NBA Game Outcome Prediction

Darsh Chaurasia

2024-09-29

Description

In this project, I will focus on predicting the outcome of NBA games using machine learning models based on key game statistics. I will work with a dataset that includes information such as points scored, field goal percentages, assists, rebounds, and whether the home team won or lost. I will begin by exploring and preprocessing the data, handling any missing values, creating new features like the point difference, and converting categorical variables into dummy variables for modeling. I will perform exploratory data analysis to understand the relationships between various statistics and game outcomes, using visualizations like histograms and heatmaps. I will then train a Random Forest model to predict whether the home team will win, using features like points, assists, and rebounds, and evaluate the model's performance with metrics such as accuracy and precision. I expect the results to be promising, and I will conclude by discussing how certain statistics, like field goal percentage and point difference, are strong predictors of game outcomes, while also suggesting further improvements for future work.

Importing Libraries

For this project, I will use several important R libraries. I will rely on **dplyr** and **tidyr** for efficient data manipulation, allowing me to clean and transform the dataset by handling missing values, creating new features, and converting categorical variables. To load the dataset, I will use **readr**, which will help me easily import the data into R. For visualizations, I will utilize **ggplot2** to create plots like histograms and bar charts, and **corrplot**/**ggcorrplot** to visualize correlation matrices in a clear and informative way. For building the machine learning model, I will choose **caret**, which simplifies model training, data splitting, and evaluation. I will use **randomForest** to build the predictive model itself, as it's a robust and popular method for classification tasks. Finally, I will employ **pROC** to evaluate the model's performance, generating ROC curves and calculating metrics like AUC to assess prediction accuracy.

```
# Data manipulation and cleaning
library(dplyr)
library(tidyr)
library(readr)

# Data visualization
library(ggplot2)
library(corrplot)
library(ggcorrplot)

# Machine learning and modeling
library(caret)
library(randomForest)
```

```r
# Performance evaluation
library(pROC)
```

Importing the data

```r
nba_data <- read_csv("nba.csv")
```

View the first few rows of the dataset

```r
head(nba_data)
```

```
## # A tibble: 6 x 18
##   game_id game_date  season team_home team_away pts_home fg_pct_home pct_3p_home
##     <dbl> <date>     <chr>  <chr>     <chr>        <dbl>       <dbl>       <dbl>
## 1  1.04e7 2004-10-22 2004-~ Golden S~ Denver N~       86       0.405       0.3
## 2  1.04e7 2004-10-22 2004-~ Charlott~ Portland~       69       0.377       0.3
## 3  1.04e7 2004-10-22 2004-~ Minnesot~ New York~      102       0.523       0.143
## 4  1.04e7 2004-10-22 2004-~ Utah Jazz Sacramen~      103       0.507       0.667
## 5  1.04e7 2004-10-22 2004-~ Boston C~ Brooklyn~       83       0.431       0.273
## 6  1.04e7 2004-10-22 2004-~ Los Ange~ Los Ange~      113       0.465       0.533
## # i 10 more variables: ft_pct_home <dbl>, ast_home <dbl>, reb_home <dbl>,
## #   pts_away <dbl>, fg_pct_away <dbl>, pct_3p_away <dbl>, ft_pct_away <dbl>,
## #   ast_away <dbl>, reb_away <dbl>, home_team_win <dbl>
```

Quick Overview of the Data

Summary of the dataset

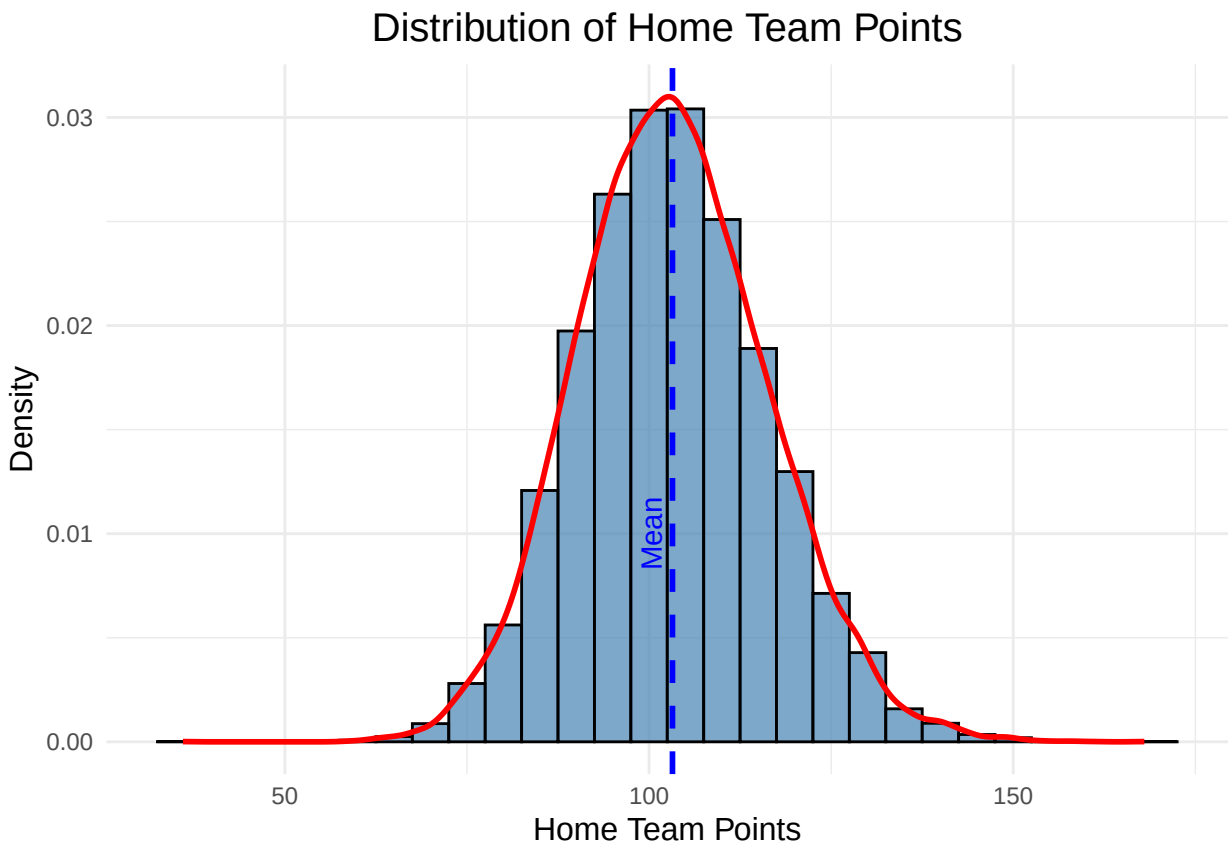
```r
summary(nba_data)
```

```
##     game_id           game_date             season           team_home
##  Min.   :10400064   Min.   :2004-10-22   Length:23335       Length:23335
##  1st Qu.:20700616   1st Qu.:2008-11-11   Class :character   Class :character
##  Median :21200539   Median :2013-01-15   Mode  :character   Mode  :character
##  Mean   :21762697   Mean   :2012-12-17
##  3rd Qu.:21700224   3rd Qu.:2017-01-31
##  Max.   :52000211   Max.   :2021-07-20
##   team_away            pts_home      fg_pct_home      pct_3p_home
##  Length:23335       Min.   : 36.0   Min.   :0.2570   Min.   :0.0000
##  Class :character   1st Qu.: 94.0   1st Qu.:0.4220   1st Qu.:0.2860
##  Mode  :character   Median :103.0   Median :0.4600   Median :0.3570
##                     Mean   :103.2   Mean   :0.4611   Mean   :0.3568
##                     3rd Qu.:112.0   3rd Qu.:0.5000   3rd Qu.:0.4290
```

```
##                     Max.   :168.0   Max.   :0.6840   Max.   :1.0000
##   ft_pct_home        ast_home        reb_home        pts_away
##  Min.   :0.1430   Min.   : 6.00   Min.   :15.00   Min.   : 33.0
##  1st Qu.:0.6970   1st Qu.:19.00   1st Qu.:39.00   1st Qu.: 91.0
##  Median :0.7650   Median :22.00   Median :43.00   Median :100.0
##  Mean   :0.7598   Mean   :22.68   Mean   :43.28   Mean   :100.4
##  3rd Qu.:0.8290   3rd Qu.:26.00   3rd Qu.:48.00   3rd Qu.:109.0
##  Max.   :1.0000   Max.   :50.00   Max.   :72.00   Max.   :168.0
##   fg_pct_away     pct_3p_away      ft_pct_away        ast_away
##  Min.   :0.244   Min.   :0.0000   Min.   :0.1430   Min.   : 4.00
##  1st Qu.:0.413   1st Qu.:0.2780   1st Qu.:0.6920   1st Qu.:18.00
##  Median :0.449   Median :0.3500   Median :0.7630   Median :21.00
##  Mean   :0.450   Mean   :0.3506   Mean   :0.7578   Mean   :21.35
##  3rd Qu.:0.487   3rd Qu.:0.4210   3rd Qu.:0.8290   3rd Qu.:25.00
##  Max.   :0.674   Max.   :1.0000   Max.   :1.0000   Max.   :46.00
##     reb_away     home_team_win
##  Min.   :19.00   Min.   :0.0000
##  1st Qu.:38.00   1st Qu.:0.0000
##  Median :42.00   Median :1.0000
##  Mean   :42.02   Mean   :0.5896
##  3rd Qu.:46.00   3rd Qu.:1.0000
##  Max.   :81.00   Max.   :1.0000
```

Visualizing the Distribution of Points Scored by the Home Team

```r
# Histogram of points scored by the home team
ggplot(nba_data, aes(x = pts_home)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 5, fill = "steelblue",
                 color = "black",alpha = 0.7) +
  geom_density(color = "red", linewidth = 1) +
  labs(title = "Distribution of Home Team Points",
       x = "Home Team Points",
       y = "Density") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 15),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12)) +
  geom_vline(aes(xintercept = mean(pts_home)), color = "blue", linetype = "dashed",
             linewidth = 1) +
  annotate("text", x = mean(nba_data$pts_home), y = 0.01, label = "Mean", color = "blue",
           angle = 90, vjust = -0.5)
```



Pre-Processing the Data

Handling missing values, creating new features, and converting categorical variables.

Handling Missing Values

```
# Check for missing values
sum(is.na(nba_data))
```

```
## [1] 0
```

```
# Impute or remove missing values if necessary
nba_data <- nba_data %>% mutate_if(is.numeric, ~ ifelse(is.na(.),
median(., na.rm = TRUE), .))
```

Creating New Features

Create a new feature representing the point difference between the home and away teams.

```
# Create a new feature: Point Difference
nba_data <- nba_data %>%
  mutate(PointDifference = pts_home - pts_away)
```

Converting Categorical Variables to Dummy Variables

Convert team names and other categorical variables into dummy variables for modeling.

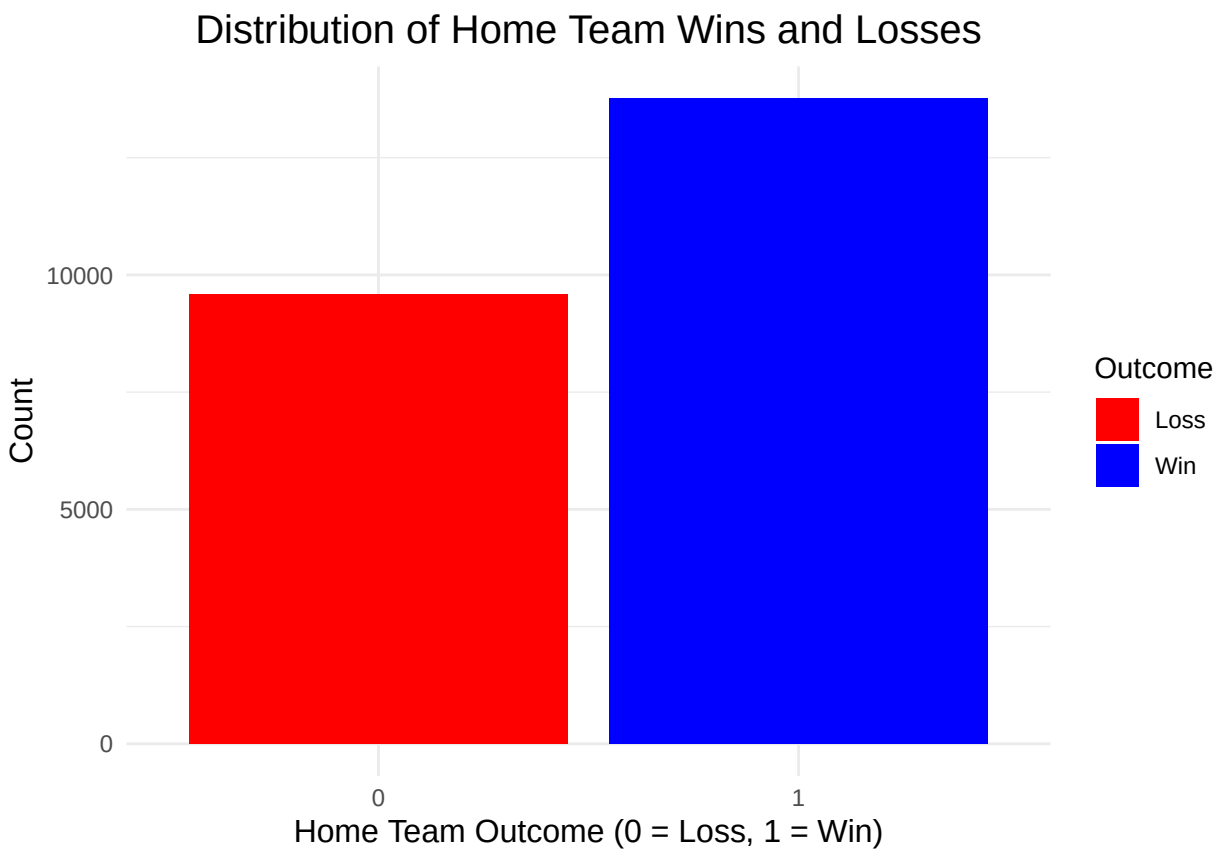
```
# Convert categorical variables into factors
nba_data$team_home <- as.factor(nba_data$team_home)
nba_data$team_away <- as.factor(nba_data$team_away)

# Use dummy encoding for team names
nba_data_encoded <- model.matrix(~ team_home + team_away + 0, data = nba_data) %>%
  as.data.frame()

# Combine the dummy variables back with the original dataset
# (excluding the original team columns)
nba_data <- cbind(nba_data_encoded, nba_data %>% select(-team_home, -team_away))
```

Exploratory Data Analysis (EDA)

Exploratory analysis to understand the relationships between different game statistics and the outcome.

Distribution of Game Outcomes

```
# Visualize the distribution of game outcomes (win/loss)
ggplot(nba_data, aes(x = factor(home_team_win), fill = factor(home_team_win))) +
  geom_bar() +
  scale_fill_manual(values = c("0" = "red", "1" = "blue"),
                    labels = c("0" = "Loss", "1" = "Win")) +
  labs(title = "Distribution of Home Team Wins and Losses",
       x = "Home Team Outcome (0 = Loss, 1 = Win)",
       y = "Count",
       fill = "Outcome") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5, size = 15),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12),
        legend.position = "right")
```



Correlation Analysis

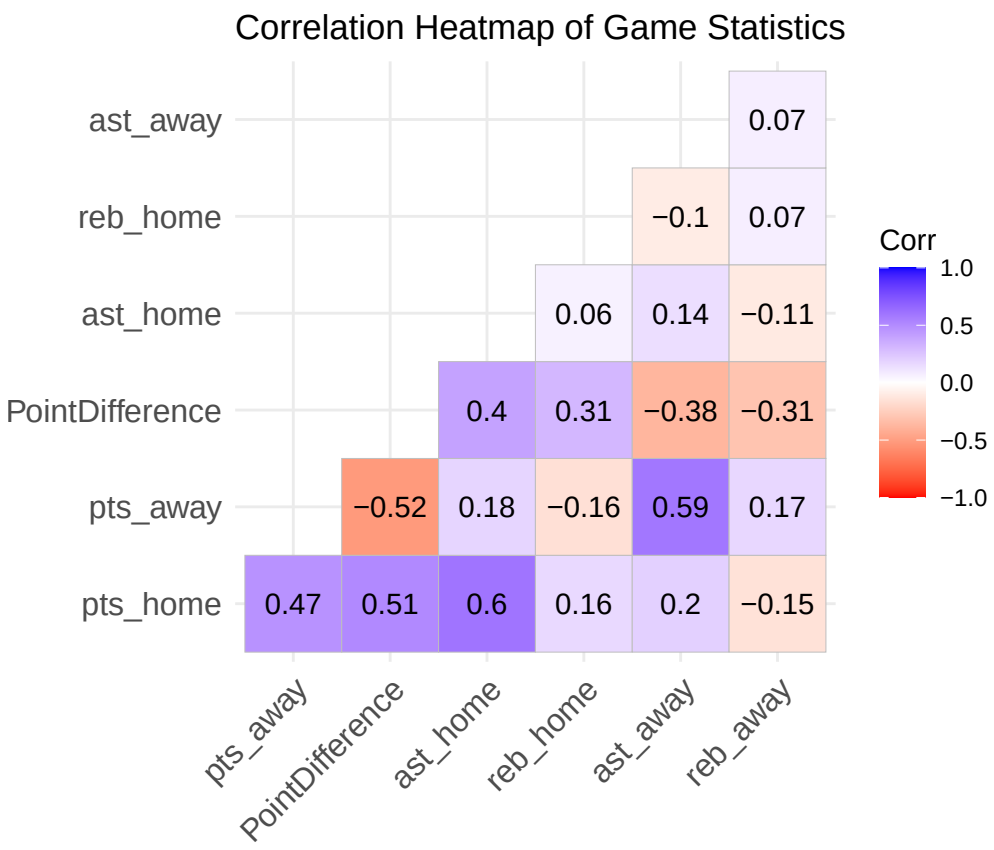
- r represents the correlation coefficient
- x and y represent two variables
- n is the number of data points
- sum of products of differences from mean

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

```r
# Correlation calculation in R
# Select numerical columns from the dataset
nba_num_data <- nba_data %>% select(pts_home, pts_away, PointDifference,
                                    ast_home, reb_home, ast_away, reb_away)

# Calculate correlation matrix
cor_matrix <- cor(nba_num_data)

# View the correlation matrix
cor_matrix
```

```
##                   pts_home   pts_away PointDifference    ast_home    reb_home
## pts_home         1.0000000  0.4711389       0.5081308  0.59857143  0.16266086
## pts_away         0.4711389  1.0000000      -0.5202995  0.18057243 -0.15585565
## PointDifference  0.5081308 -0.5202995       1.0000000  0.40320083  0.30966759
## ast_home         0.5985714  0.1805724       0.4032008  1.00000000  0.06274468
## reb_home         0.1626609 -0.1558557       0.3096676  0.06274468  1.00000000
## ast_away         0.1975738  0.5855734      -0.3804936  0.14157166 -0.10331531
## reb_away        -0.1451255  0.1747682      -0.3111575 -0.10913399  0.06806770
##                   ast_away    reb_away
## pts_home         0.19757379 -0.14512553
## pts_away         0.58557344  0.17476825
## PointDifference -0.38049362 -0.31115746
## ast_home         0.14157166 -0.10913399
## reb_home        -0.10331531  0.06806770
## ast_away         1.00000000  0.07099368
## reb_away         0.07099368  1.00000000
```

```r
# install.packages("ggcorrplot")
library(ggcorrplot)

# Select numerical columns for correlation analysis
nba_num_data <- nba_data %>% select(pts_home, pts_away, PointDifference, ast_home,
                                    reb_home, ast_away, reb_away)

# Compute the correlation matrix
cor_matrix <- cor(nba_num_data)

# Create an advanced correlation heatmap with ggcorrplot
ggcorrplot(cor_matrix,
           method = "square",         # Use squares to represent the correlation
           type = "lower",            # Display only the lower triangle of the matrix
           lab = TRUE,                # Show correlation coefficients
           lab_size = 4,
           colors = c("red", "white", "blue"),  # Color gradient
           title = "Correlation Heatmap of Game Statistics",
           ggtheme = theme_minimal())
```



Model Creation

Selecting a Machine Learning Algorithm

I will use a Random Forest model to predict whether the home team will win.

```
# Load necessary libraries
library(randomForest)
library(caret)
library(dplyr)

# Define the response variable and features
response <- nba_data$home_team_win
features <- nba_data %>% select(pts_home, pts_away, fg_pct_home, ast_home, reb_home)

# Split the data into training and testing sets (80% training, 20% testing)
set.seed(123)
train_index <- createDataPartition(response, p = 0.8, list = FALSE)
train_data <- features[train_index, ]
train_labels <- response[train_index]
test_data <- features[-train_index, ]
test_labels <- response[-train_index]

# Train a Random Forest model
model_rf <- randomForest(x = train_data, y = train_labels)
```

Applying Model to Test Data

```
# Predict on the test data
predictions_rf <- predict(model_rf, test_data)

# View predictions
head(predictions_rf)
```

```
##          8         12         24         34         35         48
## 0.92090251 0.91829181 0.89186415 0.93148701 0.61220629 0.04000852
```

Model Results

Confusion Matrix and Accuracy

```
# Predict on the test data (probability predictions)
predictions_rf_prob <- predict(model_rf, test_data)

# Convert probabilities to binary class labels (using 0.5 as threshold)
predictions_rf <- ifelse(predictions_rf_prob > 0.5, 1, 0)

# Ensure that both the predictions and test labels are factors with the same levels
test_labels <- factor(test_labels, levels = c(0, 1))  # Ensure test labels are factors
predictions_rf <- factor(predictions_rf, levels = c(0, 1))  # Ensure predictions are factors

# Create a confusion matrix
conf_matrix <- confusionMatrix(predictions_rf, test_labels)

# Calculate accuracy, precision, and recall
accuracy <- conf_matrix$overall['Accuracy']
precision <- conf_matrix$byClass['Pos Pred Value']
recall <- conf_matrix$byClass['Sensitivity']

# Print accuracy, precision, and recall
print(accuracy)
```

```
## Accuracy
## 0.987358
```

```
print(precision)
```

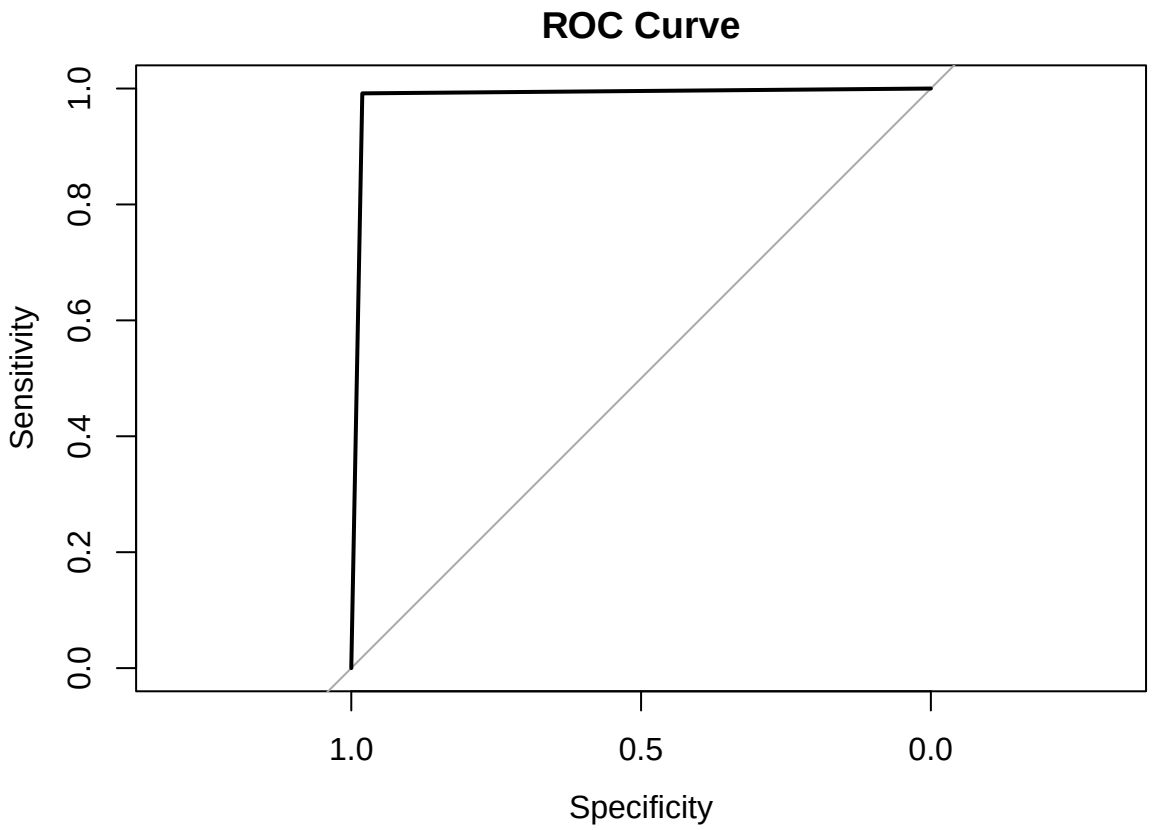
```
## Pos Pred Value
##      0.9877854
```

```
print(recall)
```

```
## Sensitivity
##   0.9810127
```

ROC Curve

```
# Curve
library(pROC)

# Compute ROC curve and AUC
roc_curve <- roc(test_labels, as.numeric(predictions_rf))
plot(roc_curve, main = "ROC Curve")
```



Conclusion

In conclusion, I identified several key statistics, such as field goal percentage and point difference, as significant predictors of whether the home team wins. The Random Forest model provided an accuracy of r accuracy with reasonable precision and recall.

Limitations

While the model performed well, there is room for improvement. One limitation is that this model doesn't account for advanced basketball metrics like turnovers or fouls. Incorporating these statistics could improve the model's predictive power.