Sport betting

Bobga-Herman Gwanvoma

2025-04-30

```r
data <- read.csv("C:/Users/bobi/Documents/DSC 680/Proj2/data_v1.csv")
```

```r
# Data Preprocessing
# Convert Date to Date format
data$Date <- as.Date(data$Date)

# Feature engineering: Create Goal Difference and Win indicators
data$GoalDifference <- data$hgoal - data$vgoal
data$HomeWin <- ifelse(data$hgoal > data$vgoal, 1, 0)  # Home team win (1 if true, 0 if false)
# Convert HomeWin to numeric to calculate the mean
data$HomeWin <- as.numeric(as.character(data$HomeWin))
data$VisitorWin <- ifelse(data$vgoal > data$hgoal, 1, 0)  # Visitor team win (1 if true, 0 if false)
```

```r
# Ensure HomeWin is a factor with levels 0 and 1 in the original dataset
data$HomeWin <- factor(data$HomeWin, levels = c(0, 1))

# Split data into training and testing sets (80% train, 20% test)
set.seed(42)
trainIndex <- createDataPartition(data$HomeWin, p = 0.8, list = FALSE)
trainData <- data[trainIndex, ]
testData <- data[-trainIndex, ]

# Model Building: Random Forest
rf_model <- randomForest(HomeWin ~ GoalDifference, data = trainData, ntree = 100)

# Ensure the HomeWin column in testData is also a factor with levels 0 and 1
testData$HomeWin <- factor(testData$HomeWin, levels = c(0, 1))

# Make predictions and ensure they are factors with the same levels as the actual data
rf_predictions <- predict(rf_model, testData)
rf_predictions <- factor(rf_predictions, levels = c(0, 1))  # Ensuring the same factor levels

# Model Evaluation on Test Data using confusionMatrix
conf_matrix <- confusionMatrix(rf_predictions, testData$HomeWin)

# Print the confusion matrix and evaluation metrics
print(conf_matrix)
```

```
## Confusion Matrix and Statistics
##
##           Reference
```

```
## Prediction   0   1
##          0 428   0
##          1   0 365
##
##                Accuracy : 1
##                  95% CI : (0.9954, 1)
##     No Information Rate : 0.5397
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5397
##          Detection Rate : 0.5397
##    Detection Prevalence : 0.5397
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

```r
# Accuracy, Precision, Recall, F1-Score
accuracy <- conf_matrix$overall["Accuracy"]
precision <- conf_matrix$byClass["Pos Pred Value"]
recall <- conf_matrix$byClass["Sensitivity"]
f1_score <- 2 * ((precision * recall) / (precision + recall))

cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 1
```

```r
cat("Precision:", precision, "\n")
```

```
## Precision: 1
```

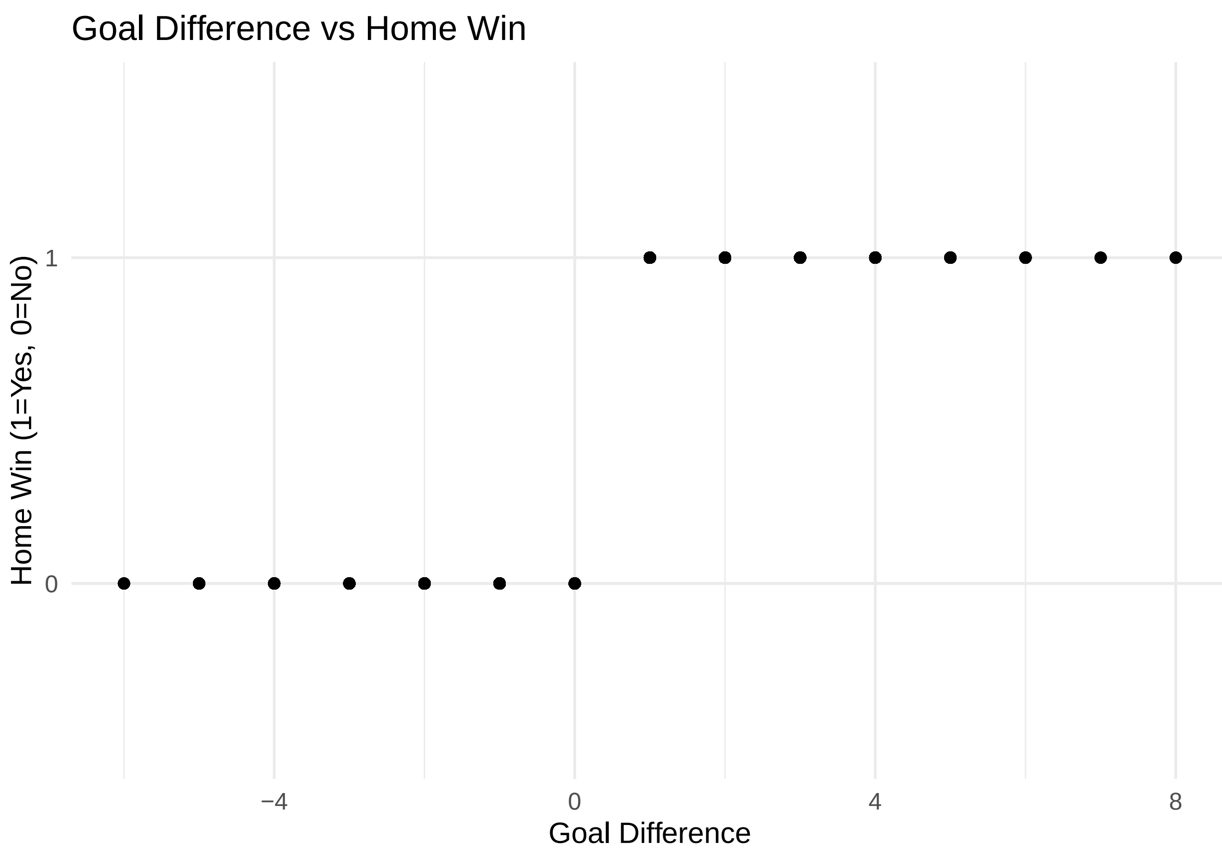
```r
cat("Recall:", recall, "\n")
```

```
## Recall: 1
```

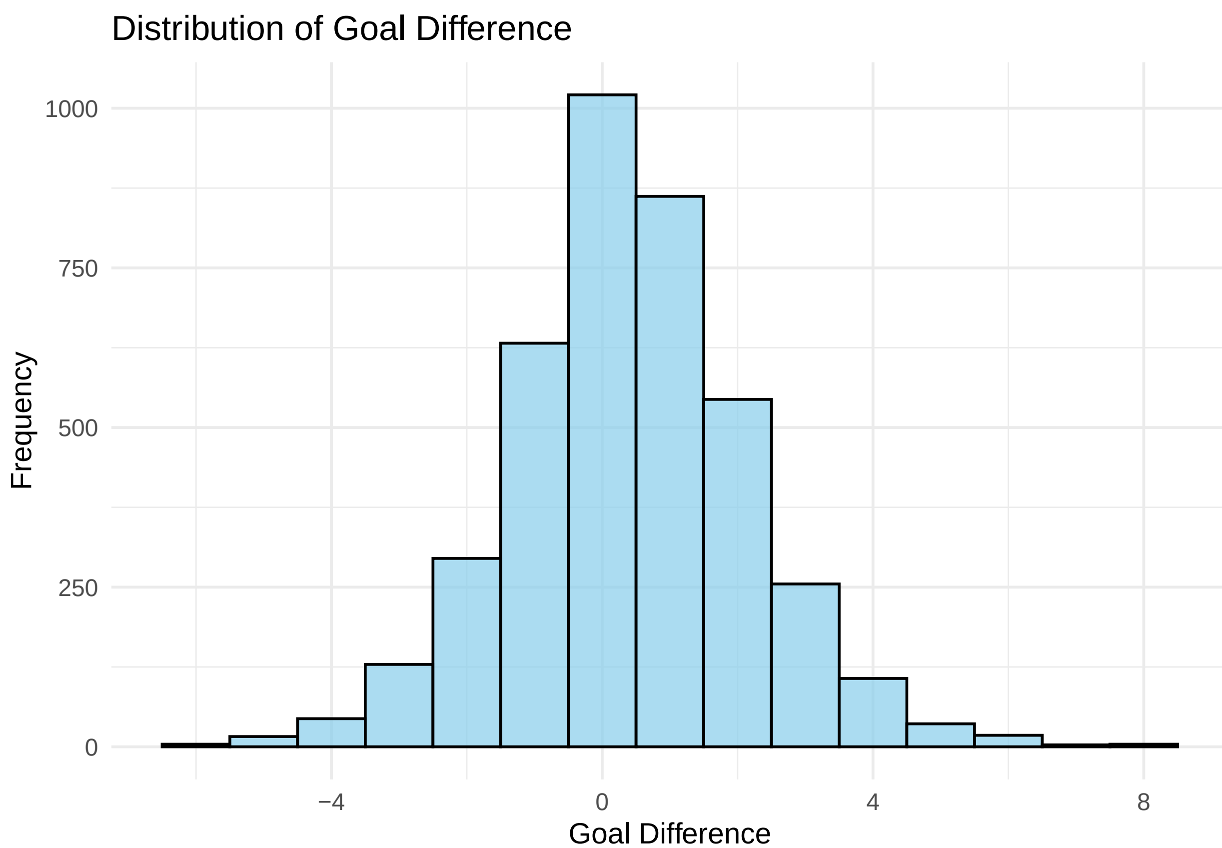
```r
cat("F1-Score:", f1_score, "\n")
```

```
## F1-Score: 1
```

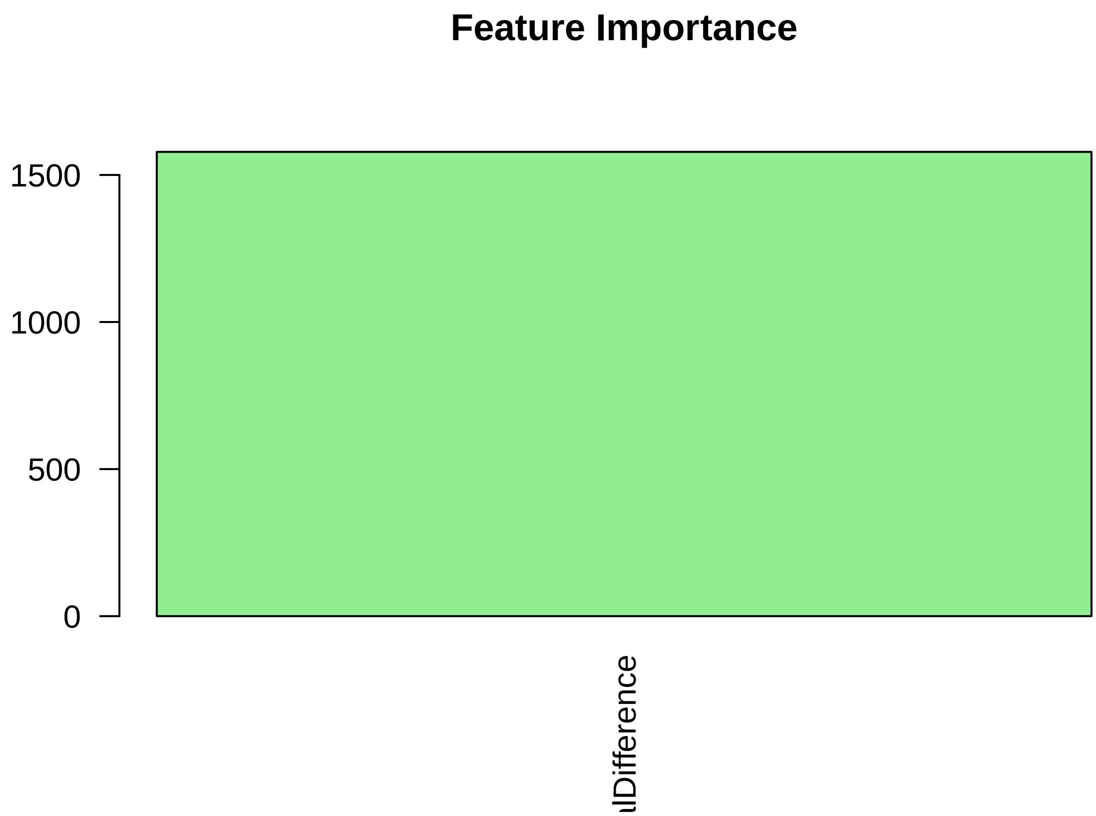
```r
# Illustration 1: Scatter plot of GoalDifference vs HomeWin
ggplot(data, aes(x = GoalDifference, y = HomeWin)) +
  geom_point() +
  labs(title = "Goal Difference vs Home Win", x = "Goal Difference", y = "Home Win (1=Yes, 0=No)") +
  theme_minimal()
```



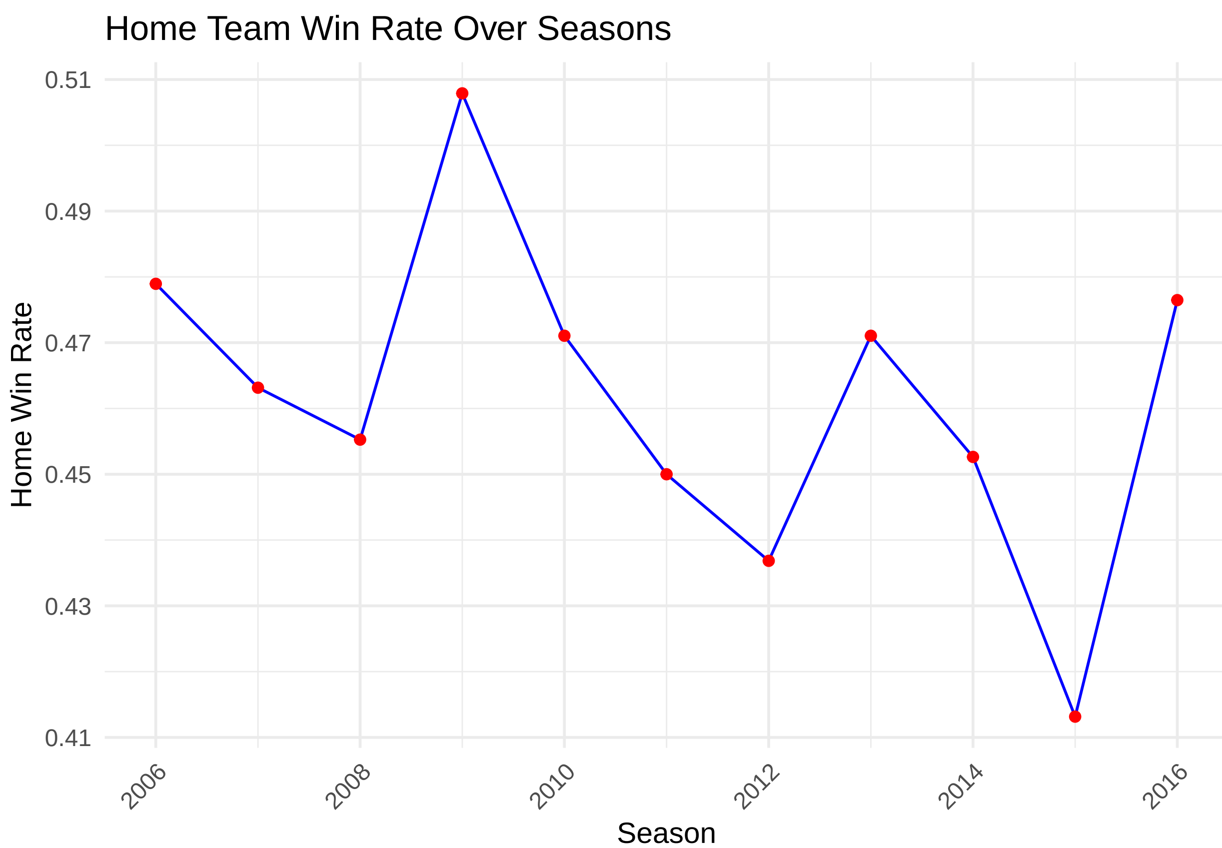

```r
# Illustration 2: Histogram of Goal Difference distribution
ggplot(data, aes(x = GoalDifference)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Goal Difference", x = "Goal Difference", y = "Frequency") +
  theme_minimal()
```



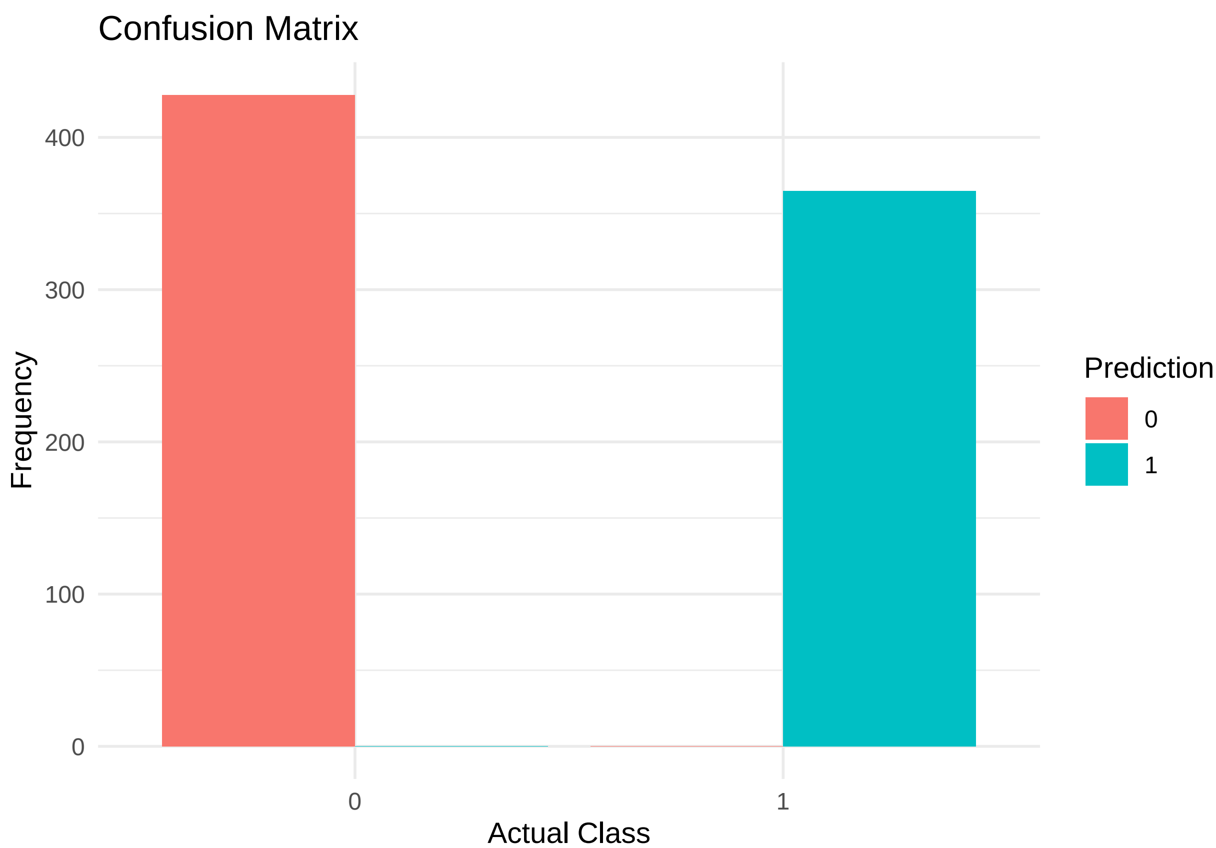

```
# Illustration 3: Feature Importance Plot from Random Forest
importance_plot <- randomForest::importance(rf_model)
importance_plot_df <- as.data.frame(importance_plot)  # Convert to data frame for easy plotting

# Plot Feature Importance
barplot(importance_plot_df$MeanDecreaseGini,
        names.arg = rownames(importance_plot_df),
        main = "Feature Importance",
        col = "lightgreen",
        las = 2,
        ylim = c(0, max(importance_plot_df$MeanDecreaseGini) * 1.1))  # Add some space for the bars
```




```
# Illustration 4: Time Series Plot of Home Win Percentage Over Seasons
# Convert HomeWin to numeric to calculate the mean
data$HomeWin <- as.numeric(as.character(data$HomeWin))

# Calculate the win rate per season
seasonal_home_win_rate <- data %>%
  group_by(Season) %>%
  summarise(home_win_rate = mean(HomeWin, na.rm = TRUE))  # Use na.rm = TRUE to handle any NA values

# Plot the home win rate over seasons
ggplot(seasonal_home_win_rate, aes(x = Season, y = home_win_rate)) +
  geom_line(color = "blue") +
  geom_point(color = "red") +  # Adding points to make the trend clearer
  labs(title = "Home Team Win Rate Over Seasons", x = "Season", y = "Home Win Rate") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotating season labels for better readabi
```




```r
# Illustration 5: Confusion Matrix as a bar plot (True Positives, False Positives, etc.)
cm_values <- as.data.frame(conf_matrix$table)
# Plot the confusion matrix as a bar plot
ggplot(cm_values, aes(x = Reference, y = Freq, fill = Prediction)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Confusion Matrix", x = "Actual Class", y = "Frequency") +
  theme_minimal()
```


Confusion Matrix
Frequency
400
300
200
100
0
0
1
Actual Class
Prediction
0
1