Random Forest Model

Group F

2024-10-25

Load the dataset

```
# Load the dataset into the object "custData"
custData <- read.csv("CustData2_Prepared.csv")
custData$Eligible <- as.factor(custData$Eligible)
```

Split the dataset into training data and testing data

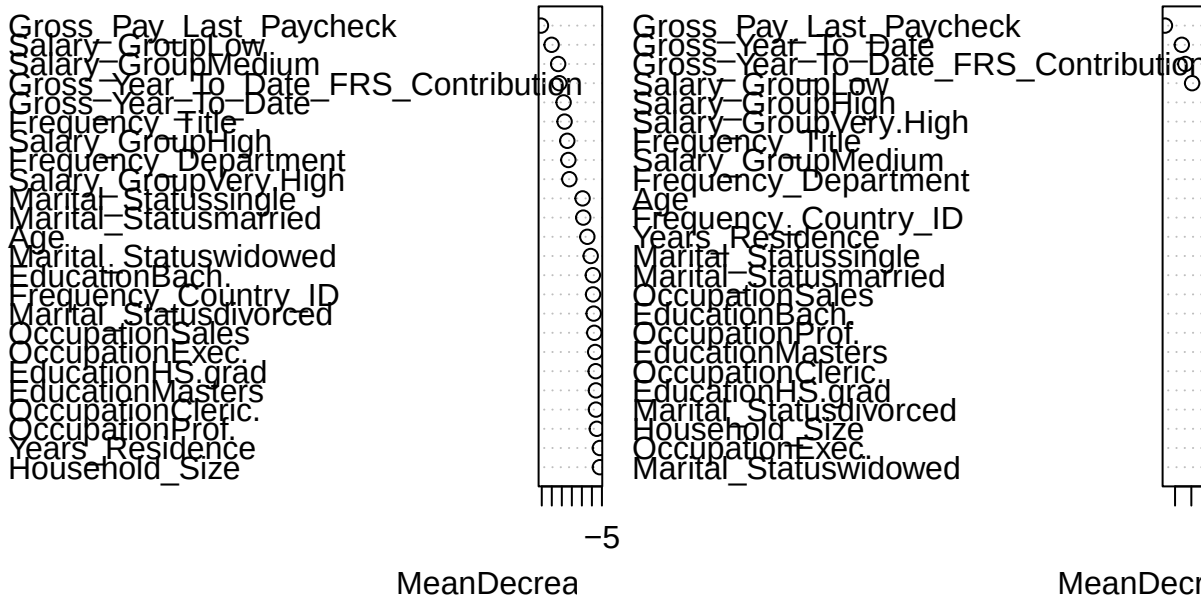
```
# Split the dataset to 80% training data and 20% testing data
set.seed(123)
train_index <- createDataPartition(custData$Eligible, p = 0.8, list = FALSE)
train_data <- custData[train_index, ]
test_data <- custData[-train_index, ]
```

Train random forest model

```
randomForest_model <- randomForest(Eligible ~ . -Annual_Salary,
                                   data = train_data, ntree = 100,
                                   mtry = 3, importance = TRUE)
```

Visualize attribute importance

```
#The attribute importance can be visualised using the random forest model
varImpPlot(randomForest_model)
```



Make predictions using random forest

```
randomForest_predictions <- predict(randomForest_model, newdata = test_data)
```

Confusion matrix

Create confusion matrix

```
# Confusion Matrix for Random Forest
randomForest_cm <- confusionMatrix(as.factor(randomForest_predictions),
                                   as.factor(test_data$Eligible))
randomForest_matrix <- randomForest_cm$table
```

Extract TP, TN, FP and FN from confusion matrix

```
# Extract TruePositive, TrueNegative, FalsePositive
# and FalseNegative for confusion matrix
randomForest_truePositive <- randomForest_matrix[1, 1]
```

```r
randomForest_trueNegative <- randomForest_matrix[2, 2]
randomForest_falsePositive <- randomForest_matrix[1, 2]
randomForest_falseNegative <- randomForest_matrix[2, 1]
```

Evaluation metrics

Calculate the evaluation metrics

```r
# Calculate Evaluation Metrics
randomForest_accuracy <-
  round(((sum(diag(randomForest_matrix)) / sum(randomForest_matrix))) * 100, 2)
randomForest_precision <-
  round((randomForest_truePositive / (randomForest_truePositive +
                                        randomForest_falsePositive)) * 100, 2)
randomForest_recall <-
  round((randomForest_truePositive / (randomForest_truePositive +
                                        randomForest_falseNegative)) * 100, 2)
randomForest_f1_score <-
  round(2 * (randomForest_precision * randomForest_recall) /
                             (randomForest_precision + randomForest_recall), 2)
```

Display the evaluation metrics

```r
cat("Random Forest Accuracy:", randomForest_accuracy, "% \n")
```

```
## Random Forest Accuracy: 97 %
```

```r
cat("Random Forest Precision:", randomForest_precision, "% \n")
```

```
## Random Forest Precision: 98.42 %
```

```r
cat("Random Forest Recall:", randomForest_recall, "% \n")
```

```
## Random Forest Recall: 93.07 %
```

```r
cat("Random Forest F1-score:", randomForest_f1_score, "% \n")
```

```
## Random Forest F1-score: 95.67 %
```

Calculate Eligibility Rates - Random Forest

```r
# Assuming `predictions` is a vector of 1s (eligible) and 0s (not eligible) from your model
# For example: predictions <- predict(model, newdata, type = "response") > 0.5

# Count eligible customers
num_eligible_customers <- sum(randomForest_predictions == 1)

# Total number of customers
total_customers <- length(randomForest_predictions)

# Calculate the eligibility percentage
eligibility_percentage <- (num_eligible_customers / total_customers) * 100

cat("Percentage of eligible customers:", round(eligibility_percentage, 2), "%\n")
```

```
## Percentage of eligible customers: 66.36 %
```

Save the random forest model

```r
saveRDS(randomForest_model, file = "random_forest_model.rds")
```