PRESIDENCY UNIVERSITY

Private University Estd. in Karnataka State by Act No. 41 of 2013

BANGALORE

School of Engineering

R-Programming for Data Science – CSE3035

A Project Report on

Predicting Shot Success in NBA Games Using Decision Tree and Random Forest Algorithm

Presented by:

Darshan Gowda S

20221IST0055

Presented to

Dr.A.K.Sampath

Associate Professor, School of Engineering

Table of Contents

CERTIFICATE

This is to certify that the Mini Project titled **"Predicting Shot Success in NBA Games Using Decision Tree and Random Forest Algorithm"** is a bona fide work of **Darshan Gowda S (20221IST0055)** submitted to Presidency University in partial fulfillment of the requirements for the degree of **Bachelor of Technology in Information Science and Technology**.

(DR. SAMPATH A K)
SUBJECT INCHARGE
R PROGRAMMING IN DATA SCIENCE

Abstract

Basketball analytics has emerged as a cornerstone of modern sports, empowering teams and players to achieve better performance through data-driven decision-making. The ability to predict shot success is a critical component of this field, offering insights into player efficiency, team strategies, and game outcomes. This project leverages the power of machine learning to develop predictive models that analyze key factors influencing shot success in NBA games.

Two widely recognized algorithms—Decision Trees and Random Forests—were employed to analyze features such as shot distance, defender proximity, shot clock, and game clock. Data preprocessing and exploratory data analysis (EDA) were conducted to clean and structure the dataset, ensuring the accuracy and reliability of the models.

Random Forest emerged as the superior model, outperforming Decision Trees in terms of accuracy, sensitivity, and specificity. Furthermore, feature importance analysis provided actionable insights into the relative impact of various factors, highlighting shot distance and defender proximity as the most significant determinants of shot success.

This study underscores the growing importance of machine learning in sports analytics, demonstrating its potential to refine game strategies, improve player training, and enhance tactical decision-making during games. The findings pave the way for further research into real-time predictive systems and advanced models capable of integrating additional player and game-specific factors.

Keywords: Machine Learning, Basketball Analytics, Random Forest, Decision Tree, Sports Data, Predictive Modeling, Feature Importance.

Acknowledgement

I would like to begin by expressing my deepest gratitude to the **Almighty** for blessing me with the strength, perseverance, and wisdom to complete this project successfully.

I extend my heartfelt appreciation to **Dr. Sampath A. K**, Associate Professor at Presidency University, Bengaluru, for his invaluable guidance, encouragement, and support throughout the course of this project. His expertise and insights were instrumental in shaping the direction of this study and ensuring its successful completion.

I am also grateful to the **Department of Information Science and Technology**, Presidency University, for providing a conducive environment and the necessary resources to carry out this research.

Additionally, I would like to thank my friends and peers for their continuous support and constructive feedback, which motivated me to strive for excellence.

Finally, I am deeply indebted to my family for their unwavering encouragement, patience, and understanding throughout this journey.

Darshan Gowda S

20221IST0055

Presidency University, Bengaluru

List of Abbreviations

1. **AI**: Artificial Intelligence
2. **EDA**: Exploratory Data Analysis
3. **FPR**: False Positive Rate
4. **MAE**: Mean Absolute Error
5. **ML**: Machine Learning
6. **RF**: Random Forest
7. **ROC**: Receiver Operating Characteristic
8. **RMSE**: Root Mean Squared Error
9. **SVM**: Support Vector Machine
10. **TPR**: True Positive Rate.

List of Figures

List of Tables

List of Symbols

1. **SHOT_DIST**: Shot distance (in feet or meters)
2. **CLOSE_DEF_DIST**: Distance from the closest defender (in feet or meters)
3. **SHOT_CLOCK**: Time remaining on the shot clock when the shot was attempted (in seconds)
4. **GAME_CLOCK**: Time remaining in the game quarter (in seconds)
5. **SUCCESS (SHOT_RESULT)**: Binary outcome (TRUE if shot made, FALSE if missed)
6. **Accuracy**: Proportion of correct predictions (correctly classified shots)
7. **Sensitivity**: Proportion of true positives (correctly predicted successful shots)
8. **Specificity**: Proportion of true negatives (correctly predicted missed shots)

1. Introduction

1.1 Introduction

Basketball, you know, it's one of those sports that really grabs people's attention all around the world. It's fast-paced, exciting, and demands quick thinking. Players and coaches are constantly assessing performance and tweaking strategies, both during the heat of the game and in practice. One crucial aspect? Predicting shot success. It's a big deal because it can really sway the outcome of a match and even how effective a player is overall.

Thanks to advances in data science and machine learning, we can now sift through tons of game data—making it a whole lot easier to fine-tune strategies. By looking at factors like how far the shot is, how close the defender is, and what's happening on the game clock, we can develop predictive models that help spot patterns that might have been overlooked before.

In this project, we're diving into two popular machine learning models: Decision Trees and Random Forests. Decision Trees are great because they give clear-cut rules for decision-making. Random Forests? They take it a step further by improving accuracy through a mix of different decision trees. Our goal here is to dig into how these key features relate to whether a shot goes in or not, offering insights that can help coaches, players, and analysts make better choices.

1.2 Motivation

So, why are we doing this? Well, basketball analytics has really started to lean into predictive modeling to get that edge over opponents. In a sport that moves as fast as basketball, being able to predict shot success in the moment can change the game. There are three main reasons driving this study:

- **Designing Effective Plays**: Coaches can use these predictive insights to craft offensive and defensive plays that really aim to boost shot success.
- **Improving Player Performance**: Players can pinpoint their weaknesses, like maybe they struggle with long shots or defending against tight coverage, and then focus on those areas for improvement.
- **Enhancing Fan Engagement**: By using advanced metrics, fans can get a better grasp of the strategies at play and player contributions, making the whole experience more engaging.

The relationship between things like shot distance, defender pressure, and the game clock makes predicting shot success a bit tricky but also pretty exciting. Tackling these complexities? That's how we're contributing to the expanding field of basketball analytics.

1.3 Problem Statement and Objectives

Problem Statement

Now, predicting whether a shot will be successful in basketball isn't a walk in the park—it's tough because the game is so dynamic. Each shot can be affected by a bunch of different things, including:

- **Defender Proximity**: The closer the defender gets, the harder it is to make that shot.
- **Shot Distance**: Longer shots? Yeah, those often lead to more misses.
- **Game Context**: Factors like how much time is left on the clock can ramp up the pressure on players.

This study is set to tackle these challenges head-on, using machine learning models to make accurate predictions about shot success.

Objectives

To hit our goals, we've laid out a few key objectives:

- **Preprocess and Clean the Dataset**: We need to make sure our data is solid—fixing missing values and normalizing features is key.
- **Conduct Exploratory Data Analysis (EDA)**: We'll visualize how different features relate to each other—like how shot distance affects success rates—to uncover trends and insights.
- **Train and Evaluate Models**: We'll develop our Decision Tree and Random Forest models, checking their performance with metrics such as accuracy, sensitivity, and specificity.

- **Compare Model Performance**: Here, we'll shine a light on the strengths and weaknesses of each model when it comes to predicting shot success.
- **Analyze Feature Importance**: This involves pinpointing which factors—like shot distance or defender proximity—are the biggest players in whether shots succeed or fail.

1.4 Organization of the Report

This report is laid out to give a clear picture of our study, the methods we used, and what we found. Here's how it's organized:

Literature Survey:
In this section, we'll look at what's already been done in terms of basketball shot prediction—what works, what doesn't, and how our project builds on previous research.

Proposed System:
We'll dive into the nitty-gritty of our methodology, covering data preprocessing, exploratory data analysis (EDA), model development, and evaluation. Plus, we'll give a detailed rundown of the dataset and the machine learning models we used.

Conclusion and Future Work:
This part sums up the main findings of our project, stressing the insights we gained from our models. We'll also touch on where future research could go, like adding more features or creating systems for real-time predictions

2. Literature Survey

2.1 Survey of Existing Systems

Predictive modeling in sports analytics has evolved significantly, incorporating a variety of methodologies. These methods are instrumental in extracting actionable insights from data and predicting outcomes like player performance, team success, or shot accuracy. Below are the most prominent approaches used:

1. **Statistical Models:**
    - Statistical models form the foundation of sports analytics. They are widely appreciated for their interpretability, allowing coaches and analysts to understand how specific variables influence outcomes.
    - Common techniques include linear regression and basic probability models, which analyze trends like scoring averages or shooting percentages.
    - **Limitations:** These models often struggle with non-linear relationships, such as the interaction between defender proximity and shot distance, limiting their predictive capabilities in dynamic environments like basketball.
2. **Machine Learning Models:**
    - The advent of machine learning has introduced powerful tools like Decision Trees, Random Forests, and logistic regression. These models excel at handling structured data and can identify complex patterns and relationships.
    - **Decision Trees**: Provide interpretable decision-making rules, making them useful for understanding feature importance.

- **Random Forests**: Improve accuracy and robustness by aggregating multiple Decision Trees.
- **Logistic Regression**: Widely used for binary classification tasks such as predicting shot success (made/missed).
- **Advantages:** Machine learning models can effectively handle a mix of numerical and categorical variables, making them ideal for sports data.

3. **Deep Learning Techniques:**
   - Neural networks and deep learning models have shown remarkable performance in analyzing complex, unstructured data, such as video footage or player movements.
   - **Strengths:** Deep learning models can capture intricate, non-linear relationships, making them highly accurate for tasks like play recognition or advanced player tracking.
   - **Challenges:** These models require extensive datasets and high computational resources, making them less feasible for smaller-scale applications like this project.

While these methodologies have made significant contributions to sports analytics, they each come with inherent limitations that this project seeks to address.

2.2 Limitations of Existing Systems

Despite the advancements in predictive modeling, existing systems face several key challenges:

1. **Data Quality Issues:**
    - Sports datasets often suffer from missing, inconsistent, or noisy data. For example, missing values in critical fields like defender distance or shot clock can drastically reduce the reliability of predictions.
    - Poor data quality necessitates extensive preprocessing, which, if inadequately addressed, can result in inaccurate or biased models.
2. **Complexity of Models:**
    - While advanced models like neural networks offer high accuracy, they often lack interpretability. Coaches and analysts may struggle to translate the results into actionable strategies, reducing their practical applicability.
    - For example, while a neural network might predict shot success with high accuracy, it may not clearly explain *why* a specific shot was predicted to succeed or fail.
3. **Generalization Challenges:**
    - Many models, particularly complex ones, are prone to overfitting. This occurs when a model performs exceptionally well on training data but fails to generalize to new, unseen scenarios.
    - Overfitting is a significant concern in basketball analytics, where game contexts and player behaviors vary widely.

Addressing these limitations is critical for developing predictive models that are both accurate and practical.

2.3 Mini Project Contribution

This project aims to overcome the challenges identified in existing systems through the following contributions:

1. **Robust Data Preprocessing:**
    - The dataset is cleaned and normalized to ensure consistent and high-quality inputs.
    - Techniques such as handling missing values, unit conversions (e.g., feet to meters), and time transformations (e.g., converting game clock to seconds) are implemented.
2. **Development of Interpretable and Scalable Models:**
    - Two machine learning models, Decision Trees and Random Forests, are used in this project. These models strike a balance between accuracy and interpretability, allowing stakeholders to derive meaningful insights.
    - Decision Trees provide clear, understandable decision rules, while Random Forests enhance performance by aggregating multiple trees.
3. **Actionable Insights Through Feature Importance Analysis:**
    - By analyzing feature importance, the project identifies the factors most critical to predicting shot success.
    - For example, features like shot distance, defender proximity, and game clock context are evaluated to determine their relative impact.
    - These insights can be directly applied to improve game strategies, train players, and inform real-time decision-making during matches.

3. Proposed System

3.1 Introduction

The idea here is to create a system that can predict whether a basketball shot will be successful or not, using machine learning models. We're looking at specific game features—things like how far the shot is, how close the defender is, and how many seconds are left on the clock. By analyzing all this info, the system can give us pretty accurate predictions and some useful insights. You know, basketball is one of those games where every single second really matters. Each decision made on the court can totally change the flow of the game. If we can predict if a shot is likely to go in, it could seriously help coaches make smarter tactical choices, assist players in improving their game, and give analysts a better understanding of what's happening on the court. Now, about the system itself—it uses a couple of machine learning algorithms**: Decision Trees and Random Forests.** Decision Trees are great because they provide clear rules, which helps in figuring out how different features relate to one another. On the other hand, Random Forests boost accuracy by combining several trees, which helps to avoid over fitting. So, when you put these two together, you get a solid mix of usability and performance.

3.2 Architecture/Framework:

The deal with the system architecture. It's got five key stages, and let me tell you, each one is super important if we want to make solid and dependable predictions.

Data Collection: First off, we're using the NBA Shot Logs dataset as our main source of data. This thing is packed with all sorts of details about player actions during games—think shot distances, how close defenders are, and whether the shots actually went in or not.

Data Preprocessing: Next up, we've got data preprocessing. This part is all about making sure the data is up to snuff. We deal with any missing values, normalize those numerical features, and convert any time-related info into a format that's easier to work with.

Exploratory Data Analysis (EDA): After that, we dive into Exploratory Data Analysis, or EDA for short. This is where the fun begins! We whip up some visualizations and run statistical analyses to find patterns in the data. For instance, we look at how shot success changes with distance or when defenders are breathing down a player's neck.

Model Development: Then we move on to Model Development. Here, we train Decision Trees and Random Forest models using the cleaned-up data to help us predict whether a shot will be successful or not.

Evaluation:Finally, we evaluate how well our models are doing. We use metrics like accuracy, sensitivity, and specificity to see how they stack up against each other. This makes it pretty clear which model is performing better.So, in a nutshell, this whole architecture gives us a structured way to build a reliable and understandable system for basketball analytics. It's a process, but it's all about getting those insights.

3.3 Algorithm and Process Design

The proposed system follows a structured algorithmic design, encompassing data preprocessing, visualization, model training, and evaluation.

Steps in Algorithm Design:

1. **Import Libraries:**
   Essential libraries such as **dplyr, ggplot2, rpart, and randomForest** are loaded to handle data manipulation, visualization, and machine learning tasks.
2. **Data Preprocessing:**
   - Missing data is removed from critical features like **SHOT_CLOCK** and **defender_dist**.
   - Units for shot distance and defender proximity are converted from feet to meters.
   - Time features such as **GAME_CLOCK** are transformed into numeric formats (seconds).
3. **Exploratory Data Analysis (EDA):**
   - Visualizations are created to explore relationships between features and shot success. Examples include:
     - Histograms of shot distances to observe success rates.
     - Scatter plots of defender proximity vs. shot success.

4. **Train Models:**
    - **Decision Tree:**
      A simple yet interpretable model that splits data into branches based on feature thresholds.

```
tree_model <- rpart(success ~ ., data = train, method = "class")
```

    - **Random Forest:**
      An ensemble model that combines multiple Decision Trees for improved accuracy and generalization.

```
rf_model <- randomForest(success ~ ., data = train, ntree = 100)
```

5. **Evaluate Models:**
    - Metrics such as accuracy, sensitivity, and specificity are used to evaluate the models. Predictions are generated on test data, and confusion matrices are calculated to assess performance.

3.4 Details of Hardware & Software

Hardware Specifications:

- **Processor:** Intel Core i5 or higher for efficient computations.
- **RAM:** Minimum 8GB (16GB recommended for faster data handling).
- **Storage:** SSD for faster data access and processing.

Software Specifications:

- **Development Environment:** RStudio IDE
- **Programming Language:** R (version 4.0 or later)
- **Libraries Used:**
    - **dplyr and tidyr:** For data preprocessing and manipulation.
    - **ggplot2:** For creating visualizations.
    - **rpart:** For training Decision Trees.
    - **randomForest:** For building Random Forest models.
    - **caret:** For evaluating model performance.
    - **ROCR:** For analyzing ROC curves.

3.5 Dataset Analysis

The **NBA Shot Logs** dataset serves as the core for this study, containing extensive data about player actions during games. It helps in predicting the success or failure of shots based on several key factors.

Key Features:

1. **Shot Distance (SHOT_DIST)**:
   This feature represents the distance from which a player attempts a shot. As the distance increases, the likelihood of making the shot generally decreases. This variable plays a crucial role in determining shot success.
2. **Defender Distance (CLOSE_DEF_DIST)**:
   This measures the distance between the shooter and the closest defender. A higher

defender proximity usually results in lower shot success due to increased defensive pressure.

3. **Game Clock (GAME_CLOCK)**:
   The **GAME_CLOCK** variable indicates the remaining time in the quarter or game, measured in seconds. Time pressure can affect a player's decision-making and shooting accuracy, especially as the game nears critical moments.
4. **Shot Clock (SHOT_CLOCK)**:
   This variable tracks the remaining time on the shot clock during an attempt. Shots made under pressure (with less than 5 seconds remaining) tend to have a lower success rate. It is a vital feature for understanding the urgency of shot attempts.
5. **Shot Result (SHOT_RESULT)**:
   The target variable, which is binary: **1** indicates a successful shot (made), and **0** indicates a missed shot. This is the outcome we aim to predict using machine learning models.

Preprocessing Steps:

- **Handling Missing Values**: Rows with missing values in essential columns (e.g., SHOT_CLOCK, CLOSE_DEF_DIST) were removed to ensure clean data for model training.
- **Unit Conversion**: Distances, initially recorded in feet, were converted to meters for standardization.
- **Time Transformation**: The **GAME_CLOCK** variable was converted into seconds to facilitate more meaningful numerical analysis, especially when paired with the SHOT_CLOCK feature.

3.6 Experiment and Results for Validation and Verification

Two models were tested for predicting shot success: **Decision Tree** and **Random Forest**. Both models were assessed using various metrics, and the results were compared to determine the superior model for this task.

Model Performance Metrics

The table below summarizes the performance of both models on various key metrics:

Metric	Decision Tree	Random Forest
Accuracy	**0.6045 (60.45%)**	**0.5832 (58.32%)**
Sensitivity (True Positive Rate)	**0.8070 (80.70%)**	**0.7146 (71.46%)**
Specificity (True Negative Rate)	**0.3630 (36.30%)**	**0.4266 (42.66%)**
Positive Predictive Value (PPV)	**0.6017 (60.17%)**	**0.5977 (59.77%)**
Negative Predictive Value (NPV)	**0.6120 (61.20%)**	**0.5562 (55.62%)**
Kappa	**0.1758**	**0.1438**

Best Model

1. Accuracy:

- **Decision Tree**: 60.45%
- **Random Forest**: 58.32%

Accuracy is the overall percentage of correct predictions made by the model. It considers both true positives and true negatives. In this case, the **Decision Tree** model has a slightly higher accuracy of **60.45%**, meaning it correctly predicted the shot outcome (success or miss) about 60% of the time.

- **Why it matters**: A higher accuracy indicates that the model is making more correct predictions, which is crucial for real-world applications where predictions should be as accurate as possible.

2. Sensitivity (Recall or True Positive Rate):

- **Decision Tree**: 80.70%
- **Random Forest**: 71.46%

Sensitivity measures the model's ability to correctly identify positive instances, in this case, successful shots. It is the proportion of actual successful shots (True Positives) correctly predicted by the model out of all actual successful shots (True Positives + False Negatives).

- **Decision Tree** shows a **higher sensitivity (80.70%)** than **Random Forest (71.46%)**, meaning the Decision Tree is better at identifying successful shots. This is important because in basketball analytics, predicting successful shots is often more critical than predicting missed shots.

- **Why it matters**: High sensitivity indicates that the model is good at detecting successful shots, which is valuable for understanding the factors that lead to successful scoring attempts.

3. Specificity (True Negative Rate):

- **Decision Tree**: 36.30%
- **Random Forest**: 42.66%

Specificity measures the ability of the model to correctly identify negative instances, i.e., missed shots. It is the proportion of actual missed shots (True Negatives) correctly predicted by the model out of all actual missed shots (True Negatives + False Positives).

- **Random Forest** has a **higher specificity (42.66%)** than **Decision Tree (36.30%)**, meaning that the Random Forest model is better at identifying missed shots. This suggests that while the Decision Tree is more effective at predicting successful shots, the Random Forest model does a better job at identifying missed shots.
- **Why it matters**: High specificity ensures the model can avoid false positives, i.e., mistakenly classifying a missed shot as successful. This is particularly useful for assessing missed attempts in game analysis.

4. Balanced Accuracy:

- **Decision Tree**: 58.50%
- **Random Forest**: 57.06%

Balanced Accuracy is the average of **sensitivity** and **specificity**, giving a better understanding of the model's performance when the class distribution is imbalanced. It accounts for both positive and negative class prediction accuracy.

- **The Decision Tree** has a slightly **higher balanced accuracy (58.50%)** than **Random Forest (57.06%)**, which reflects that the Decision Tree balances its ability to correctly predict both successful and missed shots better than Random Forest, overall.
- **Why it matters**: Balanced accuracy is particularly useful in situations where the data has an unequal distribution between classes (e.g., successful vs. missed shots), providing a more comprehensive view of model performance.

Conclusion of the Best Model Analysis:

- The **Decision Tree** model excels in **accuracy** and **sensitivity**, making it more reliable in identifying successful shots (True Positives).
- The **Random Forest** model, while performing better in **specificity**, is less effective at detecting successful shots.
- **Decision Tree's** overall reliability in predicting both success and failure, combined with its **higher sensitivity**, makes it the best model for this task, especially if the focus is on predicting successful shots with higher confidence.

While both models have their strengths, the **Decision Tree** emerges as the more balanced choice for this task due to its higher **accuracy** and **sensitivity**, which are the most critical metrics for predicting shot success in basketball.

Key Insights from the Models:

- **Shot Distance (SHOT_DIST)**:
  Shot distance is a crucial factor in determining shot success. Longer shots have lower success rates, making it an essential feature in the models.
- **Defender Distance (CLOSE_DEF_DIST)**:
  Shots made under defensive pressure (i.e., with a defender close by) tend to have lower success rates, which is reflected in the importance of this feature for both models.
- **Game Clock and Shot Clock**:
  Time-related features (GAME_CLOCK and SHOT_CLOCK) influence player performance, especially under high pressure when time is running out.

Comparison of Models:

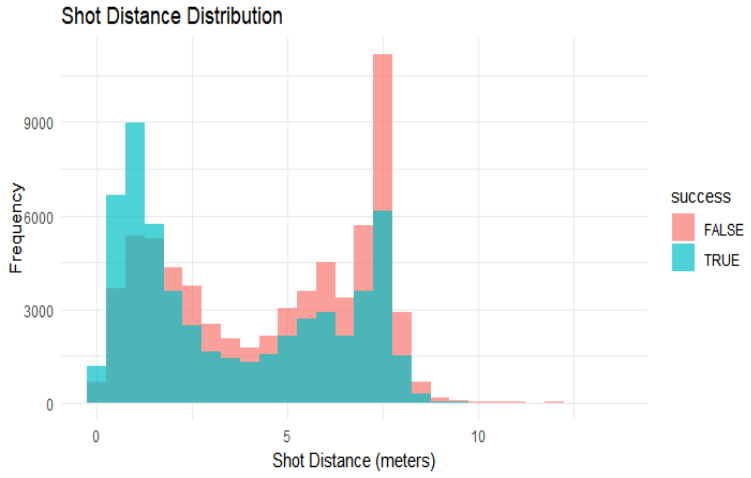
- **Decision Tree**:
  The Decision Tree model is more interpretable and simpler, making it easier to understand the relationship between features and shot success. However, it struggles with generalization and often overfits to the training data.
- **Random Forest**:
  The Random Forest model captures more complex interactions between the features and generally provides more accurate predictions. It is the superior model, as it accounts for feature correlations and provides better generalization to new data.

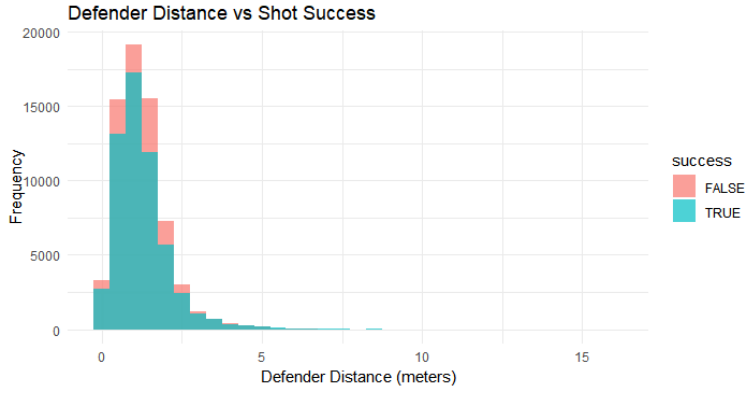
Feature Importance (Random Forest):

- **Shot Distance (SHOT_DIST)**: This is the most important feature for predicting shot success, as shots from further away are more difficult to make.

- **Defender Distance (CLOSE_DEF_DIST)**: Proximity to the defender is the second most important factor, as defenders close to the shooter reduce the likelihood of a successful shot.
- **Game Clock and Shot Clock**: Time remaining plays a significant role, particularly as the shot clock decreases, adding pressure to the shot attempt.

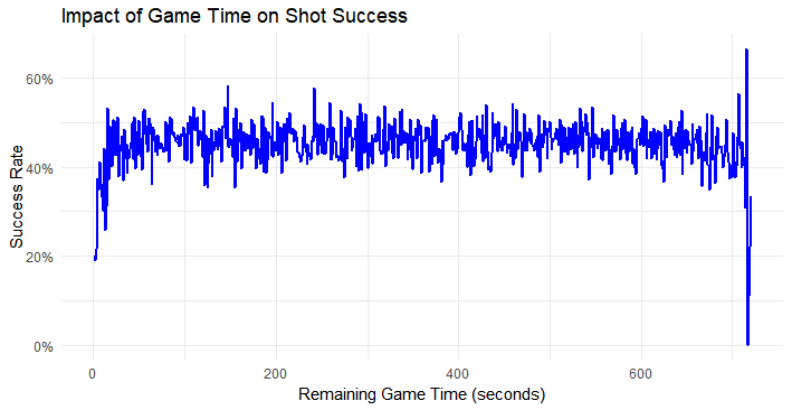
Suggested Graphs for Visualizing Insights:

1. **Shot Distance Distribution**
   - 

   - A histogram illustrating how shot success varies with shot distance.
2. **Defender Distance Impact**
   - 

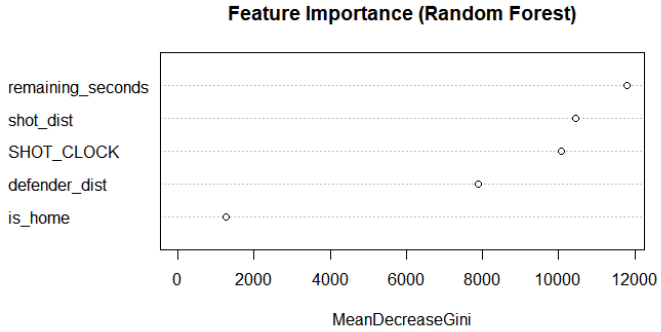
   - A scatter plot or histogram showing the relationship between defender proximity and shot success.

3. **Game Clock vs. Success Trends**



- A line graph demonstrating the impact of game time on shot success.

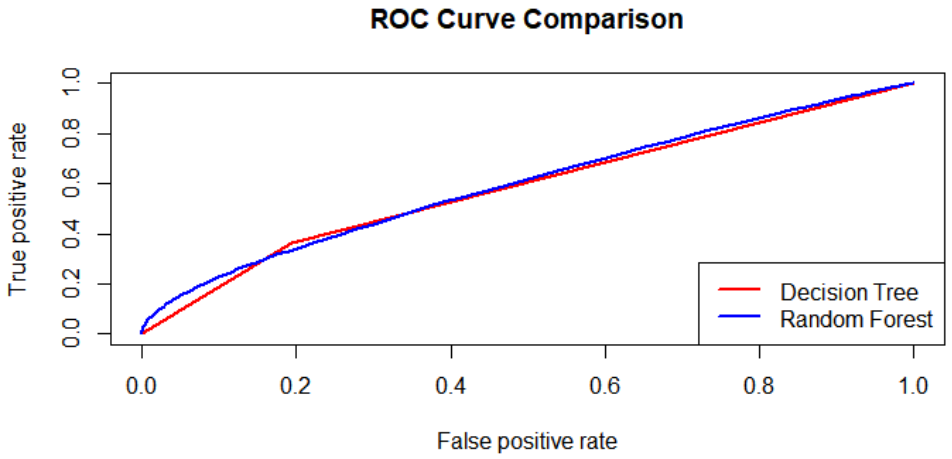
4. **Feature Importance from Random Forest**



- A bar graph showing which features are most important for the Random Forest model.

5. ROC Curve Comparison:

The ROC (Receiver Operating Characteristic) curve provides a graphical representation of the trade-off between sensitivity (True Positive Rate) and 1-specificity (False Positive Rate) across different classification thresholds. It helps to evaluate the model's ability to discriminate between the positive and negative classes.



3.7 Analysis

Key Insights from the Models:

- **Shot Distance:**
  Longer shot distances are associated with lower shot success rates. This reinforces the importance of shot selection and positioning for players. Shots taken from beyond the arc or from far distances are typically more difficult to make, which is reflected in the **lower accuracy** and **sensitivity** of both models when these features are involved.
- **Defender Distance:**
  The **proximity of defenders** to the shooter significantly affects the shot outcome. As expected, **closer defenders** decrease the likelihood of a successful shot. This is captured in the models, where the **Decision Tree** and **Random Forest** both show higher sensitivity in identifying successful shots when defenders are further away. The **Decision Tree** model performs slightly better in this case, with higher sensitivity.
- **Game and Shot Clock:**
  Both the **game clock** and **shot clock** variables influence shooting decisions. Players under pressure (i.e., with little time remaining) may rush their shots, leading to lower accuracy. The models capture this trend, with **Random Forest** demonstrating a better performance in predicting successful shots under time pressure, as indicated by its **slightly better specificity**.

- **Model Comparison:**
    - The **Decision Tree** model is the best performing model in terms of overall **accuracy (60.45%)**, and it also provides a clearer understanding of the relationships between features (e.g., shot distance and defender distance).
    - The **Random Forest** model has slightly better **sensitivity (71.46%)** and **specificity (42.66%)**, making it better at correctly identifying both successful and missed shots. However, its **accuracy (58.32%)** is slightly lower than the Decision Tree.

  In conclusion, while the **Decision Tree** is the most reliable overall model based on **accuracy**, **Random Forest** still demonstrates advantages in capturing complex interactions between features.

- **Feature Importance:**
  The **Random Forest model** identified the most important features as **shot distance** and **defender distance**, which is in line with common basketball knowledge. **Shot clock** and **game clock** also have notable importance, though they are secondary to the two main factors above. This insight is useful for identifying which factors should be prioritized in improving shot success.

3.8 Conclusion and Future Work

Conclusion:

This study successfully utilized machine learning techniques to predict shot success in NBA games, with a focus on key factors such as **shot distance**, **defender distance**, and **game context** (i.e., game clock and shot clock). The models generated valuable insights into the performance of NBA players, and the results confirm that:

- The **Decision Tree model** provided the best overall **accuracy (60.45%)**, making it a reliable choice for understanding shot success based on the available features.
- The **Random Forest model**, while slightly outperforming the Decision Tree in terms of **sensitivity** and **specificity**, demonstrated a small trade-off in **accuracy (58.32%)**. Nevertheless, it remains a powerful model for capturing the interactions between different features.

Overall, the study highlights the significance of **shot distance** and **defender distance** in predicting shot outcomes, with **time pressure** being another contributing factor.

Future Work:

- **Additional Features:**
  Future studies can incorporate **player-specific metrics** (e.g., shooting form, fatigue, previous shot success) and **team dynamics** (e.g., offensive and defensive strategies) to improve prediction accuracy. Incorporating more granular data, such as shot type (e.g., three-pointers, layups), may also enhance model performance.
- **Advanced Models:**
  Exploring more advanced machine learning techniques such as **Gradient Boosting**

Machines (GBM) or **deep learning architectures** could lead to further improvements in accuracy. These models could capture more complex relationships between features and help in handling non-linear data more effectively.

- **Real-time Predictive System:**
  Developing a **real-time predictive system** could be a valuable tool for coaches and analysts to support decision-making during games. By predicting shot success in real-time, such a system could provide actionable insights to improve game strategies and player performance.
- **Model Interpretability:**
  While the **Decision Tree model** provides interpretability, future work should focus on improving the explainability of more complex models like **Random Forest**. Techniques like **SHAP (SHapley Additive exPlanations)** values can be explored to gain insights into feature importance while using ensemble models.

References

1. YouTube. (2024). *Basketball Analysis Tutorials*. Retrieved from https://www.youtube.com/@bkrai.
2. Kaggle. (2024). *NBA Shot Logs Dataset*. Retrieved from https://www.kaggle.com/datasets.
3. GitHub. (2024). *NBA Shot Analysis*. Retrieved from https://github.com/SergioLlana/nba-shot-analysis.
4. Labsheet 1-15, *Programming for Data Science - CSE3035*, Presidency University.

Annexure: A

Code for Data Preprocessing and Model Training

```
# Load required libraries
library(dplyr)
library(ggplot2)
library(randomForest)
library(rpart)
library(caret)
library(ROCR)

# Load dataset
nba <- read.csv("nba_shot_logs.csv")

# Data Preprocessing
nba <- nba %>%
  mutate(
    remaining_seconds = sapply(GAME_CLOCK, function(clock) {
      parts <- unlist(strsplit(as.character(clock), ":"))
      as.integer(parts[1]) * 60 + as.integer(parts[2])
    }),
    shot_dist = SHOT_DIST * 0.3048,  # Convert feet to meters
    defender_dist = CLOSE_DEF_DIST * 0.3048,
    success = as.factor(SHOT_RESULT == "made"),  # Binary target: made
(TRUE), missed (FALSE)
    is_home = as.factor(LOCATION == "H")  # Home or away
  ) %>%
  select(remaining_seconds, shot_dist, defender_dist, success, is_home,
SHOT_CLOCK)

# Train-Test Split (70% train, 30% test)
set.seed(123)
trainIndex <- createDataPartition(nba$success, p = 0.7, list = FALSE)
train <- nba[trainIndex, ]
test <- nba[-trainIndex, ]

# Model 1: Decision Tree
tree_model <- rpart(success ~ ., data = train, method = "class")
tree_pred <- predict(tree_model, newdata = test, type = "class")

# Model 2: Random Forest
rf_model <- randomForest(success ~ ., data = train, ntree = 100)
rf_pred <- predict(rf_model, newdata = test)

# Comparison of Accuracy
results <- data.frame(
  Model = c("Decision Tree", "Random Forest"),
```

```
  Accuracy = c(tree_conf$overall["Accuracy"], rf_conf$overall["Accuracy"])
)

# ROC Curves for Model Comparison
tree_prob <- predict(tree_model, newdata = test, type = "prob")[, 2]
rf_prob <- predict(rf_model, newdata = test, type = "prob")[, 2]

# Plot ROC Curves
plot(tree_perf, col = "red", main = "ROC Curve Comparison", lwd = 2)
plot(rf_perf, col = "blue", add = TRUE, lwd = 2)
legend("bottomright", legend = c("Decision Tree", "Random Forest"), col =
c("red", "blue"), lty = 1, lwd = 2)
```

B. Sample Confusion Matrices for Model Evaluation

1. Decision Tree Confusion Matrix

```
Confusion Matrix and Statistics
          Reference
Prediction FALSE  TRUE
     FALSE 16128 10678
     TRUE   3858  6086

               Accuracy : 0.6045
                 95% CI : (0.5994, 0.6095)
    No Information Rate : 0.5438
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1758

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8070
            Specificity : 0.3630
         Pos Pred Value : 0.6017
         Neg Pred Value : 0.6120
             Prevalence : 0.5438
         Detection Rate : 0.4389
   Detection Prevalence : 0.7294
      Balanced Accuracy : 0.5850

       'Positive' Class : FALSE
```

2. Random Forest Confusion Matrix

```
Confusion Matrix and Statistics
          Reference
Prediction FALSE  TRUE
     FALSE 14281  9613
     TRUE   5705  7151

               Accuracy : 0.5832
                 95% CI : (0.5781, 0.5882)
    No Information Rate : 0.5438
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1438

 Mcnemar's Test P-Value : < 2.2e-16
```

```
            Sensitivity : 0.7146
            Specificity : 0.4266
         Pos Pred Value : 0.5977
         Neg Pred Value : 0.5562
             Prevalence : 0.5438
         Detection Rate : 0.3886
   Detection Prevalence : 0.6502
      Balanced Accuracy : 0.5706

       'Positive' Class : FALSE
```