STAT 208 Project: Predicting Employee Attrition

Team Breaking Bias

Objective 1: Predicting Employee Attrition

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## -- 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
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.3
```

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

```
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
```

```
library(e1071)
```

```
train <- read.csv("D:/MASTERS/SPRING25/StatisticalDataMiningMethods_STAT208/Project/archive (1)/train.cs
test <- read.csv("D:/MASTERS/SPRING25/StatisticalDataMiningMethods_STAT208/Project/archive (1)/test.csv

# Convert target variable
train$Attrition <- ifelse(train$Attrition == "Left", 1, 0)
test$Attrition <- ifelse(test$Attrition == "Left", 1, 0)

# Drop Employee ID
train <- train %>% select(-Employee.ID)
test <- test %>% select(-Employee.ID)
```

```
# Convert character columns to factors
factor_cols <- sapply(train, is.character)
train[factor_cols] <- lapply(train[factor_cols], factor)
test[factor_cols] <- lapply(test[factor_cols], factor)
```

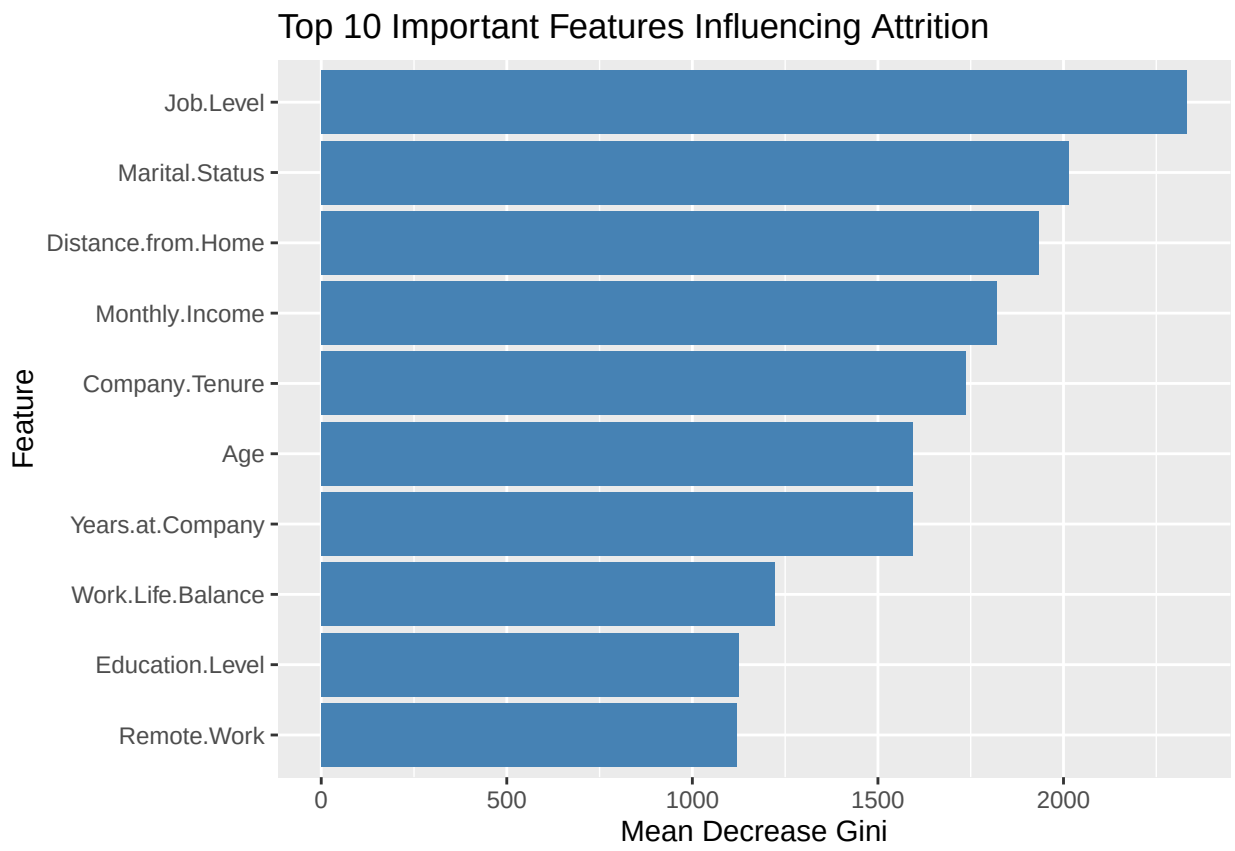
```
set.seed(123)
splitIndex <- createDataPartition(train$Attrition, p = 0.8, list = FALSE)
train_data <- train[splitIndex,]
val_data <- train[-splitIndex,]
train_data$Attrition <- as.factor(train_data$Attrition)
val_data$Attrition <- as.factor(val_data$Attrition)

model_rf <- randomForest(Attrition ~ ., data = train_data, importance = TRUE)
pred_rf <- predict(model_rf, val_data)
confusionMatrix(as.factor(pred_rf), as.factor(val_data$Attrition))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4752 1538
##          1 1492 4137
##
##                Accuracy : 0.7458
##                  95% CI : (0.7379, 0.7536)
##     No Information Rate : 0.5239
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.4902
##
##  Mcnemar's Test P-Value : 0.4136
##
##             Sensitivity : 0.7611
##             Specificity : 0.7290
##          Pos Pred Value : 0.7555
```

```
##          Neg Pred Value : 0.7349
##              Prevalence : 0.5239
##          Detection Rate : 0.3987
##    Detection Prevalence : 0.5277
##       Balanced Accuracy : 0.7450
##
##        'Positive' Class : 0
##
```

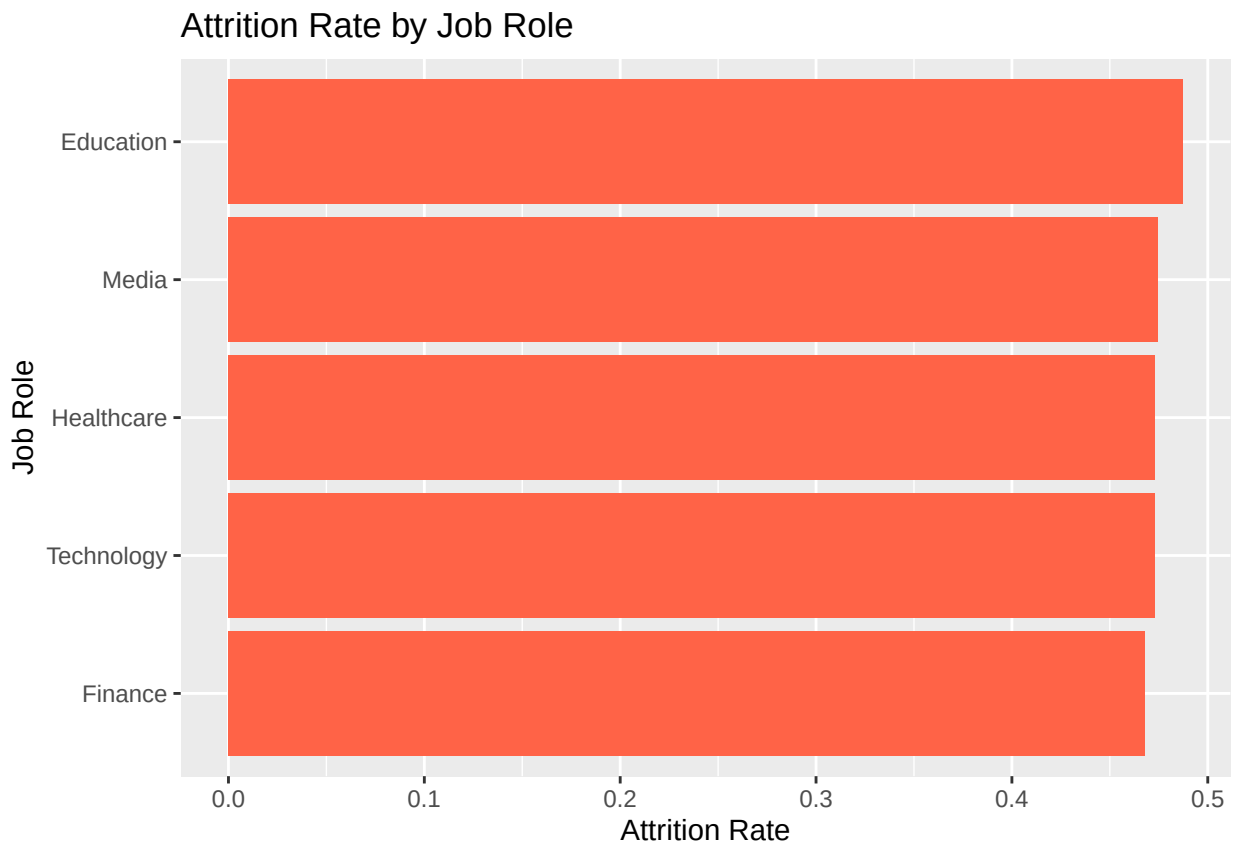
Objective 2: Feature Importance

```
importance_df <- as.data.frame(importance(model_rf))
importance_df$Feature <- rownames(importance_df)
importance_df <- importance_df %>% arrange(desc(MeanDecreaseGini))

# Plot top 10 features
top_features <- importance_df[1:10,]
ggplot(top_features, aes(x = reorder(Feature, MeanDecreaseGini), y = MeanDecreaseGini)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Top 10 Important Features Influencing Attrition",
       x = "Feature", y = "Mean Decrease Gini")
```



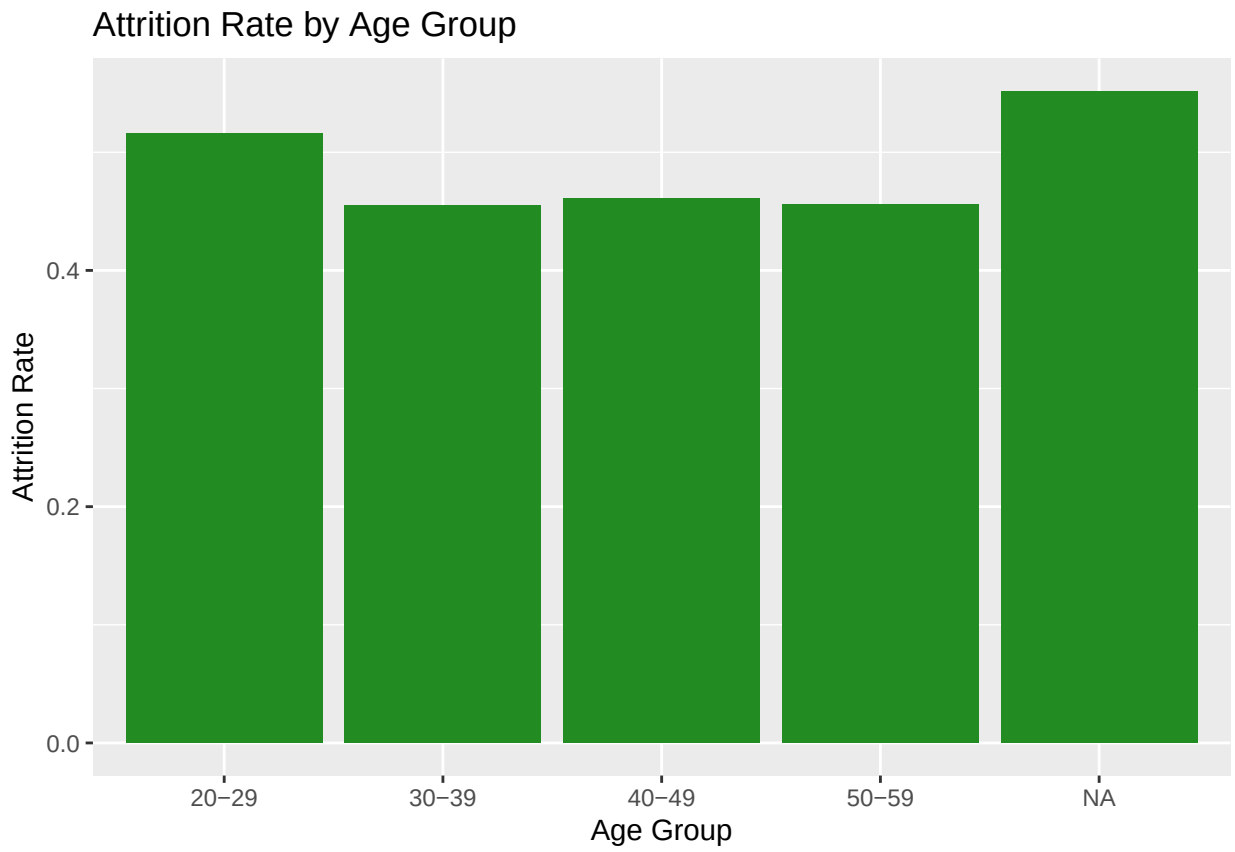
Objective 3: Department and Age Group Analysis

```r
# Attrition rate by department
dept_attr <- train %>%
  group_by(Job.Role) %>%
  summarise(AttritionRate = mean(Attrition)) %>%
  arrange(desc(AttritionRate))

ggplot(dept_attr, aes(x = reorder(Job.Role, AttritionRate), y = AttritionRate)) +
  geom_col(fill = "tomato") +
  coord_flip() +
  labs(title = "Attrition Rate by Job Role",
       x = "Job Role", y = "Attrition Rate")
```




```r
# Age group bins
train$AgeGroup <- cut(train$Age, breaks = c(20, 30, 40, 50, 60), right = FALSE,
                      labels = c("20-29", "30-39", "40-49", "50-59"))

age_attr <- train %>%
  group_by(AgeGroup) %>%
  summarise(AttritionRate = mean(Attrition))

ggplot(age_attr, aes(x = AgeGroup, y = AttritionRate)) +
  geom_col(fill = "forestgreen") +
```

```
  labs(title = "Attrition Rate by Age Group",
       x = "Age Group", y = "Attrition Rate")
```



Policy Recommendations

Offer flexible remote work options

Invest in mentorship and development programs for younger employees

Focus retention efforts on high-attrition departments

Use feature insights to guide HR policy decisions