POM_681 Final Project

Nandhini Vijayakumar,Suvidha Sharma, Akanksha Sahitya Bhupathiraju, Jyothi Pavan Kutala

2025-05-01

```r
# Load libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(e1071)
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:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(xgboost)
```

```
##
## Attaching package: 'xgboost'
##
```

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

```r
library(ROCR)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```r
library(smotefamily)
library(rpart)
library(rpart.plot)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```r
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(ggplot2)
library(patchwork)
library(ggcorrplot)
# Load dataset
hr <- read.csv("/Users/nandhinivijayakumar/Downloads/WA_Fn-UseC_-HR-Employee-Attrition.csv")
head(hr)
```

```
##   Age Attrition    BusinessTravel DailyRate             Department
## 1  41       Yes     Travel_Rarely      1102                  Sales
## 2  49        No Travel_Frequently       279 Research & Development
## 3  37       Yes     Travel_Rarely      1373 Research & Development
## 4  33        No Travel_Frequently      1392 Research & Development
## 5  27        No     Travel_Rarely       591 Research & Development
## 6  32        No Travel_Frequently      1005 Research & Development
##   DistanceFromHome Education EducationField EmployeeCount EmployeeNumber
## 1                1         2  Life Sciences             1              1
## 2                8         1  Life Sciences             1              2
## 3                2         2          Other             1              4
## 4                3         4  Life Sciences             1              5
## 5                2         1        Medical             1              7
## 6                2         2  Life Sciences             1              8
##   EnvironmentSatisfaction Gender HourlyRate JobInvolvement JobLevel
## 1                       2 Female         94              3        2
## 2                       3   Male         61              2        2
## 3                       4   Male         92              2        1
## 4                       4 Female         56              3        1
## 5                       1   Male         40              3        1
## 6                       4   Male         79              3        1
##                 JobRole JobSatisfaction MaritalStatus MonthlyIncome MonthlyRate
## 1       Sales Executive               4        Single          5993       19479
```

```
## 2    Research Scientist              2       Married          5130       24907
## 3 Laboratory Technician              3        Single          2090        2396
## 4    Research Scientist              3       Married          2909       23159
## 5 Laboratory Technician              2       Married          3468       16632
## 6 Laboratory Technician              4        Single          3068       11864
##   NumCompaniesWorked Over18 OverTime PercentSalaryHike PerformanceRating
## 1                  8      Y      Yes                11                 3
## 2                  1      Y       No                23                 4
## 3                  6      Y      Yes                15                 3
## 4                  1      Y      Yes                11                 3
## 5                  9      Y       No                12                 3
## 6                  0      Y       No                13                 3
##   RelationshipSatisfaction StandardHours StockOptionLevel TotalWorkingYears
## 1                        1            80                0                 8
## 2                        4            80                1                10
## 3                        2            80                0                 7
## 4                        3            80                0                 8
## 5                        4            80                1                 6
## 6                        3            80                0                 8
##   TrainingTimesLastYear WorkLifeBalance YearsAtCompany YearsInCurrentRole
## 1                     0               1              6                  4
## 2                     3               3             10                  7
## 3                     3               3              0                  0
## 4                     3               3              8                  7
## 5                     3               3              2                  2
## 6                     2               2              7                  7
##   YearsSinceLastPromotion YearsWithCurrManager
## 1                       0                    5
## 2                       1                    7
## 3                       0                    0
## 4                       3                    0
## 5                       2                    2
## 6                       3                    6
```

```r
#.................................DATA CLEANING.........................................

# Check for missing values
colSums(is.na(hr))
```

```
##                      Age                Attrition           BusinessTravel
##                        0                        0                        0
##                DailyRate               Department         DistanceFromHome
##                        0                        0                        0
##                Education           EducationField            EmployeeCount
##                        0                        0                        0
##           EmployeeNumber  EnvironmentSatisfaction                   Gender
##                        0                        0                        0
##               HourlyRate           JobInvolvement                 JobLevel
##                        0                        0                        0
##                  JobRole          JobSatisfaction            MaritalStatus
##                        0                        0                        0
##            MonthlyIncome              MonthlyRate       NumCompaniesWorked
##                        0                        0                        0
##                   Over18                 OverTime        PercentSalaryHike
##                        0                        0                        0
```

```
##        PerformanceRating RelationshipSatisfaction            StandardHours
##                        0                        0                        0
##         StockOptionLevel        TotalWorkingYears    TrainingTimesLastYear
##                        0                        0                        0
##          WorkLifeBalance           YearsAtCompany       YearsInCurrentRole
##                        0                        0                        0
##  YearsSinceLastPromotion     YearsWithCurrManager
##                        0                        0
```

```
# Drop unnecessary columns (like EmployeeNumber, Over18, StandardHours, EmployeeCount - not useful)
hr <- hr %>% select(-c(EmployeeNumber, Over18, StandardHours, EmployeeCount))

hr <- hr %>%dplyr::mutate_if(is.character, as.factor)

# Make Attrition a factor
hr$Attrition <- as.factor(hr$Attrition)

# Check structure
str(hr)
```

```
## 'data.frame':    1470 obs. of  31 variables:
##  $ Age                     : int  41 49 37 33 27 32 59 30 38 36 ...
##  $ Attrition               : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 2 3 2 3 2 3 3
##  $ DailyRate               : int  1102 279 1373 1392 591 1005 1324 1358 216 1299 ...
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...
##  $ DistanceFromHome        : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ Education               : int  2 1 2 4 1 2 3 1 3 3 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...
##  $ EnvironmentSatisfaction : int  2 3 4 4 1 4 3 4 4 3 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...
##  $ HourlyRate              : int  94 61 92 56 40 79 81 67 44 94 ...
##  $ JobInvolvement          : int  3 2 2 3 3 3 4 3 2 3 ...
##  $ JobLevel                : int  2 2 1 1 1 1 1 1 3 2 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1
##  $ JobSatisfaction         : int  4 2 3 3 2 4 1 3 3 3 ...
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...
##  $ MonthlyIncome           : int  5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...
##  $ MonthlyRate             : int  19479 24907 2396 23159 16632 11864 9964 13335 8787 16577 ...
##  $ NumCompaniesWorked      : int  8 1 6 1 9 0 4 1 0 6 ...
##  $ OverTime                : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 2 1 1 1 ...
##  $ PercentSalaryHike       : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ PerformanceRating       : int  3 4 3 3 3 3 4 4 4 3 ...
##  $ RelationshipSatisfaction: int  1 4 2 3 4 3 1 2 2 2 ...
##  $ StockOptionLevel        : int  0 1 0 0 1 0 3 1 0 2 ...
##  $ TotalWorkingYears       : int  8 10 7 8 6 8 12 1 10 17 ...
##  $ TrainingTimesLastYear   : int  0 3 3 3 3 2 3 2 2 3 ...
##  $ WorkLifeBalance         : int  1 3 3 3 3 2 2 3 3 2 ...
##  $ YearsAtCompany          : int  6 10 0 8 2 7 1 1 9 7 ...
##  $ YearsInCurrentRole      : int  4 7 0 7 2 7 0 0 7 7 ...
##  $ YearsSinceLastPromotion : int  0 1 0 3 2 3 0 0 1 7 ...
##  $ YearsWithCurrManager    : int  5 7 0 0 2 6 0 0 8 7 ...
```

```
#..................................EDA (Exploratory Data Analysis)......................
```

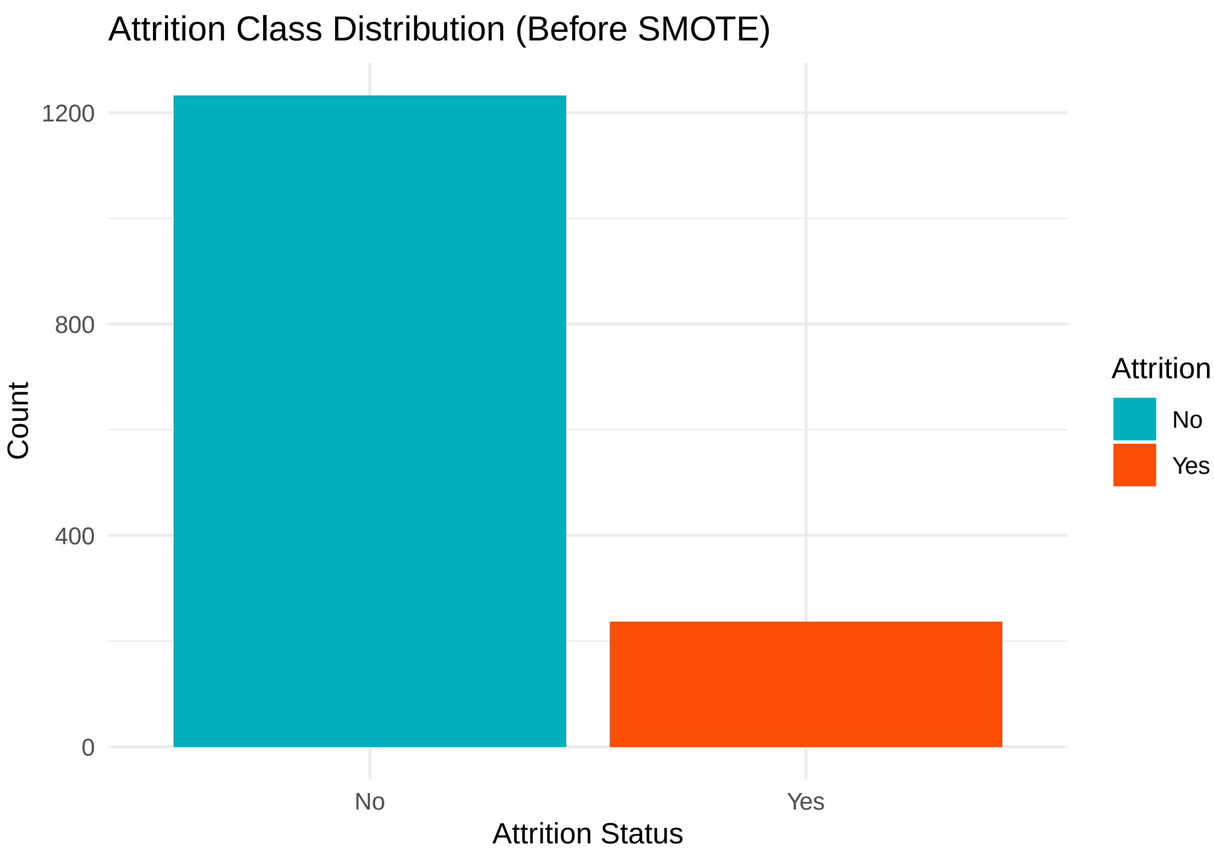
```r
# Attrition distribution(checking data imbalance)
table(hr$Attrition)
```

```
##
##   No  Yes
## 1233  237
```

```r
prop.table(table(hr$Attrition))
```

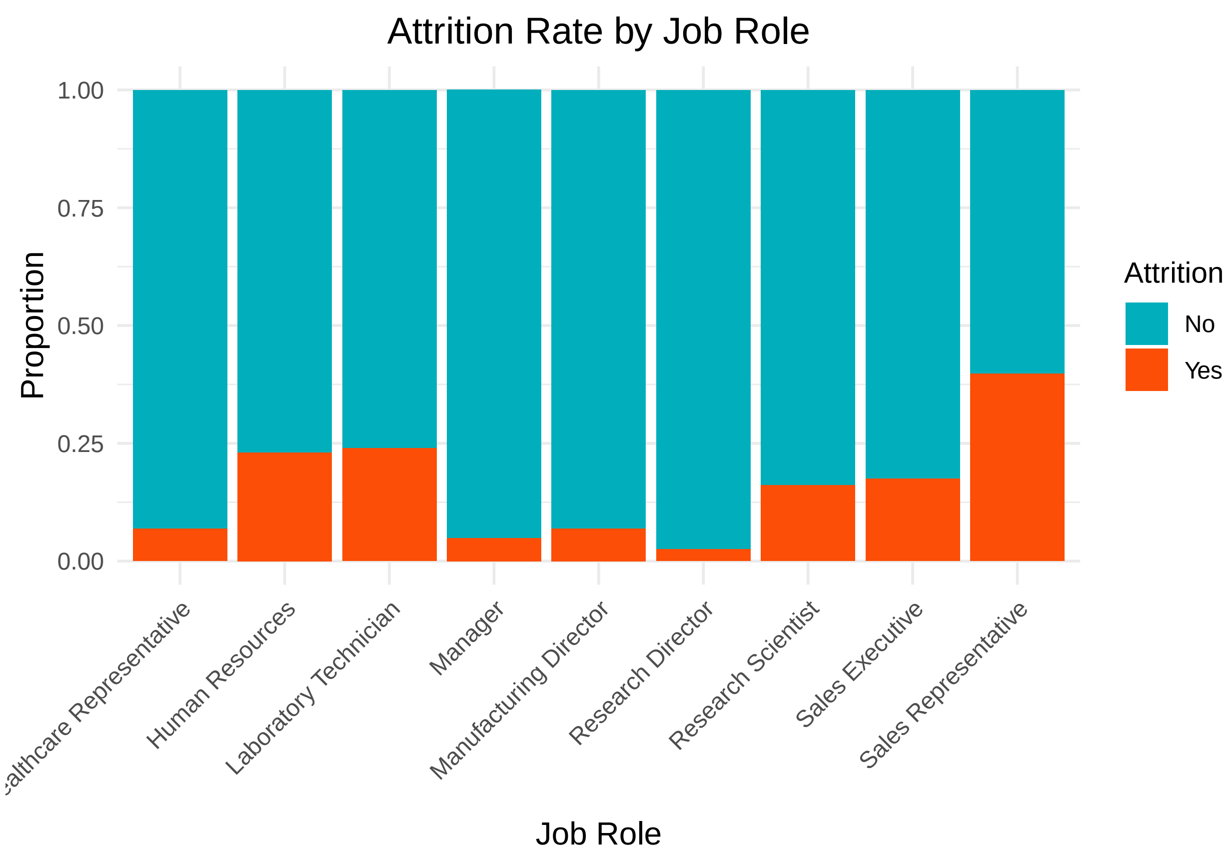
```
##
##        No       Yes
## 0.8387755 0.1612245
```

```r
# Plot 1: Attrition Class Distribution (Before SMOTE)
 ggplot(hr, aes(x=Attrition, fill=Attrition)) +
  geom_bar() +
  ggtitle("Attrition Class Distribution (Before SMOTE)") +
  theme_minimal() +
  labs(x="Attrition Status", y="Count") +
  scale_fill_manual(values=c("#00AFBB", "#FC4E07"))
```




```r
# Visualize Attrition by JobRole
# Visualize Attrition by JobRole (fixed overlapping labels)
ggplot(hr, aes(x = JobRole, fill = Attrition)) +
  geom_bar(position = "fill") +
  labs(title = "Attrition Rate by Job Role", x = "Job Role", y = "Proportion") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
```

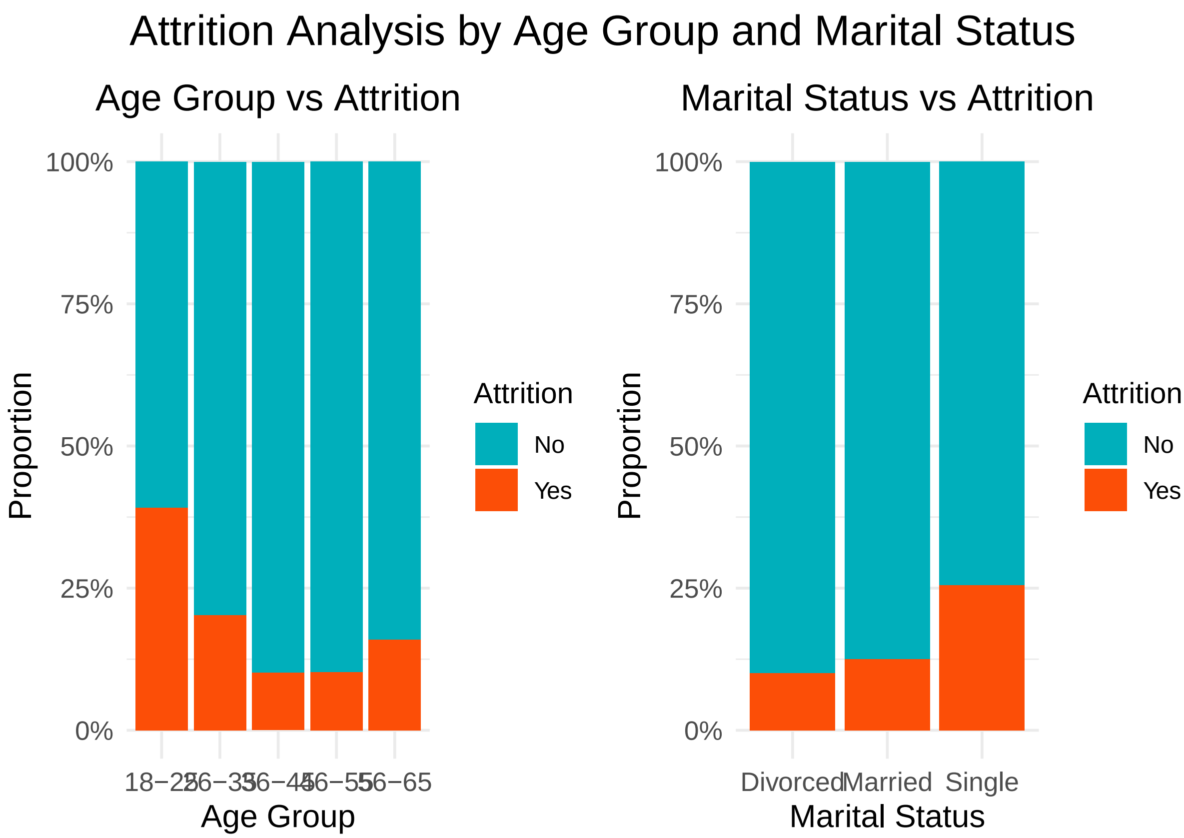
```r
    plot.title = element_text(hjust = 0.5, size = 14),
    axis.title = element_text(size = 12)
  ) +
  scale_fill_manual(values = c("#00AFBB", "#FC4E07"))
```




```r
# Create Age Groups
hr$AgeGroup <- cut(hr$Age,
                   breaks = c(18, 25, 35, 45, 55, 65),
                   labels = c("18-25", "26-35", "36-45", "46-55", "56-65"),
                   right = FALSE)
```

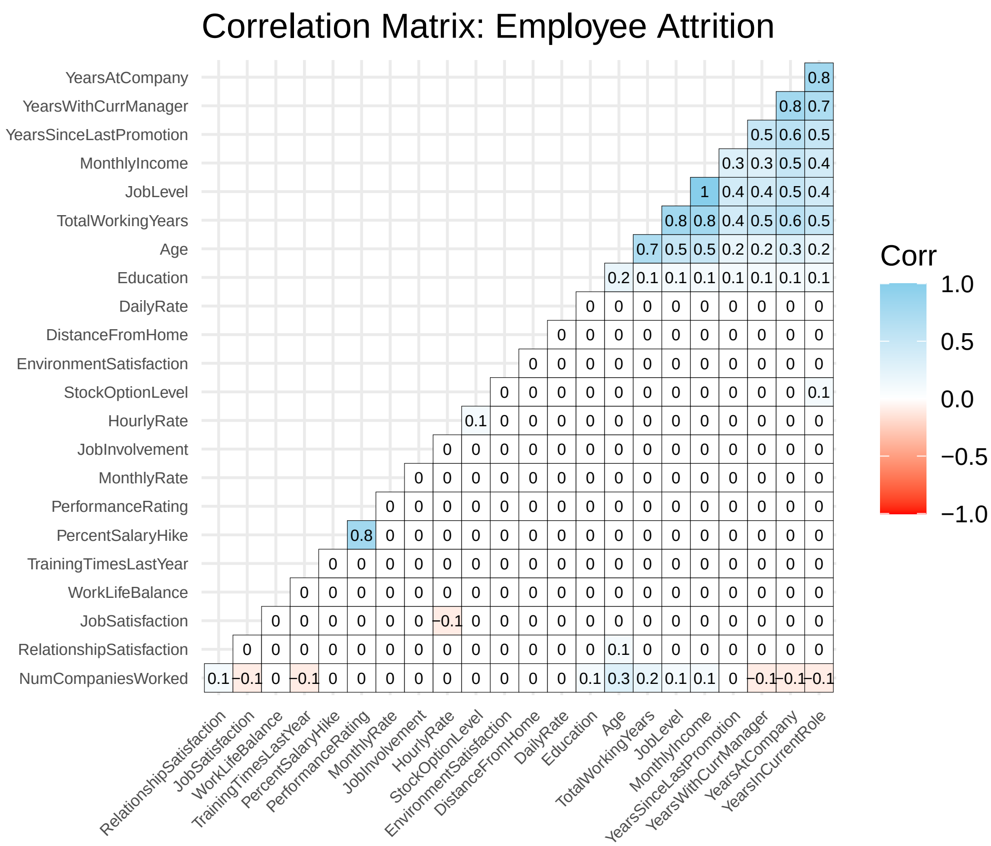
```r
# Plot 2: Age Group vs Attrition
p1 <- ggplot(hr, aes(x=AgeGroup, fill=Attrition)) +
  geom_bar(position="fill") +
  scale_y_continuous(labels=scales::percent) +
  labs(title="Age Group vs Attrition", x="Age Group", y="Proportion") +
  scale_fill_manual(values=c("#00AFBB", "#FC4E07")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size=14),
        axis.title = element_text(size=12),
        axis.text = element_text(size=10))

# Plot 3: Marital Status vs Attrition
p2 <- ggplot(hr, aes(x=MaritalStatus, fill=Attrition)) +
  geom_bar(position="fill") +
  scale_y_continuous(labels=scales::percent) +
  labs(title="Marital Status vs Attrition", x="Marital Status", y="Proportion") +
```

```r
  scale_fill_manual(values=c("#00AFBB", "#FC4E07")) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size=14),
        axis.title = element_text(size=12),
        axis.text = element_text(size=10))

# Combine side-by-side using patchwork
combined_plot <- (p1 | p2) +
  plot_annotation(title = "Attrition Analysis by Age Group and Marital Status",
                  theme = theme(plot.title = element_text(hjust = 0.5, size=16)))

print(combined_plot)
```




```r
##correlation matrix

library(dplyr)
library(ggcorrplot)

# Select only numeric columns and remove constant columns
nums <- hr %>%
  select_if(is.numeric) %>%
  select(where(~ var(.x, na.rm = TRUE) != 0))

# Compute correlation matrix
corr <- round(cor(nums, use = "complete.obs"), 1)

# Plot correlation matrix using ggcorrplot
```

```r
ggcorrplot(corr,
           type = "lower",
           lab = TRUE,
           lab_size = 2,
           method = "square",
           colors = c("red", "white", "skyblue"),
           title = "Correlation Matrix: Employee Attrition",
           hc.order = TRUE,
           hc.method = "complete",
           tl.cex = 6,
           outline.color = "black",
           ggtheme = theme_minimal() +
             theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)))
```



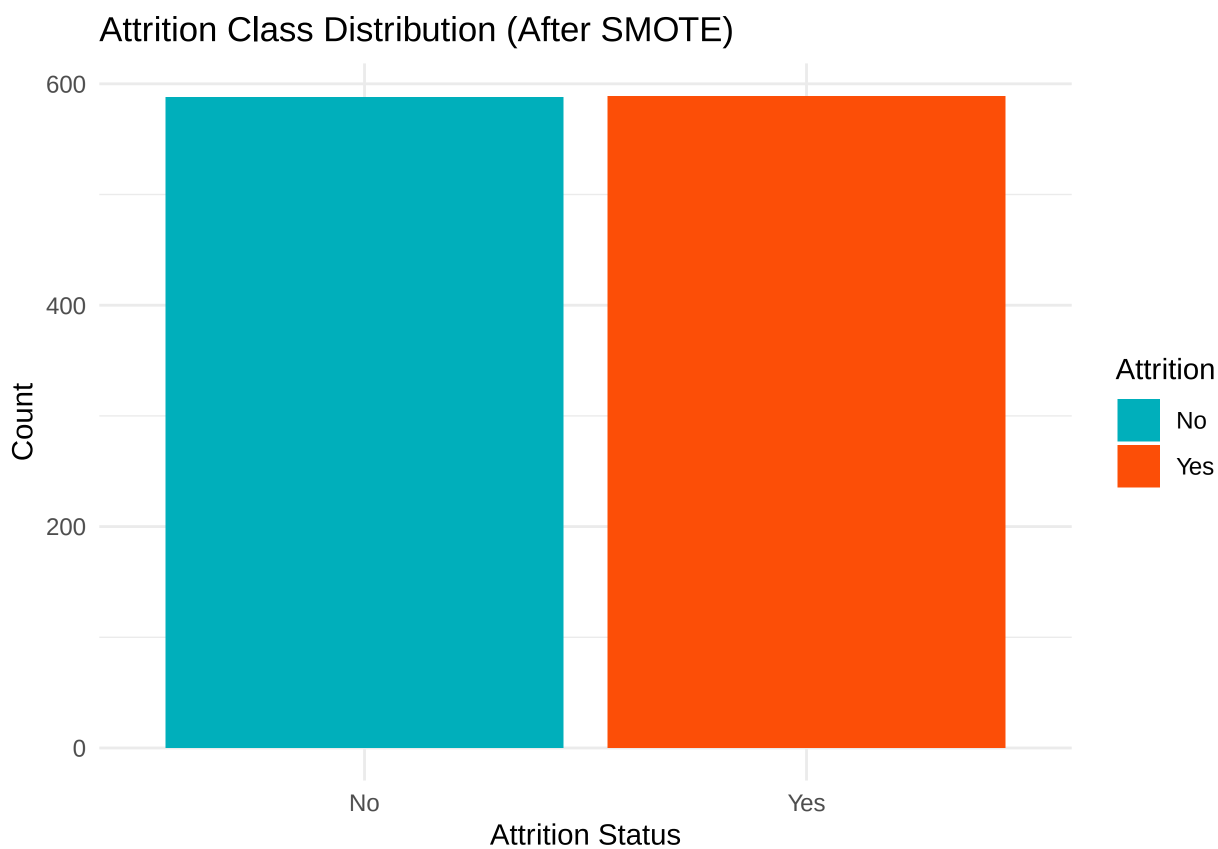

```r
#..................................Feature Engineering..................................

# Create bins for DistanceFromHome
hr$DistanceGroup <- cut(hr$DistanceFromHome,
                        breaks = c(0, 5, 15, 30),
                        labels = c("Near", "Medium", "Far"))

# Create bins for MonthlyIncome
hr$IncomeGroup <- cut(hr$MonthlyIncome,
                      breaks = quantile(hr$MonthlyIncome, probs=c(0, 0.33, 0.66, 1)),
                      labels = c("Low", "Medium", "High"),
                      include.lowest = TRUE)
```

```r
#.....................................Data Preprocessing.....................................
# Prepare X and y
# Data Preprocessing: Encode categorical variables
#dummies <- dummyVars(Attrition ~ ., data = hr, fullRank = TRUE)
#hr_transformed <- data.frame(predict(dummies, newdata = hr))
#hr_transformed$Attrition <- ifelse(hr$Attrition == "Yes", 1, 0)
#hr_transformed$Attrition <- as.factor(hr_transformed$Attrition)

# ---------------------- Split Data ----------------------
set.seed(999)
trainIndex <- createDataPartition(hr$Attrition, p = 0.8, list = FALSE)
train <- hr[trainIndex, ]
test <- hr[-trainIndex, ]


# ---------------------- SMOTE on Training Only ----------------------
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```r
train_new <- ROSE(Attrition ~ ., data = train, seed = 999)$data
# Train/test split
#train <- hr_balanced
#X_train <- train %>% select(-Attrition)
#y_train <- train$Attrition
#smote_output <- SMOTE(X_train, y_train, K = 5)
hr_balanced <- train_new
```

```r
# Bar plot - After SMOTE
ggplot(hr_balanced, aes(x=Attrition, fill=Attrition)) +
  geom_bar() +
  ggtitle("Attrition Class Distribution (After SMOTE)") +
  theme_minimal() +
  labs(x="Attrition Status", y="Count") +
  scale_fill_manual(values=c("#00AFBB", "#FC4E07"))
```




```r
#............................ [Logistic Regression].......................................

# Fit the logistic regression model
model_logit <- glm(Attrition ~ ., data=hr_balanced, family="binomial")
summary(model_logit)
```

```
##
## Call:
## glm(formula = Attrition ~ ., family = "binomial", data = hr_balanced)
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -1.118e+01  5.437e+02  -0.021 0.983587
## Age                              -1.972e-02  1.106e-02  -1.784 0.074444 .
## BusinessTravelTravel_Frequently   1.786e+00  3.321e-01   5.377 7.59e-08 ***
## BusinessTravelTravel_Rarely       9.295e-01  2.926e-01   3.177 0.001488 **
## DailyRate                        -5.048e-05  1.584e-04  -0.319 0.750011
## DepartmentResearch & Development  1.681e+01  5.437e+02   0.031 0.975339
## DepartmentSales                   1.520e+01  5.437e+02   0.028 0.977700
## DistanceFromHome                  1.891e-02  1.192e-02   1.587 0.112596
## Education                         1.314e-01  6.674e-02   1.969 0.048896 *
## EducationFieldLife Sciences      -2.799e+00  8.821e-01  -3.173 0.001509 **
## EducationFieldMarketing          -2.432e+00  9.171e-01  -2.652 0.008006 **
## EducationFieldMedical            -2.674e+00  8.746e-01  -3.057 0.002237 **
## EducationFieldOther              -2.180e+00  9.299e-01  -2.344 0.019075 *
## EducationFieldTechnical Degree   -1.427e+00  8.940e-01  -1.596 0.110526
## EnvironmentSatisfaction          -2.843e-01  5.811e-02  -4.892 9.97e-07 ***
```

```
## GenderMale                       -6.336e-02  1.689e-01  -0.375 0.707613
## HourlyRate                       -2.085e-03  3.180e-03  -0.656 0.512122
## JobInvolvement                   -2.592e-01  8.709e-02  -2.977 0.002914 **
## JobLevel                          8.517e-02  9.292e-02   0.917 0.359370
## JobRoleHuman Resources            1.618e+01  5.437e+02   0.030 0.976262
## JobRoleLaboratory Technician      9.186e-01  4.138e-01   2.220 0.026416 *
## JobRoleManager                    2.566e-01  5.298e-01   0.484 0.628101
## JobRoleManufacturing Director    -6.924e-01  4.576e-01  -1.513 0.130278
## JobRoleResearch Director         -1.692e+00  7.317e-01  -2.313 0.020742 *
## JobRoleResearch Scientist        -7.193e-02  4.281e-01  -0.168 0.866589
## JobRoleSales Executive            3.005e+00  7.778e-01   3.864 0.000112 ***
## JobRoleSales Representative       2.472e+00  8.627e-01   2.866 0.004163 **
## JobSatisfaction                  -2.153e-01  5.950e-02  -3.619 0.000296 ***
## MaritalStatusMarried              3.199e-01  2.329e-01   1.373 0.169635
## MaritalStatusSingle               1.371e+00  2.765e-01   4.957 7.15e-07 ***
## MonthlyIncome                    -2.522e-05  2.462e-05  -1.025 0.305482
## MonthlyRate                      -2.055e-05  9.026e-06  -2.277 0.022815 *
## NumCompaniesWorked                6.337e-02  2.669e-02   2.374 0.017603 *
## OverTimeYes                       2.009e+00  1.827e-01  10.997  < 2e-16 ***
## PercentSalaryHike                -4.848e-02  2.010e-02  -2.412 0.015886 *
## PerformanceRating                -1.043e-02  2.121e-01  -0.049 0.960793
## RelationshipSatisfaction         -1.326e-01  6.086e-02  -2.179 0.029317 *
## StockOptionLevel                 -1.133e-01  9.006e-02  -1.258 0.208392
## TotalWorkingYears                -7.542e-03  1.162e-02  -0.649 0.516260
## TrainingTimesLastYear            -2.350e-01  5.212e-02  -4.508 6.53e-06 ***
## WorkLifeBalance                  -2.390e-01  8.371e-02  -2.855 0.004301 **
## YearsAtCompany                    8.575e-03  1.373e-02   0.625 0.532254
## YearsInCurrentRole               -6.703e-02  2.404e-02  -2.788 0.005309 **
## YearsSinceLastPromotion           7.419e-02  2.396e-02   3.096 0.001963 **
## YearsWithCurrManager             -4.150e-02  2.225e-02  -1.865 0.062190 .
## AgeGroup26-35                    -5.497e-01  3.353e-01  -1.639 0.101126
## AgeGroup36-45                    -9.339e-01  3.910e-01  -2.389 0.016909 *
## AgeGroup46-55                    -7.237e-01  4.996e-01  -1.449 0.147422
## AgeGroup56-65                    -2.292e-01  6.351e-01  -0.361 0.718149
## DistanceGroupMedium               1.952e-01  2.156e-01   0.905 0.365257
## DistanceGroupFar                  4.549e-01  3.126e-01   1.455 0.145675
## IncomeGroupMedium                -1.347e+00  2.780e-01  -4.846 1.26e-06 ***
## IncomeGroupHigh                  -3.072e-01  4.128e-01  -0.744 0.456679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1631.7  on 1176  degrees of freedom
## Residual deviance: 1026.5  on 1124  degrees of freedom
## AIC: 1132.5
##
## Number of Fisher Scoring iterations: 14
```

```r
# Get predictions as probabilities (type = "response")
probabilities_rf <- predict(model_logit, newdata = test, type = "response")

# Convert probabilities to binary predictions (using threshold 0.5)
predictions_rf_class <- ifelse(probabilities_rf > 0.5, "Yes", "No")
```

```r
# Ensure both 'predictions_rf_class' and 'test$Attrition' are factors with the same levels
test$Attrition <- factor(test$Attrition, levels = c("No", "Yes"))
predictions_rf_class <- factor(predictions_rf_class, levels = c("No", "Yes"))

# Create confusion matrix (indicating positive class as 'Yes')
library(caret)
conf_logit <- confusionMatrix(predictions_rf_class, test$Attrition, positive = 'Yes')

# Print confusion matrix
conf_logit
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  195  16
##        Yes  51  31
##
##                Accuracy : 0.7713
##                  95% CI : (0.7189, 0.8182)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 0.9991
##
##                   Kappa : 0.3476
##
##  Mcnemar's Test P-Value : 3.271e-05
##
##             Sensitivity : 0.6596
##             Specificity : 0.7927
##          Pos Pred Value : 0.3780
##          Neg Pred Value : 0.9242
##              Prevalence : 0.1604
##          Detection Rate : 0.1058
##    Detection Prevalence : 0.2799
##       Balanced Accuracy : 0.7261
##
##        'Positive' Class : Yes
##
```

```r
# Create ROC object
roc_logit <- roc(test$Attrition, probabilities_rf)
```

```
## Setting levels: control = No, case = Yes
```

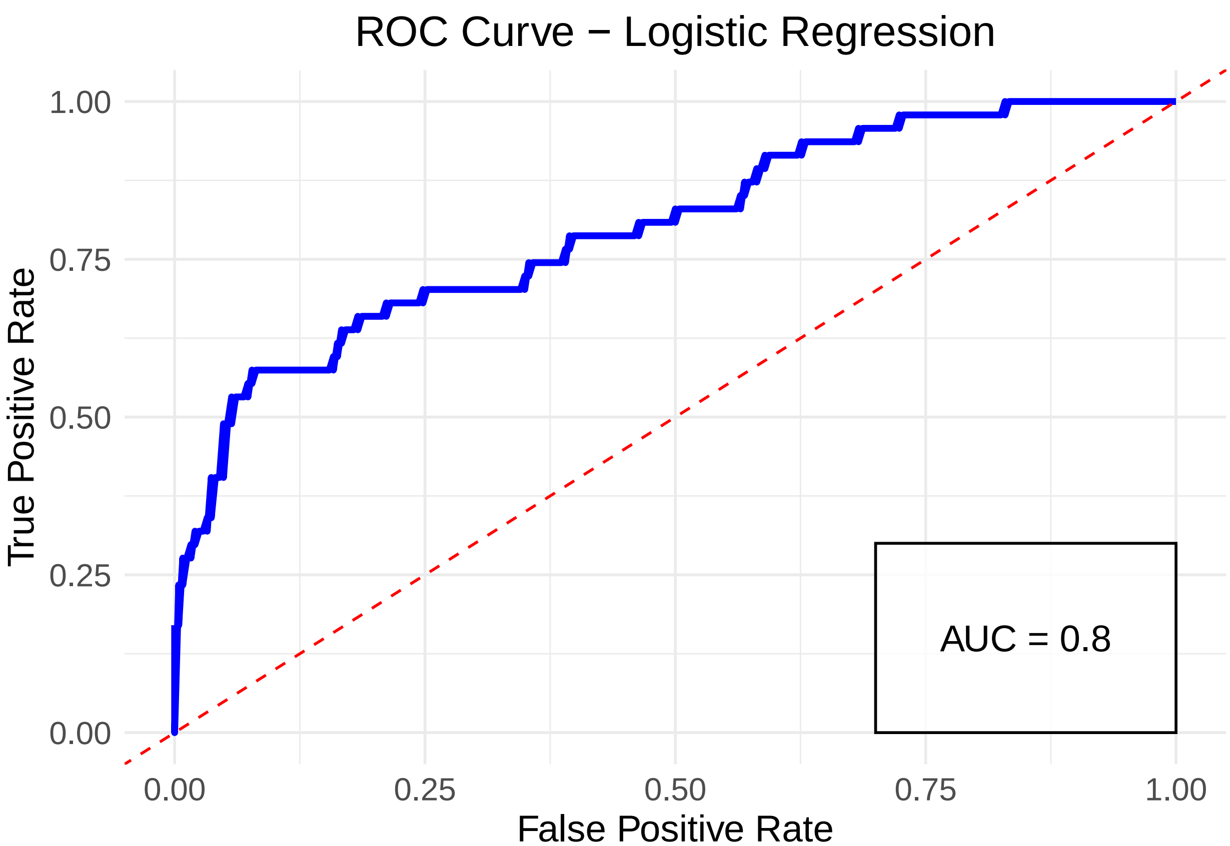
```
## Setting direction: controls < cases
```

```r
# Prepare data
df_logit <- data.frame(
  fpr = 1 - roc_logit$specificities,
  tpr = roc_logit$sensitivities
)
```

```r
# Plot
ggplot(df_logit, aes(x=fpr, y=tpr)) +
  geom_line(color="blue", size=1.2) +
```

```r
  geom_abline(intercept=0, slope=1, linetype="dashed", color="red") +
  labs(title="ROC Curve - Logistic Regression",
       x="False Positive Rate", y="True Positive Rate") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 16),
        axis.title = element_text(size = 14),
        axis.text = element_text(size = 12)) +
  annotate("rect", xmin = 0.7, xmax = 1, ymin = 0, ymax = 0.3,
           fill = "white", color = "black", alpha = 0.8) +
  annotate("text", x = 0.85, y = 0.15,
           label = paste("AUC =", round(auc(roc_logit), 3)), size = 5)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

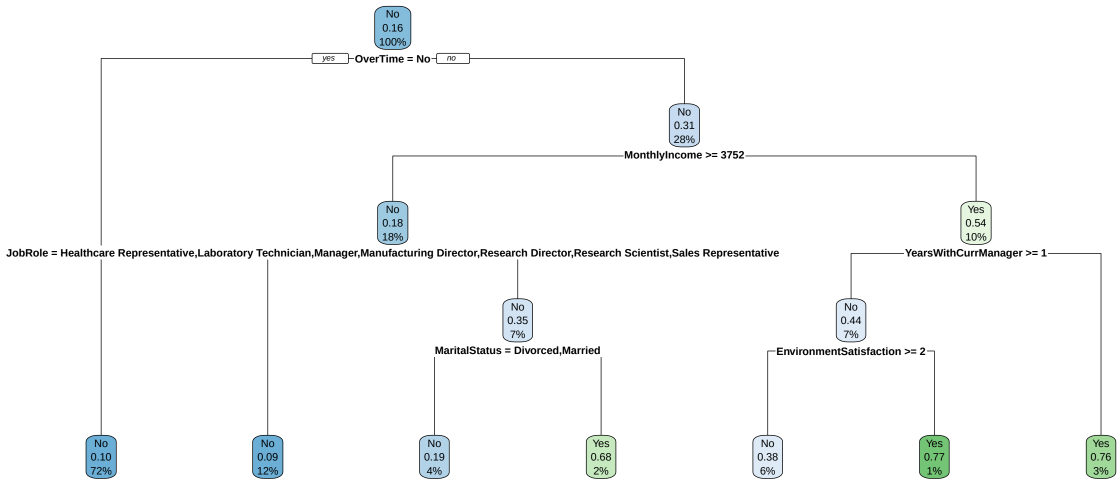



```r
#.................................[Decision Tree]................................................

library(rpart)
model_tree <- rpart(Attrition ~ ., data=hr_balanced, method="class")

library(rpart.plot)
# Set control parameters to make tree smaller
control <- rpart.control(maxdepth = 4, minsplit = 30, cp = 0.006)
prob_tree <- predict(model_tree, newdata=test, type="prob")[,2]
```

```r
conf_tree <- confusionMatrix(as.factor(predict(model_tree, newdata=test, type="class")), as.factor(test$
conf_tree
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  188  17
##        Yes  58  30
##
##                Accuracy : 0.744
##                  95% CI : (0.69, 0.793)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2975
##
##  Mcnemar's Test P-Value : 3.86e-06
##
##             Sensitivity : 0.7642
##             Specificity : 0.6383
##          Pos Pred Value : 0.9171
##          Neg Pred Value : 0.3409
##              Prevalence : 0.8396
##          Detection Rate : 0.6416
##    Detection Prevalence : 0.6997
##       Balanced Accuracy : 0.7013
##
##        'Positive' Class : No
##
```

```r
# Build a new tree
model_tree_simple <- rpart(Attrition ~ ., data=train, method="class", control=control)

# Plot the simpler tree
rpart.plot(model_tree_simple, extra = 106, fallen.leaves = TRUE)
```

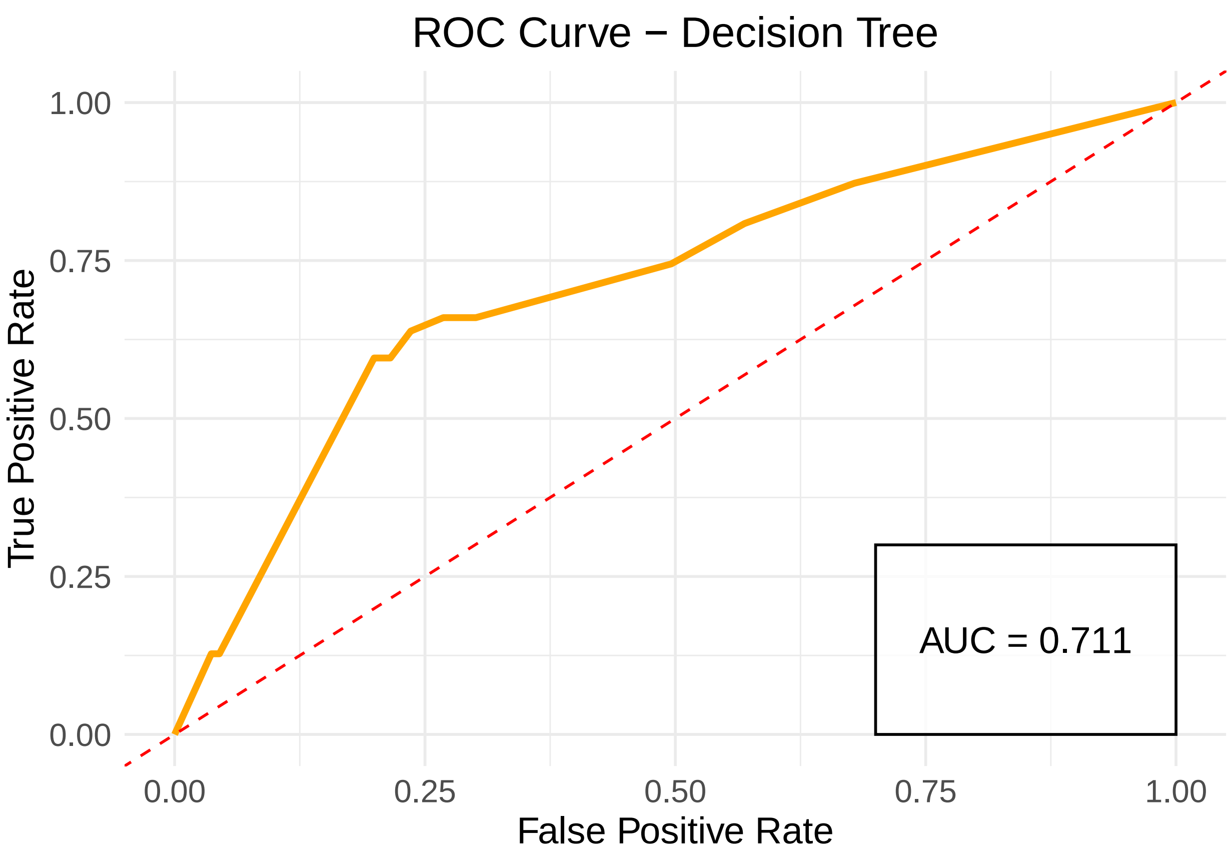



```r
# Create ROC object
roc_tree <- roc(test$Attrition, prob_tree)
```

```
## Setting levels: control = No, case = Yes

## Setting direction: controls < cases
```

```r
# Prepare data
df_tree <- data.frame(
  fpr = 1 - roc_tree$specificities,
  tpr = roc_tree$sensitivities
)
```

```r
# Plot
ggplot(df_tree, aes(x=fpr, y=tpr)) +
  geom_line(color="orange", size=1.2) +
  geom_abline(intercept=0, slope=1, linetype="dashed", color="red") +
  labs(title="ROC Curve - Decision Tree",
       x="False Positive Rate", y="True Positive Rate") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 16),
        axis.title = element_text(size = 14),
        axis.text = element_text(size = 12)) +
  annotate("rect", xmin = 0.7, xmax = 1, ymin = 0, ymax = 0.3,
           fill = "white", color = "black", alpha = 0.8) +
  annotate("text", x = 0.85, y = 0.15,
           label = paste("AUC =", round(auc(roc_tree), 3)), size = 5)
```



#...[SVM]..

```r
model_svm <- svm(as.factor(Attrition) ~ ., data=hr_balanced, kernel="linear", probability=TRUE)
summary(model_svm)
```

```
##
## Call:
## svm(formula = as.factor(Attrition) ~ ., data = hr_balanced, kernel = "linear",
##     probability = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  1
##
## Number of Support Vectors:  598
##
##  ( 297 301 )
##
##
## Number of Classes:  2
##
## Levels:
##  No Yes
```

```r
# Predict classes
pred_svm <- predict(model_svm, newdata=test, probability=TRUE)

# Confusion Matrix
conf_svm <- confusionMatrix(as.factor(pred_svm), as.factor(test$Attrition))
conf_svm
```

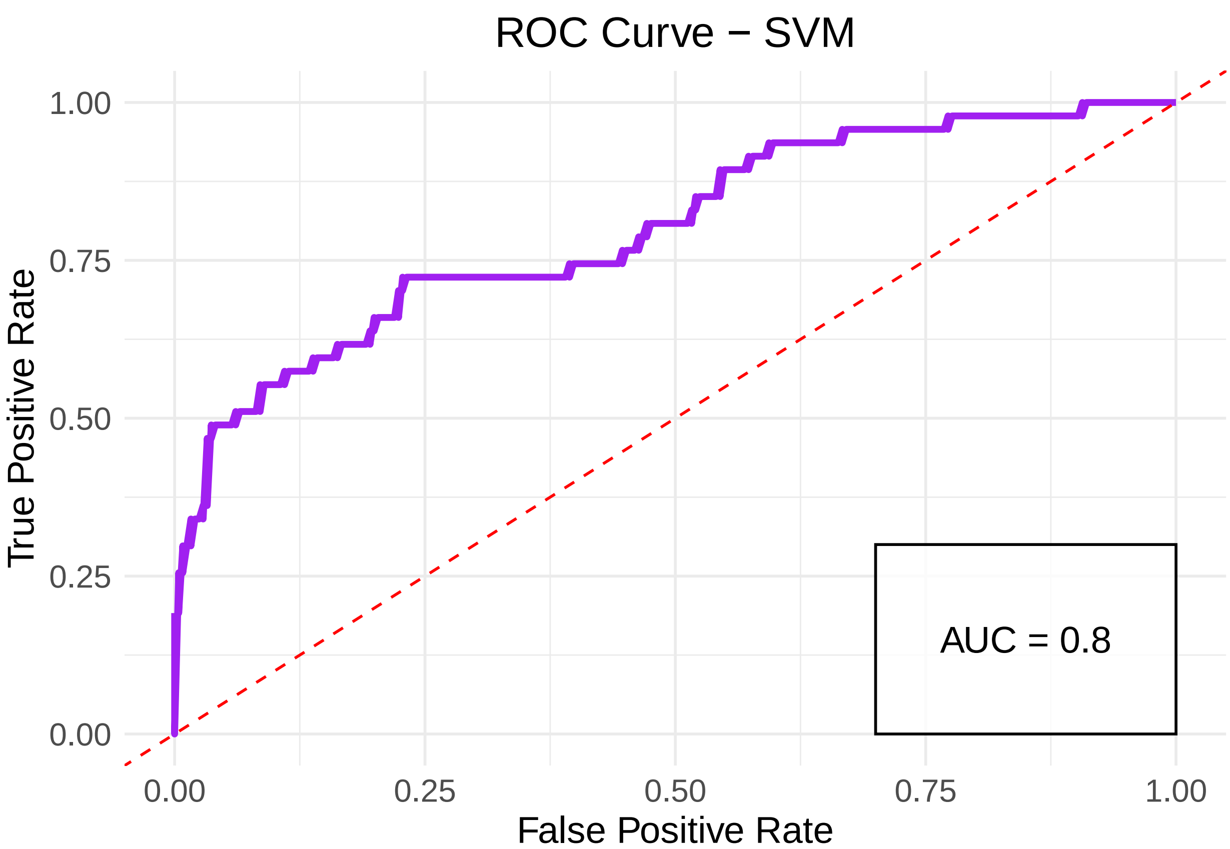
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  196  16
##        Yes  50  31
##
##                Accuracy : 0.7747
##                  95% CI : (0.7225, 0.8213)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 0.9985
##
##                   Kappa : 0.353
##
##  Mcnemar's Test P-Value : 4.865e-05
##
##             Sensitivity : 0.7967
##             Specificity : 0.6596
##          Pos Pred Value : 0.9245
##          Neg Pred Value : 0.3827
##              Prevalence : 0.8396
##          Detection Rate : 0.6689
##    Detection Prevalence : 0.7235
##       Balanced Accuracy : 0.7282
##
##        'Positive' Class : No
##
```

```r
# Get probabilities
prob_svm <- attr(predict(model_svm, newdata=test, probability=TRUE), "probabilities")[,2]

# ROC Curve
# Create ROC object
roc_svm <- roc(test$Attrition, prob_svm)
```

```
## Setting levels: control = No, case = Yes
## Setting direction: controls < cases
```

```r
# Prepare data
df_svm <- data.frame(
  fpr = 1 - roc_svm$specificities,
  tpr = roc_svm$sensitivities
)
```

```r
# Plot
ggplot(df_svm, aes(x=fpr, y=tpr)) +
  geom_line(color="purple", size=1.2) +
  geom_abline(intercept=0, slope=1, linetype="dashed", color="red") +
  labs(title="ROC Curve - SVM",
       x="False Positive Rate", y="True Positive Rate") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 16),
        axis.title = element_text(size = 14),
        axis.text = element_text(size = 12)) +
  annotate("rect", xmin = 0.7, xmax = 1, ymin = 0, ymax = 0.3,
           fill = "white", color = "black", alpha = 0.8) +
  annotate("text", x = 0.85, y = 0.15,
           label = paste("AUC =", round(auc(roc_svm), 3)), size = 5)
```

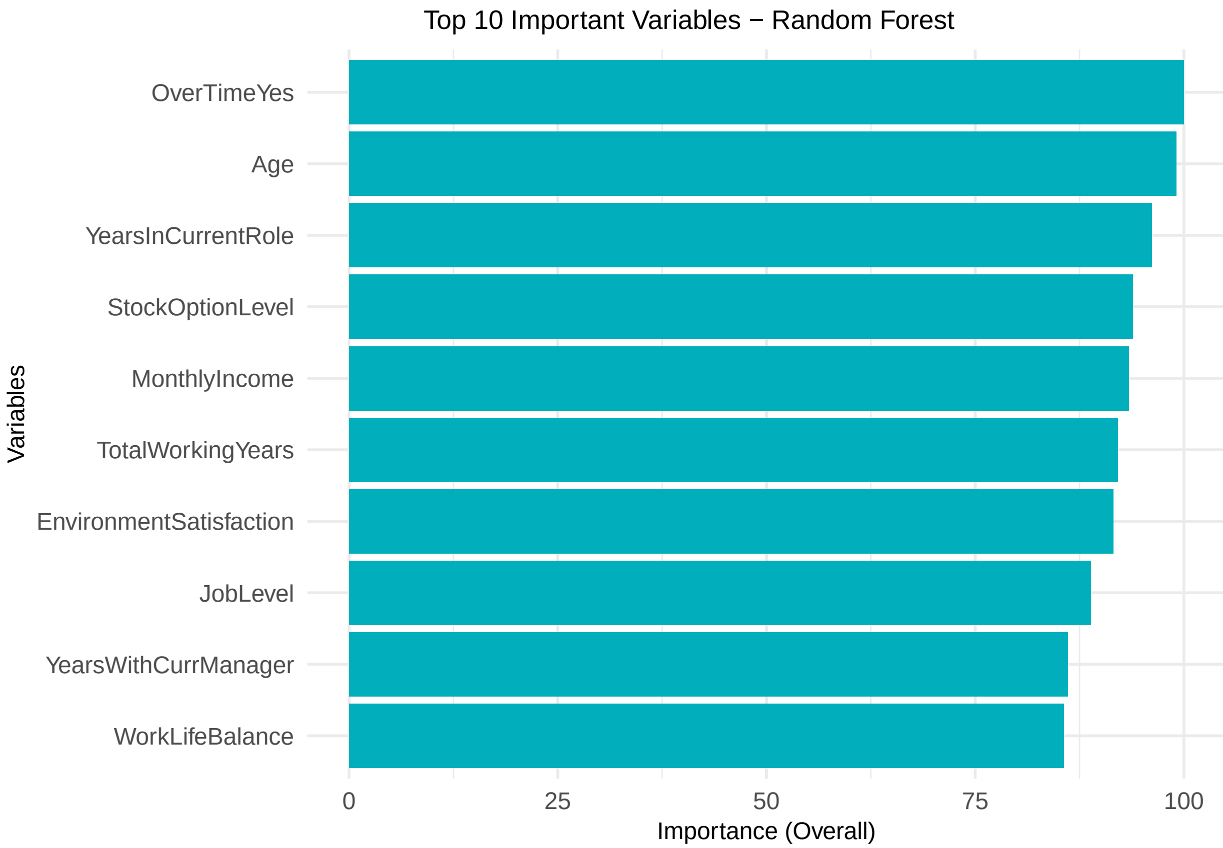



```r
#.....................................[Random Forest]......................................
library(caret)
library(randomForest)
library(dplyr)
library(ggplot2)

table(train$Attrition)
```

```
##
##   No Yes
## 987 190
```

```r
# Set up training control
train_control <- trainControl(method = "cv", number = 5)

# Train model using caret with method = 'rf'
model_rf <- train(as.factor(Attrition) ~ .,
                  data = hr_balanced,
                  method = "rf",
                  trControl = train_control)

# Get variable importance
importance_rf <- varImp(model_rf)$importance

# Convert rownames to a proper column for plotting
importance_rf <- importance_rf %>%
  mutate(Variable = rownames(.)) %>%
  arrange(desc(Overall)) %>%
```

```r
  top_n(10, Overall)  # Optional: top 10 variables

# Plot variable importance
ggplot(importance_rf, aes(x = reorder(Variable, Overall), y = Overall)) +
  geom_col(fill = "#00AFBB") +
  coord_flip() +
  labs(title = "Top 10 Important Variables - Random Forest",
       x = "Variables", y = "Importance (Overall)") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.3, size = 10),
        axis.title = element_text(size = 9),
        axis.text = element_text(size = 9))
```




```r
# Predict Probabilities for Random Forest
prob_rf <- predict(model_rf, newdata=test %>% select(-Attrition), type="prob")[,2]
pred_rf <- predict(model_rf, newdata=test %>% select(-Attrition))
conf_rf <- confusionMatrix(as.factor(pred_rf), as.factor(test$Attrition))
conf_rf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  206  25
##        Yes  40  22
##
##                Accuracy : 0.7782
```

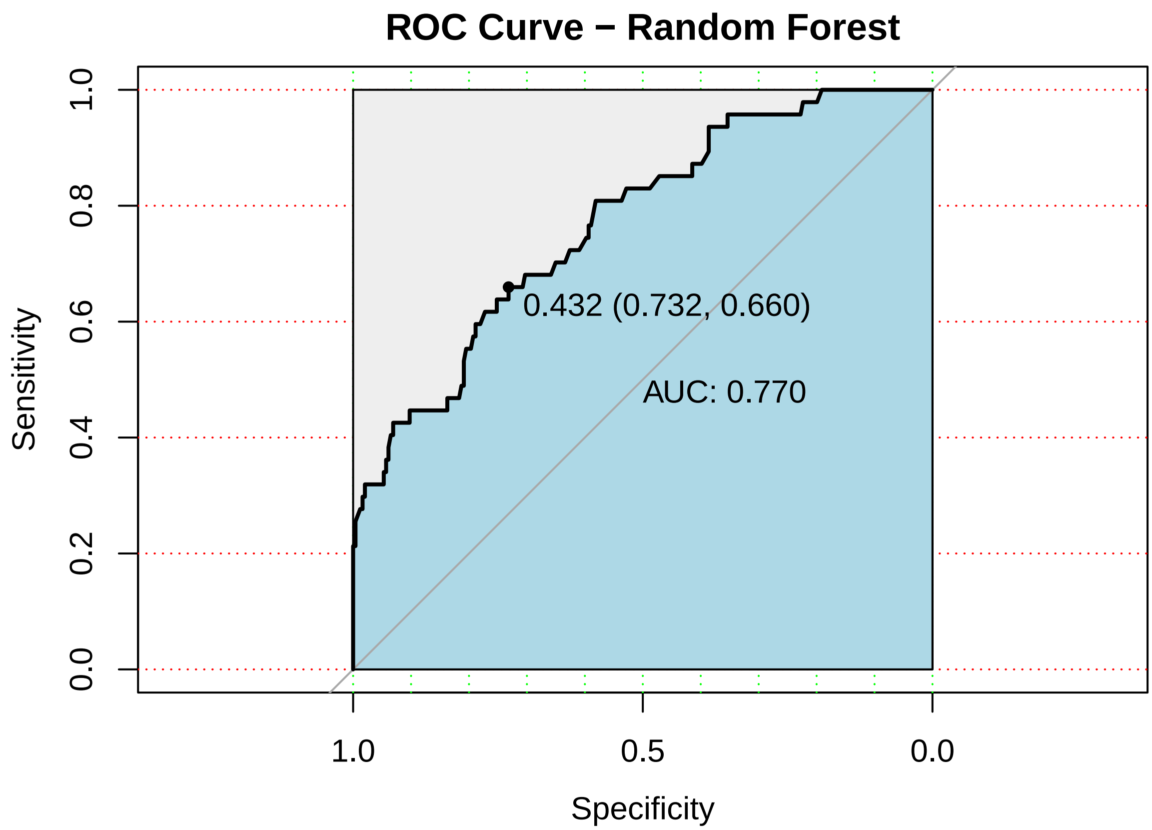
```
##                  95% CI : (0.7262, 0.8244)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 0.99766
##
##                   Kappa : 0.2706
##
##  Mcnemar's Test P-Value : 0.08248
##
##             Sensitivity : 0.8374
##             Specificity : 0.4681
##          Pos Pred Value : 0.8918
##          Neg Pred Value : 0.3548
##              Prevalence : 0.8396
##          Detection Rate : 0.7031
##    Detection Prevalence : 0.7884
##       Balanced Accuracy : 0.6527
##
##        'Positive' Class : No
##
```

```r
# ROC Curve for Random Forest
prob_rf <- predict(model_rf, test, type = "prob")[, 2]  # Probability of 'Yes'
roc_rf <- roc(test$Attrition, prob_rf, percent = FALSE)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```r
plot.roc(roc_rf,
         print.auc = TRUE,
         auc.polygon = TRUE,
         grid = c(0.1, 0.2),
         grid.col = c("green", "red"),
         max.auc.polygon = TRUE,
         auc.polygon.col = "lightblue",
         print.thres = TRUE,
         main = 'ROC Curve - Random Forest')
```




```r
# ......................................[XGBoost]..........................................

# Define cross-validation control
cvcontrol <- trainControl(
  method = "repeatedcv",              # Use repeated cross-validation
  number = 5,                         # 5-fold CV
  repeats = 1,                        # Repeat CV 3 times
  classProbs = TRUE,                  # Enable class probabilities
  summaryFunction = twoClassSummary, # Use AUC as the performance metric
  search = "random"                   # Randomized search
)
# Replace invalid characters with underscores and ensure valid factor names
levels(train$Attrition) <- make.names(levels(train$Attrition))

# Perform Randomized Search with fewer hyperparameter combinations
set.seed(123)
model_xgb <- train(as.factor(Attrition) ~ .,
                   data = hr_balanced,
                   method = "xgbTree",
                   trControl = cvcontrol,
                   tuneLength = 10,
                   metric = "ROC"
)

# Print the best model parameters
print(model_xgb$bestTune)
```

```
##    nrounds max_depth       eta    gamma colsample_bytree min_child_weight
```

```
## 3     526          9 0.1742067 3.18181        0.4063891                 4
##   subsample
## 3 0.6064874
```

```
# Plot the top 15 variable importance
var_imp <- varImp(model_xgb, scale = TRUE)
top_15_vars <- head(var_imp$importance, 15)
```

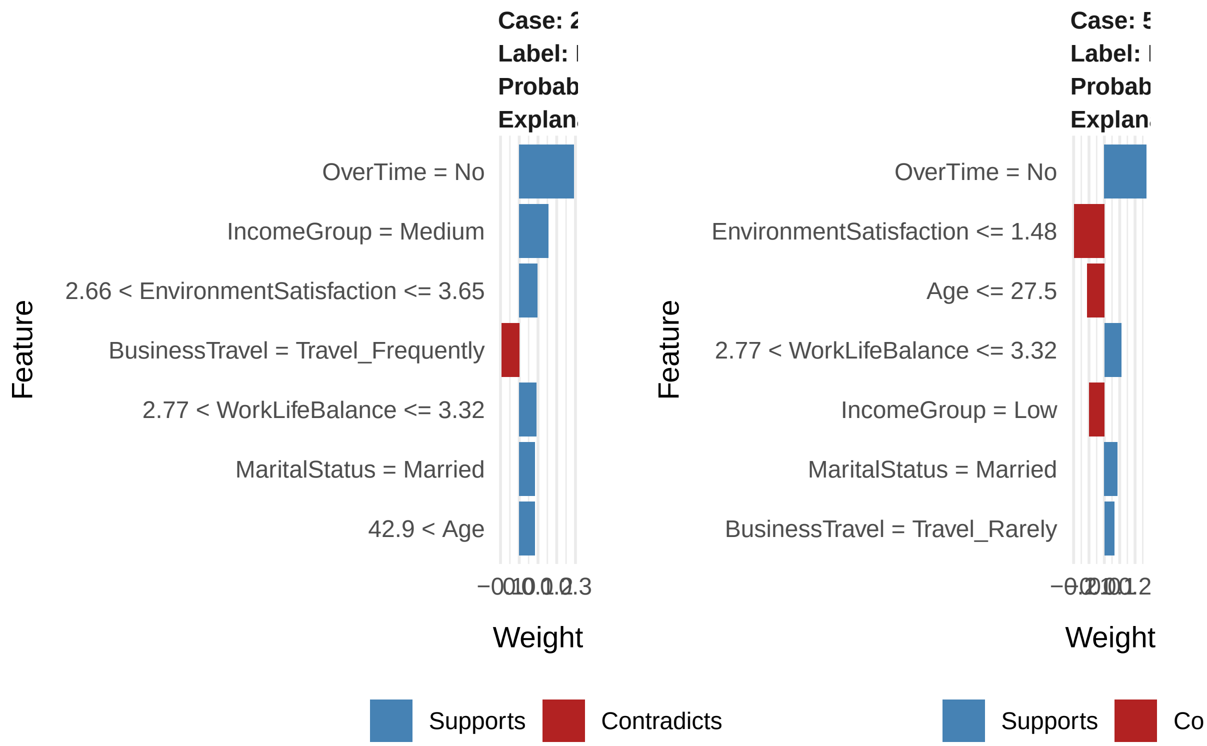
```
# Install or update lime package

library(lime)
```

```
##
## Attaching package: 'lime'

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

```
library(xgboost)
library(caret)
library(tidyverse)
library(gridExtra)
```

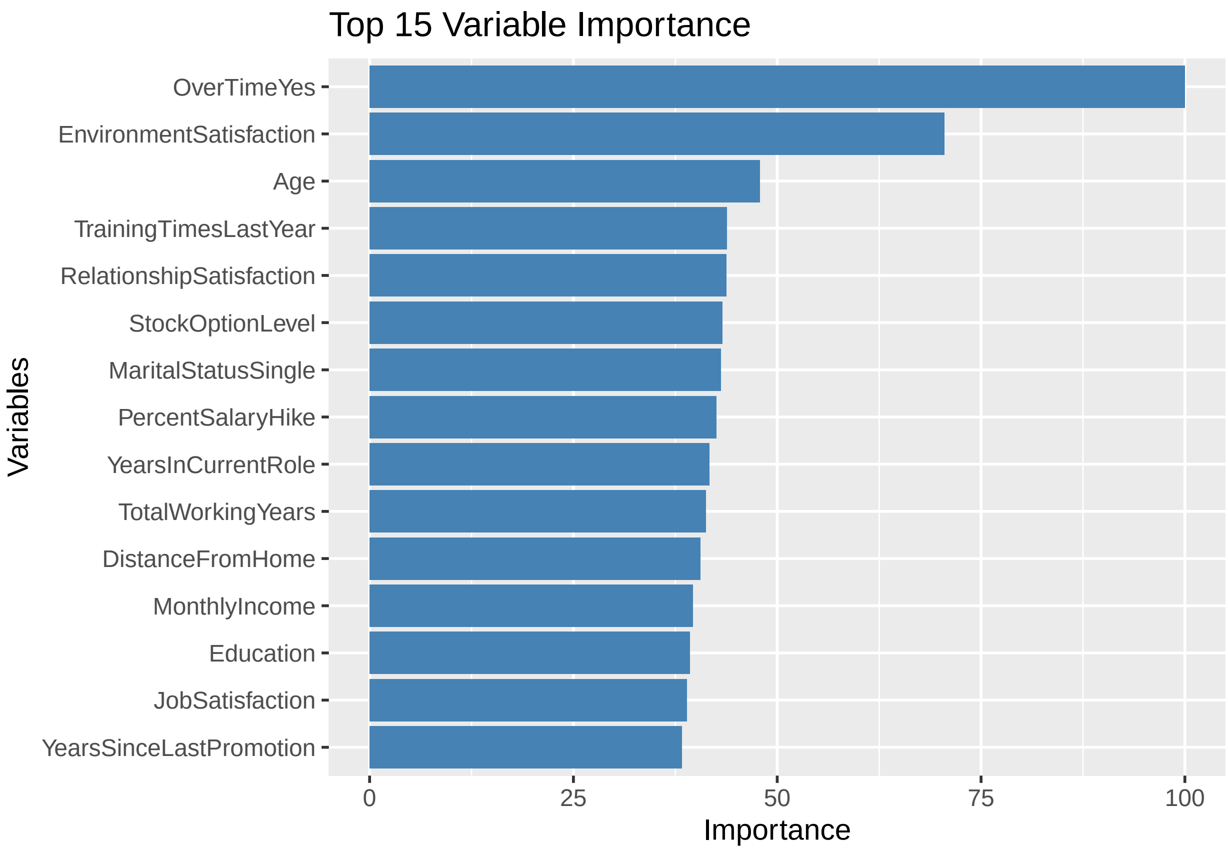
```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:randomForest':
##
##     combine

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

```
# Create the LIME explainer object
explainer_xgb <- lime::lime(hr_balanced[, -which(names(hr_balanced) == "Attrition")], model_xgb)

# Choose 4 different test samples (or rows from your test dataset)
test_samples <- test[1:2, -which(names(test) == "Attrition")]

# Create the explanations for each of the 2 test samples
explanations <- lapply(1:2, function(i) {
  lime::explain(test_samples[i, , drop = FALSE], explainer_xgb, n_labels = 1, n_features = 7)
})

# Generate the LIME plots for feature importances
plots <- lapply(explanations, function(explanation) {
  lime::plot_features(explanation)  # Make sure this generates the feature importance plot
})

 plot1 <- lime::plot_features(explanations[[1]])
 plot2 <- lime::plot_features(explanations[[2]])

 combined_plots <- plot1 + plot2 + plot_layout(ncol = 2)
 print(combined_plots)
```




```r
grid_plots <- wrap_plots(combined_plots, ncol = 1, nrow = 4)
```

```r
# Plot variable importance
ggplot(top_15_vars, aes(x = reorder(rownames(top_15_vars), Overall), y = Overall)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  xlab("Variables") +
  ylab("Importance") +
  ggtitle("Top 15 Variable Importance")
```

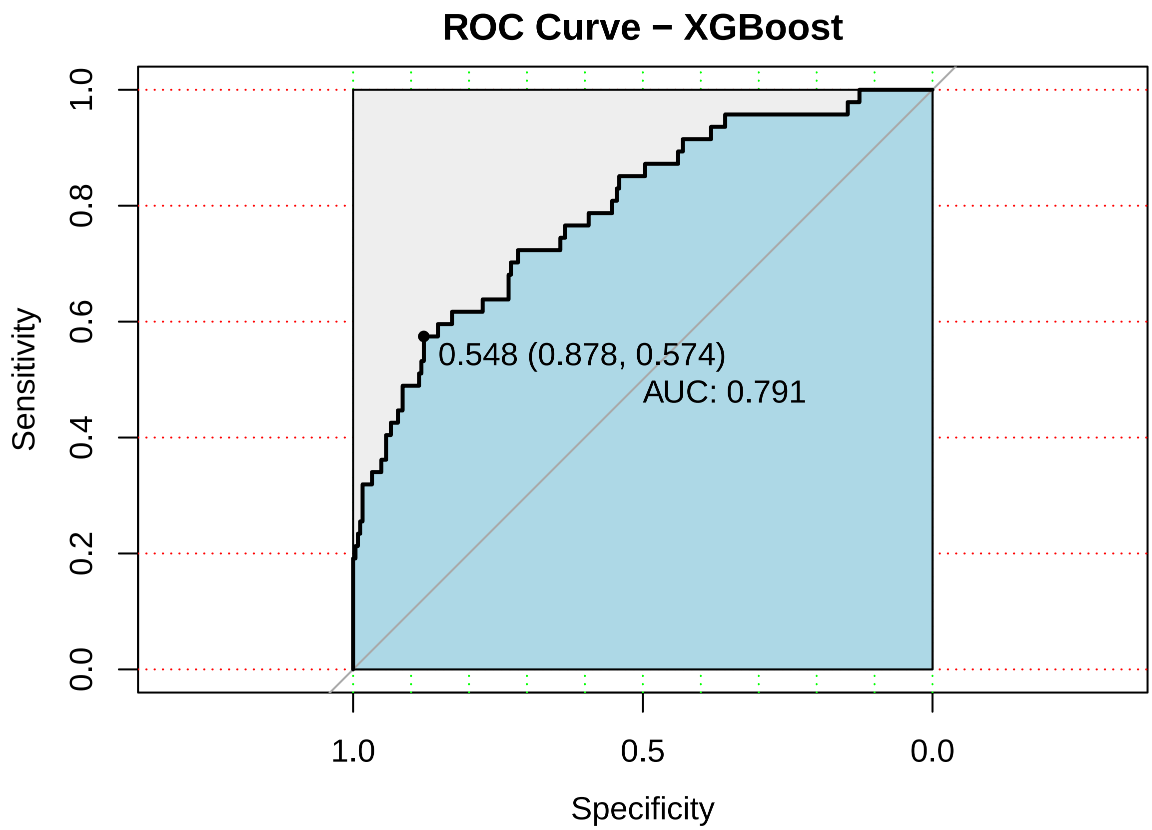



```r
# Get the model predictions
prob_pred <- predict(model_xgb, newdata = test, type = "prob")[,2]

roc_xgb <- roc(test$Attrition, prob_pred, percent = FALSE)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```r
plot.roc(roc_xgb,
         print.auc = TRUE,
         auc.polygon = TRUE,
         grid = c(0.1, 0.2),
         grid.col = c("green", "red"),
         max.auc.polygon = TRUE,
         auc.polygon.col = "lightblue",
         print.thres = TRUE,
         main = 'ROC Curve - XGBoost')
```




```r
# Predict Probabilities for XGB
prob_xgb <- predict(model_xgb, newdata=test, type="prob")[,2]
pred_xgb <- predict(model_xgb, newdata=test )
conf_xgb <- confusionMatrix((pred_xgb), (test$Attrition))
conf_xgb
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  209  19
##        Yes  37  28
##
##                Accuracy : 0.8089
##                  95% CI : (0.7591, 0.8523)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 0.9321
##
##                   Kappa : 0.3856
##
##  Mcnemar's Test P-Value : 0.0231
##
##             Sensitivity : 0.8496
##             Specificity : 0.5957
##          Pos Pred Value : 0.9167
##          Neg Pred Value : 0.4308
##              Prevalence : 0.8396
##          Detection Rate : 0.7133
```

```
##    Detection Prevalence : 0.7782
##       Balanced Accuracy : 0.7227
##
##        'Positive' Class : No
##
```

```r
#.......................................Final Summary Table.............................

results_final <- data.frame(
  Model = c("Random Forest", "Logistic Regression", "XGBoost", "SVM", "Decision Tree"),

  AUC = round(c(
    auc(roc_rf),
    auc(roc_logit),
    auc(roc_xgb),
    auc(roc_svm),
    auc(roc_tree)
  ), 3),

  Accuracy = round(c(
    conf_rf$overall["Accuracy"],
    conf_logit$overall["Accuracy"],
    conf_xgb$overall["Accuracy"],
    conf_svm$overall["Accuracy"],
    conf_tree$overall["Accuracy"]
  ), 3),

  Sensitivity = round(c(
    conf_rf$byClass["Sensitivity"],
    conf_logit$byClass["Sensitivity"],
    conf_xgb$byClass["Sensitivity"],
    conf_svm$byClass["Sensitivity"],
    conf_tree$byClass["Sensitivity"]
  ), 3),

  Specificity = round(c(
    conf_rf$byClass["Specificity"],
    conf_logit$byClass["Specificity"],
    conf_xgb$byClass["Specificity"],
    conf_svm$byClass["Specificity"],
    conf_tree$byClass["Specificity"]
  ), 3),

  Balanced_Accuracy = round(c(
    conf_rf$byClass["Balanced Accuracy"],
    conf_logit$byClass["Balanced Accuracy"],
    conf_xgb$byClass["Balanced Accuracy"],
    conf_svm$byClass["Balanced Accuracy"],
    conf_tree$byClass["Balanced Accuracy"]
  ), 3)
)

# View final beautiful table
print(results_final)
```

```
##                 Model   AUC Accuracy Sensitivity Specificity Balanced_Accuracy
## 1       Random Forest 0.770    0.778       0.837       0.468             0.653
## 2 Logistic Regression 0.800    0.771       0.660       0.793             0.726
## 3             XGBoost 0.791    0.809       0.850       0.596             0.723
## 4                 SVM 0.800    0.775       0.797       0.660             0.728
## 5       Decision Tree 0.711    0.744       0.764       0.638             0.701
```