[[feature]][turnoverData$left == 1]

  test_result <- t.test(group_0, group_1)

# Extract relevant information
  data.frame(
    Feature = feature,
    P_Value = test_result$p.value,
    Mean_Group_0 = mean(group_0, na.rm = TRUE),
    Mean_Group_1 = mean(group_1, na.rm = TRUE),
    Difference = mean(group_0, na.rm = TRUE) - mean(group_1, na.rm = TRUE)
  )
})

# Combine results into a single data frame
t_test_results_df <- do.call(rbind, t_test_results)

# Sort by p-value (most significant first)
t_test_results_df <- t_test_results_df[order(t_test_results_df$P_Value), ]

print(t_test_results_df)
```

```
##                 Feature       P_Value Mean_Group_0 Mean_Group_1   Difference
## 1    satisfaction_level  0.000000e+00   0.66680959 4.400980e-01  0.226711579
## 6         Work_accident 2.402805e-138   0.17500875 4.732568e-02  0.127683071
## 5    time_spend_company 1.595078e-110   3.38003150 3.876505e+00 -0.496473679
## 7 promotion_last_5years  2.524306e-27   0.02625131 5.320638e-03  0.020930674
## 4  average_montly_hours  5.907055e-14 199.06020301 2.074192e+02 -8.359007295
## 3        number_project  3.034068e-02   3.78666433 3.855503e+00 -0.068838327
## 2       last_evaluation  4.682750e-01   0.71547340 7.181126e-01 -0.002639175
```

ANOVA

```
# Convert categorical variables to factors
turnoverData$Department <- as.factor(turnoverData$Department)
turnoverData$salary <- as.factor(turnoverData$salary)

# Perform Two-Way ANOVA: Test the effect of both 'Department' and 'salary' on 'satisf
action_level'
anova_department_salary_sl <- aov(satisfaction_level ~ Department * salary, data = tu
rnoverData)

# Perform Two-Way ANOVA: Test the effect of both 'Department' and 'salary' on 'left'
anova_department_salary_left <- aov(left ~ Department * salary, data = turnoverData)

# View summary of the Two-Way ANOVA results
summary(anova_department_salary_sl)
```

```
##                       Df Sum Sq Mean Sq F value Pr(>F)
## Department             9    1.2  0.1333   2.166 0.0214 *
## salary                 2    2.4  1.1782  19.138  5e-09 ***
## Department:salary     18    2.0  0.1139   1.849 0.0155 *
## Residuals          14969  921.5  0.0616
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(anova_department_salary_left)
```

```
##                       Df Sum Sq Mean Sq F value   Pr(>F)
## Department             9   15.8    1.75  10.009 1.76e-15 ***
## salary                 2   64.3   32.16 183.959  < 2e-16 ***
## Department:salary     18   23.6    1.31   7.486  < 2e-16 ***
## Residuals          14969 2617.2    0.17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Machine Learning Models

Logistic Regression

```
# Fit the logistic regression model
log_model <- glm(left ~ satisfaction_level + salary + Work_accident + promotion_last_
5years,
                 data = turnoverData, family = "binomial")

# Summary of the model to view coefficients and p-values
summary(log_model)
```

```
##
## Call:
## glm(formula = left ~ satisfaction_level + salary + Work_accident +
##     promotion_last_5years, family = "binomial", data = turnoverData)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -0.36961    0.12624  -2.928  0.00341 **
## satisfaction_level    -3.83387    0.08986 -42.665  < 2e-16 ***
## salarylow              1.80925    0.12253  14.765  < 2e-16 ***
## salarymedium           1.31339    0.12376  10.612  < 2e-16 ***
## Work_accident         -1.46895    0.08732 -16.823  < 2e-16 ***
## promotion_last_5years -1.22178    0.24964  -4.894 9.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 16465  on 14998  degrees of freedom
## Residual deviance: 13398  on 14993  degrees of freedom
## AIC: 13410
##
## Number of Fisher Scoring iterations: 5
```

```
# Extract coefficients and p-values
coefficients <- summary(log_model)$coefficients

# Create a data frame of results
importance <- data.frame(
  Feature = rownames(coefficients)[-1], # Exclude intercept
  Coefficient = coefficients[-1, 1],   # Coefficients
  P_Value = coefficients[-1, 4]        # P-values
)

# Sort by absolute coefficient (most influential to least)
importance <- importance[order(abs(importance$Coefficient), decreasing = TRUE), ]

# Print the sorted importance table
print(importance)
```

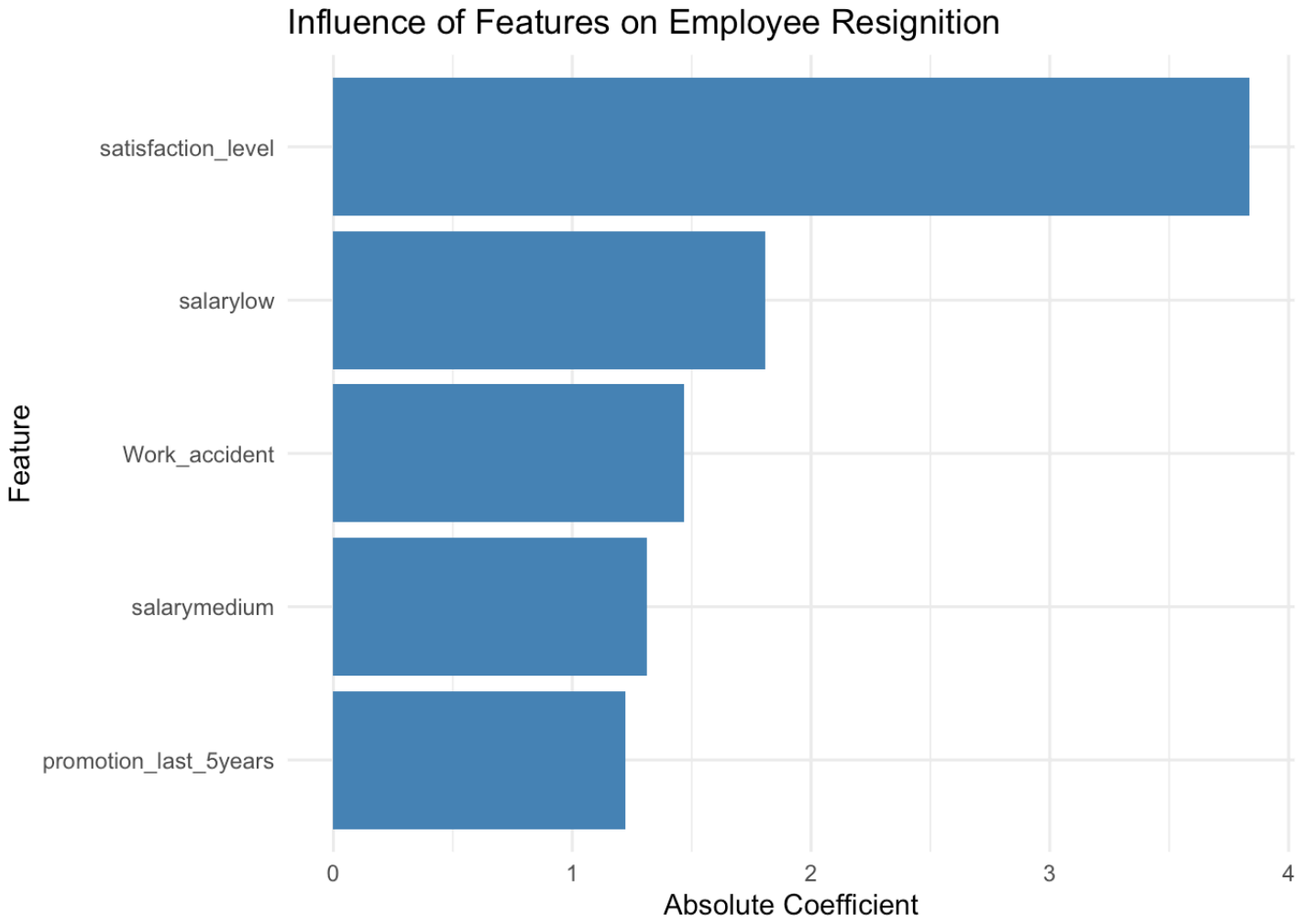
```
##                                   Feature Coefficient      P_Value
## satisfaction_level       satisfaction_level   -3.833865 0.000000e+00
## salarylow                         salarylow    1.809253 2.447602e-49
## Work_accident                 Work_accident   -1.468948 1.656431e-63
## salarymedium                   salarymedium    1.313389 2.617801e-26
## promotion_last_5years promotion_last_5years   -1.221776 9.871116e-07
```

```
# Make predictions on the dataset
turnoverData$predicted_prob <- predict(log_model, type = "response") # Predicted prob
abilities
turnoverData$predicted_class <- ifelse(turnoverData$predicted_prob > 0.5, 1, 0) # Pre
dicted class (threshold = 0.5)

# Create a confusion matrix
library(caret) # For confusionMatrix function
confusion_matrix <- confusionMatrix(
  as.factor(turnoverData$predicted_class),
  as.factor(turnoverData$left),
  positive = "1" # Define "1" as the positive class (employees who quit)
)

# Print the confusion matrix
print(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 10678  2497
##          1   750  1074
##
##                Accuracy : 0.7835
##                  95% CI : (0.7768, 0.7901)
##     No Information Rate : 0.7619
##     P-Value [Acc > NIR] : 1.832e-10
##
##                   Kappa : 0.2827
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3008
##             Specificity : 0.9344
##          Pos Pred Value : 0.5888
##          Neg Pred Value : 0.8105
##              Prevalence : 0.2381
##          Detection Rate : 0.0716
##    Detection Prevalence : 0.1216
##       Balanced Accuracy : 0.6176
##
##        'Positive' Class : 1
##
```

Logistic Regression Visualization

```
# Create a bar chart of the most influential features on 'left' by coefficients
ggplot(importance, aes(x = reorder(Feature, abs(Coefficient)), y = abs(Coefficient)))
+
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Influence of Features on Employee Resignition", x = "Feature", y = "A
bsolute Coefficient") +
  theme_minimal()
```



Random Forest

```
# Convert the 'left' column to a factor and store it as 'response'
turnoverData$response <- as.factor(turnoverData$left)

# Split into training and test sets
set.seed(123)
trainIndex <- createDataPartition(turnoverData$response, p = 0.8, list = FALSE)
trainData <- turnoverData[trainIndex, ]
testData <- turnoverData[-trainIndex, ]


# Train Random Forest
rf_model <- randomForest(response ~ satisfaction_level+salary+Work_accident+promotion
_last_5years, data = trainData, ntree = 100, importance = TRUE)

# Predict on test set
rf_preds <- predict(rf_model, testData)

# Evaluate
confusionMatrix(rf_preds, testData$response)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2229  262
##          1   56  452
##
##                Accuracy : 0.894
##                  95% CI : (0.8824, 0.9048)
##     No Information Rate : 0.7619
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6755
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9755
##             Specificity : 0.6331
##          Pos Pred Value : 0.8948
##          Neg Pred Value : 0.8898
##              Prevalence : 0.7619
##          Detection Rate : 0.7432
##    Detection Prevalence : 0.8306
##       Balanced Accuracy : 0.8043
##
##        'Positive' Class : 0
##
```

Support Vector Machines (SVM)

```
# Convert the 'left' column to a factor and store it as 'response'
turnoverData$response <- as.factor(turnoverData$left)
# Identify numeric columns
numeric_columns <- sapply(turnoverData, is.numeric)

# Scale numeric columns
train_scaled <- scale(turnoverData[, numeric_columns])
test_scaled <- scale(turnoverData[, numeric_columns])

# Convert scaled data back to a data frame
train_scaled <- data.frame(train_scaled)
test_scaled <- data.frame(test_scaled)

# Add back non-numeric columns (response, treatment, Department, salary)
train_scaled$response <- turnoverData$response
test_scaled$response <- turnoverData$response

# Train SVM
svm_model <- svm(response ~ satisfaction_level+salary+Work_accident+promotion_last_5y
ears, data = trainData, kernel = "radial")

# Predict on test set
svm_preds <- predict(svm_model, testData)

# Evaluate
confusionMatrix(svm_preds, testData$response)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2160  539
##          1  125  175
##
##                Accuracy : 0.7786
##                  95% CI : (0.7633, 0.7933)
##     No Information Rate : 0.7619
##     P-Value [Acc > NIR] : 0.01634
##
##                   Kappa : 0.2378
##
##  Mcnemar's Test P-Value : < 2e-16
##
##             Sensitivity : 0.9453
##             Specificity : 0.2451
##          Pos Pred Value : 0.8003
##          Neg Pred Value : 0.5833
##              Prevalence : 0.7619
##          Detection Rate : 0.7202
##    Detection Prevalence : 0.9000
##       Balanced Accuracy : 0.5952
##
##        'Positive' Class : 0
##
```