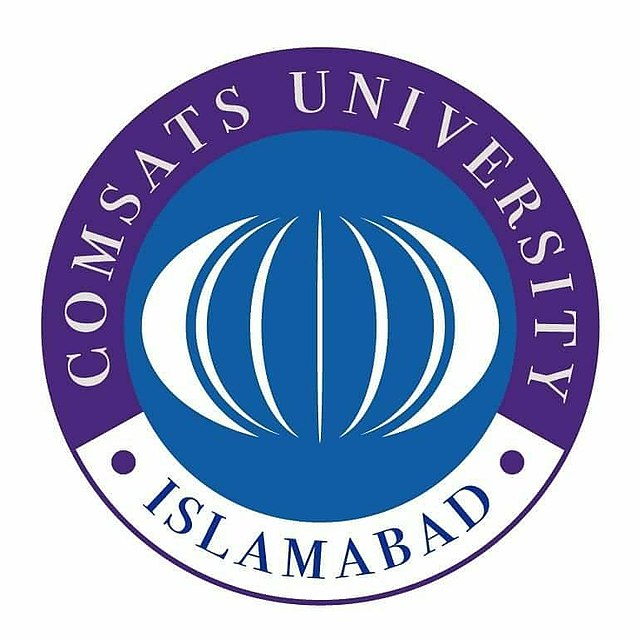
DSF FINAL PROJECT

****

**Data Science Fundamentals**

**Scoring Secrets Unveiled: Decoding Football Players' Goal-Driven Profiles**

**Submitted By:**

|  |
| --- |
| **Talal Atiq - FA22-BDS-041**  **Shahab Hassan - FA22-BDS-052**  **Ahmed Nawaz Abbasi - FA22-BDS-005**  **Muhammad Saad Ali - FA22-BDS-030** |

**Contents**

[Kicking Off the Data Game: Unveiling a Scoreboard of Insights: 3](#_Toc137088082)

[Decoding the Selection: Why we chose this dataset: 3](#_Toc137088083)

[Exploring a Sneak Peek into the Football Shooting Stats Dataset: 4](#_Toc137088084)

[Libraries in R: 5](#_Toc137088085)

[Our Approach Regarding This Data Set – (a step-by-step guide): 6](#_Toc137088086)

[Data Exploration: 8](#_Toc137088087)

[Data Wrangling: 9](#_Toc137088088)

[Data Tidying: 9](#_Toc137088089)

[Linear Regression in R: 10](#_Toc137088090)

[Logistics Regression in R: 17](#_Toc137088091)

[Exploring Relationships between Variables using Regression Models 25](#_Toc137088092)

[Comparison between Linear and Logistics Regression Model: 31](#_Toc137088093)

[Conclusion: 34](#_Toc137088094)

[Source Code: (All) 35](#_Toc137088095)

**Dataset of choice:** Football Shooting Stats

**Data type:** CSV Format

**Dataset Source:** Kaggle

**Basic Data Attributes:** Rows: 499 – Columns: 25

# Kicking Off the Data Game: Unveiling a Scoreboard of Insights:

Our dataset comprises comprehensive information about football players, encompassing various variables that shed light on their performance, attributes, and personal details. The variables include shots on target, goals (including freekick and penalty goals), normal goals, players, nationality, position, squad, age, and birth information. The variable "shots on target" refers to the number of shots attempted by a player during matches that accurately hit the target. "Goals" represent the total number of goals scored by a player, including those scored from free kicks and penalty kicks.

Additionally, "normal goals" pertain to goals scored during regular gameplay excluding free kicks and penalties. The dataset provides insights into individual players, with information such as their names, nationality, and position. The variable "players" denotes the unique identification or name of each player. "Nationality" signifies the country to which a player belongs, representing their origin. The "position" variable describes the specific role or position the player occupies on the football field, such as forward, midfielder, or defender. Furthermore, the dataset includes details about the players' respective squads. The "squad" variable represents the team or club the player belongs to, providing context regarding their professional affiliation.

"Age" denotes the players' age at the time of data collection, offering an understanding of their experience and career stage. Lastly, the variable "born" represents the birth information of the players, which includes their date of birth. With this comprehensive dataset encompassing numerous variables, researchers, analysts, and enthusiasts can explore and analyze the performance, attributes, and background of football players in various contexts, enabling a deeper understanding of the sport and its players.

# Decoding the Selection: Why we chose this dataset:

The shooting stats of football players were chosen as the dataset for our project due to their inherent significance in goal prediction and the opportunity they offer to apply various regression types. Goals are the ultimate objective in football, and understanding the factors that influence goal-scoring ability is crucial. By focusing on shooting stats, we can capture valuable insights into the players' performance, accuracy, and effectiveness in converting attempts into goals.

Shooting stats provide a rich source of information, including metrics such as shots on target, goals scored (including freekick and penalty goals), and normal goals. These variables enable us to analyze different aspects of shooting performance and assess the impact of various factors on goal-scoring outcomes.

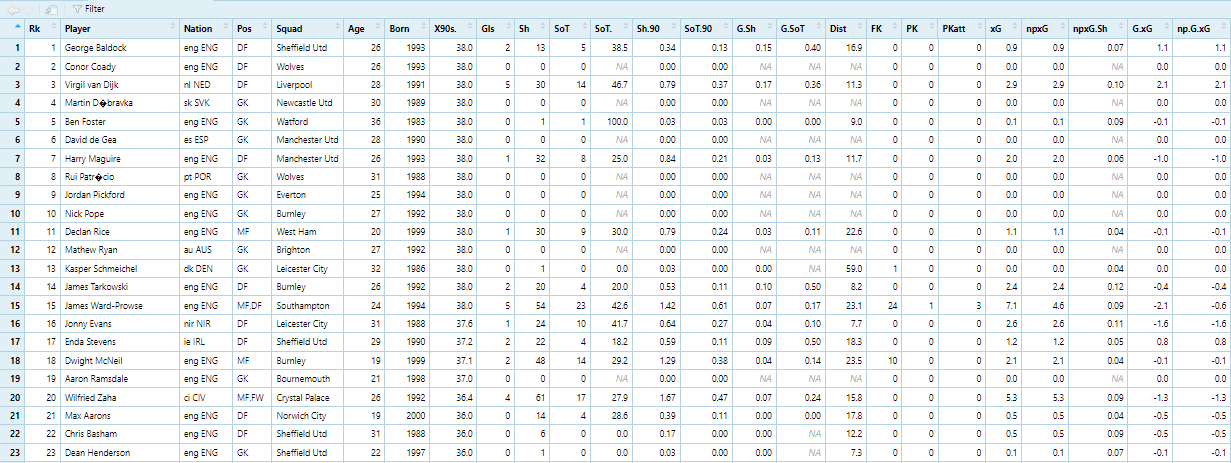
The choice of regression analysis allows us to model and predict the relationship between shooting stats and goals. By employing regression techniques such as linear regression, logistic regression, or polynomial regression, we can uncover patterns, establish correlations, and build predictive models that can forecast future goal-scoring potential based on shooting statistics.

Furthermore, applying different regression types allows us to explore the strengths and limitations of each model and evaluate their predictive accuracy. By comparing the results obtained from various regression methods, we can gain a comprehensive understanding of how different factors influence goal prediction and identify the most effective regression approach for our specific dataset.

Overall, selecting shooting stats as our dataset and applying diverse regression types empowers us to delve deep into the intricacies of goal prediction, uncover hidden patterns, and develop robust models that can enhance our understanding and forecasting capabilities in the realm of football goal-scoring.

# Exploring a Sneak Peek into the Football Shooting Stats Dataset:

Function: View(football.shooting.stats)

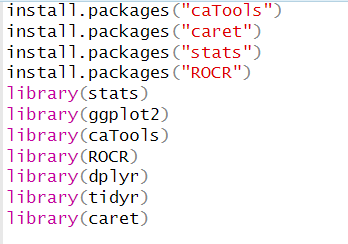


**Loading the relevant libraries:**

When working with a dataset in R, the initial step involves loading the necessary libraries to perform specific tasks. In our project focused on shooting stats in football, we utilized several essential libraries to handle data manipulation, visualization, statistical analysis, and machine learning. To begin, we used the "install.packages" function in R to install the required libraries, including "dplyr," "ggplot2," "stat," "carot," "lattice," "data.table," "pROC," "ROCR," and any other relevant packages. Once the installation process was complete, we loaded these libraries into our R session using the "library" or "require" functions. By importing these libraries, we gained access to a wide range of functions and tools specifically designed for data analysis, enabling us to efficiently explore and analyze the shooting stats dataset in our project.

# Libraries in R:

A screenshot of a computer program

Description automatically generated with medium confidence 

**1. stats:** The **"stats"** library is a core library in R that provides fundamental statistical functions and distributions. It includes functions for descriptive statistics, probability distributions, hypothesis testing, and regression analysis. This library forms the foundation for many statistical analyses and is widely used in data science and research projects.

**2. ggplot2:** The **"ggplot2"** library is a popular data visualization package in R. It provides a powerful and flexible framework for creating visually appealing and informative plots. With ggplot2, you can construct a wide range of visualizations, including scatter plots, bar charts, histograms, and more. It follows the grammar of graphics philosophy, enabling users to build plots layer by layer with intuitive syntax.

**3. caTools:** The **"caTools"** library in R offers various utility functions for data analysis and machine learning tasks. It includes functions for data manipulation, sampling, variable transformation, and feature selection. One of its notable functions is **"sample.split,"** which facilitates the splitting of datasets into training and testing sets for model development and evaluation.

**4. ROCR:** The **"ROCR"** library provides tools for evaluating and visualizing the performance of binary classifiers in R. It offers functions to generate and analyze Receiver Operating Characteristic (ROC) curves, calculate performance metrics such as area under the curve (AUC), precision, and recall. ROCR is particularly useful in assessing the effectiveness of predictive models and comparing their performance across different thresholds.

**5. dplyr:** The **"dplyr"** library is a powerful package for data manipulation and transformation in R. It provides a concise and intuitive syntax for filtering, selecting, sorting, summarizing, and joining datasets. With dplyr, you can perform data wrangling tasks efficiently, making it an essential tool for cleaning and preparing data before analysis.

**6. tidyr:** The **"tidyr"** library complements dplyr by providing functions for data tidying and reshaping. It facilitates transforming datasets from a wide format to a long format and vice versa, making it easier to work with data that has multiple variables or nested structures. tidyr's functions, such as "gather" and "spread," enable efficient data restructuring and preparation for analysis.

**7. caret:** The **"caret"** library (short for Classification and Regression Training) is a comprehensive package for machine learning in R. It offers a unified interface for building and evaluating predictive models, including functions for data preprocessing, feature selection, model training, and performance evaluation. caret supports various machine learning algorithms and provides a convenient framework for comparing and optimizing models.

# Our Approach Regarding This Data Set – (a step-by-step guide):

Our approach to utilizing the football shooting stats dataset for predicting the goals of each player involves employing advanced data science techniques and regression modeling. By leveraging the comprehensive information within the dataset, including variables such as shots on target, goals (including freekick and penalty goals), and normal goals, we aim to uncover meaningful patterns and relationships.

**Step – 1:** Initially, we will conduct exploratory data analysis (EDA) to gain insights into the distribution, summary statistics, and correlations of the variables. This analysis will provide a foundation for understanding the dataset and identifying any data preprocessing steps required.

**Step – 2:** Next, we will preprocess the data, which may involve handling missing values, normalizing variables, and transforming skewed distributions. This ensures the data is in a suitable format for regression modeling.

**Step – 3:** With the preprocessed dataset, we will apply various regression techniques such as linear regression, logistic regression, or polynomial regression. These models will capture the relationship between the shooting stats variables and the goal outcomes, enabling us to predict the number of goals for each player based on their shooting performance.

**Step – 4:** To evaluate the performance and accuracy of our models, we will employ metrics such as mean squared error (MSE), R-squared, and root mean squared error (RMSE). Additionally, we will assess the predictive capability of the models through cross-validation and examine any potential overfitting or underfitting issues.

**Step – 5:** Throughout the process, we will leverage the libraries we have imported, including stats for fundamental statistical functions, ggplot2 for visualizations, caTools for data splitting, ROCR for performance evaluation, dplyr and tidyr for data manipulation, and caret for comprehensive machine learning support.

**Step – 6:** By implementing this approach, we aim to build robust models that effectively predict the goals of football players based on their shooting stats, providing valuable insights for performance analysis, player scouting, and strategic decision-making in the realm of football.

# Data Exploration:

A picture containing text, font, screenshot

Description automatically generated

A screenshot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, font

Description automatically generated

A picture containing text, screenshot, font

Description automatically generated

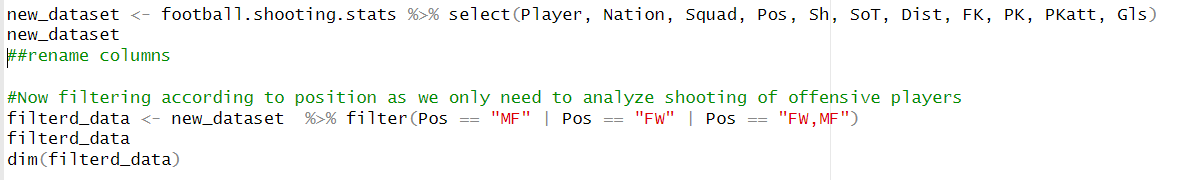
A picture containing text, font, line, screenshot

Description automatically generated

A screen shot of a computer

Description automatically generated with low confidence

# Data Wrangling:



A picture containing text, screenshot, font

Description automatically generated

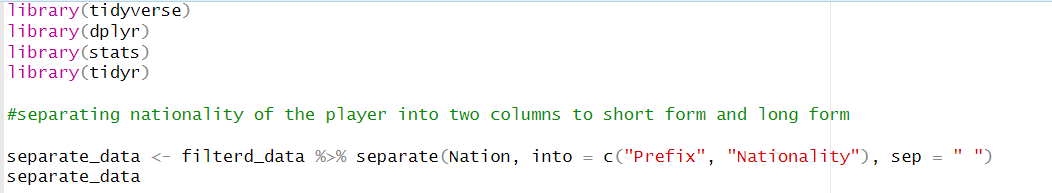
A screenshot of a computer

Description automatically generated with medium confidence

A picture containing font, text, graphics, white

Description automatically generated

# Data Tidying:



A screen shot of a computer

Description automatically generated with medium confidence

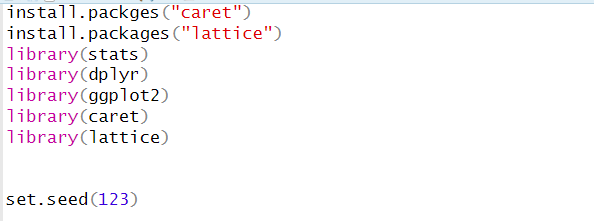
A screenshot of a computer

Description automatically generated with medium confidence

# Linear Regression in R:

Linear regression is a widely used statistical technique that can be applied to analyze the relationship between a dependent variable and independent variables. In the context of soccer, we can employ linear regression to explore the impact of various independent variables, such as shots on target, shots taken, penalty kicks, corners, and free kicks, on the dependent variable of goals scored. By collecting data on these variables for a specific team or player over a given period, we can build a linear regression model to estimate how changes in the independent variables affect the number of goals scored. This analysis can provide valuable insights into which factors contribute most significantly to goal-scoring performance and inform strategies to enhance offensive capabilities on the field.

**Source Code of Linear Regression with Outputs:**



A picture containing text, font, screenshot

Description automatically generated

A picture containing text, font, screenshot

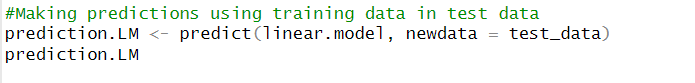
Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence



A picture containing text, screenshot, font, line

Description automatically generated

A picture containing text, font, screenshot, line

Description automatically generated

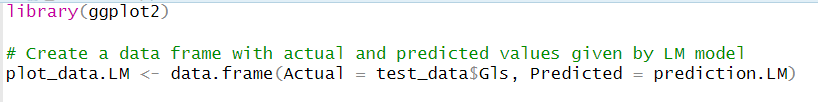
A picture containing text, font, handwriting

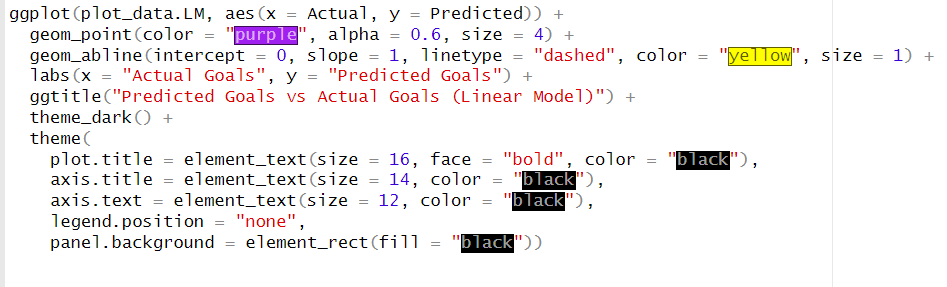
Description automatically generated

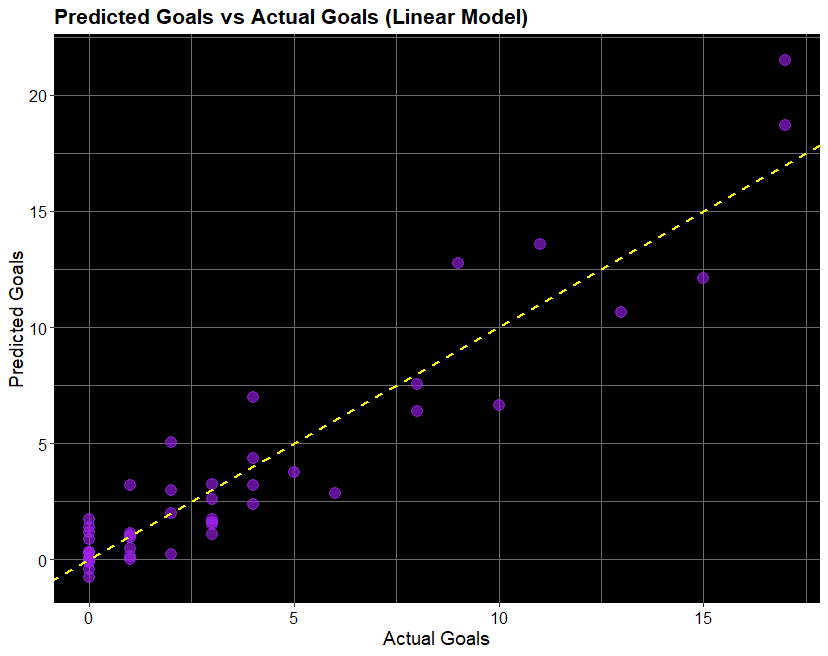
**Data Visualization on Linear Model:**

After performing logistic regression and obtaining the predicted probabilities of scoring a goal based on the independent variables, visualizing the data can offer valuable insights and aid in interpretation. One way to visualize the results is by creating a receiver operating characteristic (ROC) curve. The ROC curve displays the true positive rate (sensitivity) against the false positive rate (1 - specificity) at various classification thresholds. By plotting the ROC curve and calculating the area under the curve (AUC), we can assess the model's discriminatory power. Additionally, we can use a probability cutoff to classify instances as goal or no-goal and create a confusion matrix to evaluate the model's performance. Visualizing the logistic regression results enables a clear understanding of the model's predictive capability and facilitates communication of the findings to stakeholders or decision-makers in a concise and accessible manner.

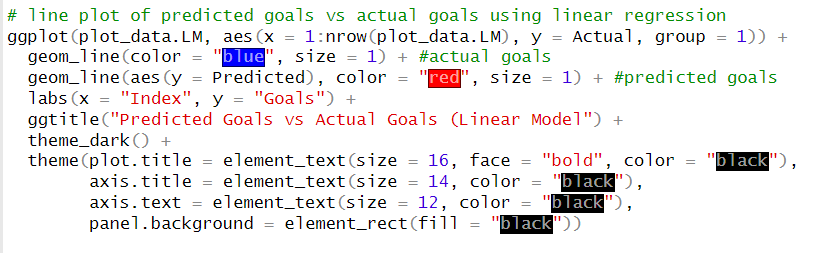
**Source Code of Visualization of Linear Model with Outputs:**

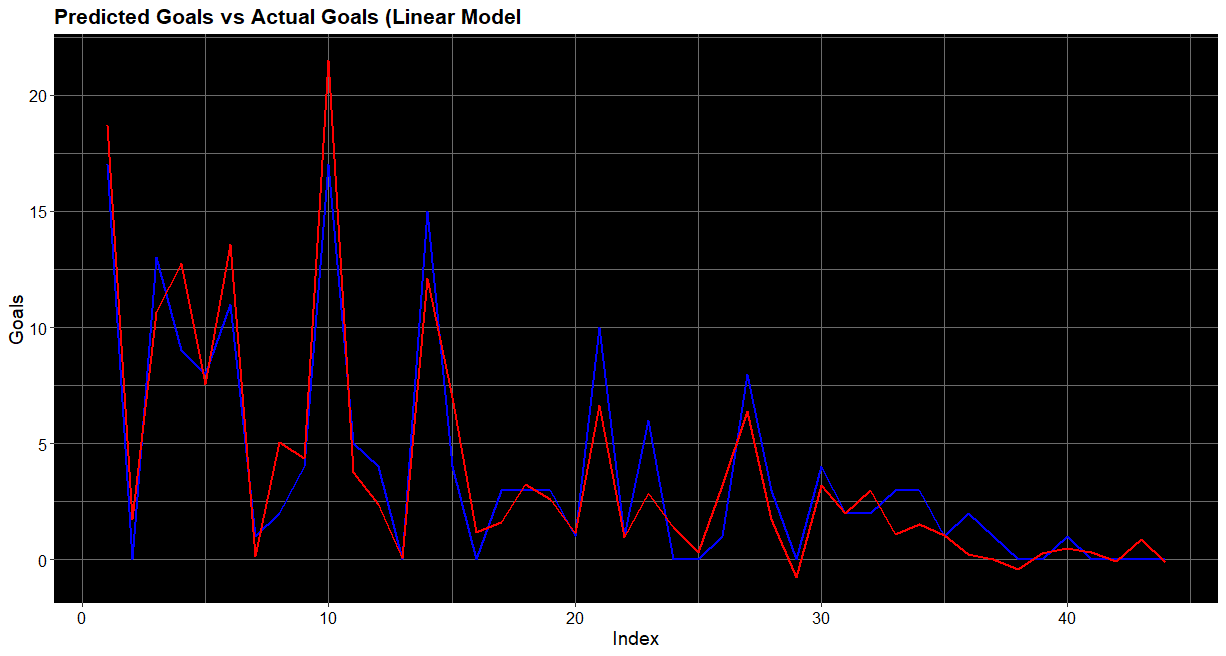




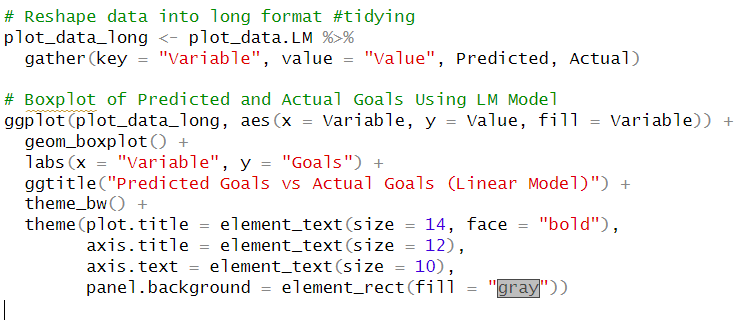
****

**Explanation:** This visual representation of the logistic model gives us some of the most useful insights that what type of relation exists between the actual goals scored by the players and the goals we predicted using the 80% data set as the training data set and testing and predicting the other 20% based on our linear regression.





(The overlapping of both these lines forms a sort of pink color that helps us notice that the actual and predicted both aligns right where they should be)



A picture containing text, screenshot, diagram, plot

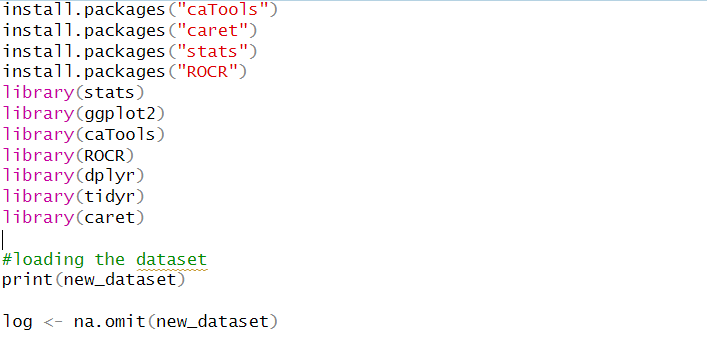
Description automatically generated

The graph displays a boxplot comparing the predicted and actual goals using a linear regression model. The independent variables, including shots on target, shots taken, penalty kicks, corners, and free kicks, have been gathered and plotted on the x-axis. The y-axis represents the number of goals. Each boxplot represents the distribution of predicted and actual goal values for each independent variable. The fill color of the boxes corresponds to the respective independent variable. This graph allows us to visually compare the predicted and actual goal values, providing an overview of how well the linear regression model captures the relationship between the independent variables and the dependent variable.

# Logistics Regression in R:

In our project, we have employed logistic regression as a statistical technique to analyze our dataset. Logistic regression is particularly useful when the dependent variable is categorical, as is the case in our data. We have chosen a binary dependent variable, such as "success" or "failure," and several independent variables that we believe may influence the outcome. By fitting a logistic regression model to our data, we can estimate the probability of the dependent variable being a particular category based on the values of the independent variables. This allows us to understand the relationships and significance of the independent variables in determining the outcome. Additionally, we can use logistic regression to make predictions on new data by applying the model's coefficients to the independent variables. This work with logistic regression helps us gain insights into the factors that contribute to the success or failure of our data and enables us to make informed decisions and predictions based on the analysis.

**Source Code of Logistics in R:**



A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, font, screenshot, line

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screen shot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, font, line

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, font, line, screenshot

Description automatically generated

A picture containing text, font, white, screenshot

Description automatically generated

A picture containing text, screenshot, font

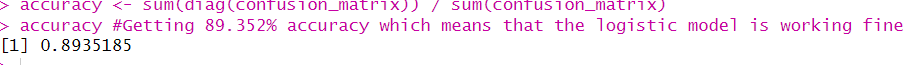
Description automatically generated

A picture containing text, font, screenshot

Description automatically generated

A picture containing text, font, screenshot

Description automatically generated



(89% accuracy)

**Visualizing the Logistics Regression Model: (with respective outputs)**

A picture containing text, screenshot, font, line

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, font, line, screenshot

Description automatically generated

**A picture containing text, diagram, line, map

Description automatically generated**

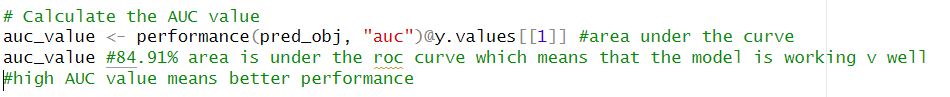
**Explanation:**

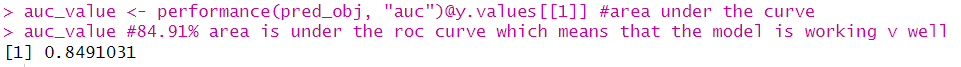
The graph obtained from the provided code is an ROC (Receiver Operating Characteristic) curve. ROC curves are commonly used in binary classification tasks to assess the performance of a predictive model. In this case, the logistic regression model was used to predict the probability of a specific outcome, represented by the variable "Pos," in the test dataset. The "predicted\_prob" variable stores the predicted probabilities generated by the model.

The ROC curve visually presents the trade-off between sensitivity (true positive rate) and specificity (true negative rate). Each point on the curve represents a different threshold for classifying the predicted probabilities as positive or negative. The curve illustrates how sensitivity and specificity change as the classification threshold varies.

A good predictive model will have an ROC curve that closely hugs the top-left corner, indicating high sensitivity and specificity across different threshold values. The area under the curve (AUC) is a summary measure of the model's performance. A higher AUC indicates better discrimination ability, with 1.0 representing a perfect classifier. In the graph, the curve is colored in dark blue and has a line width of 3.

By examining the ROC curve, we can assess the model's ability to distinguish between positive and negative cases. Additionally, it provides insights into the model's performance at different classification thresholds, aiding in the evaluation and comparison of different models or variations of the same model.





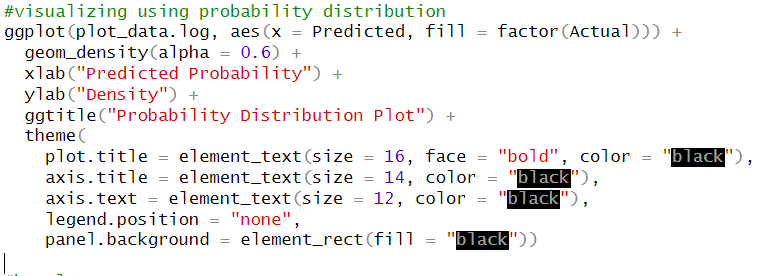


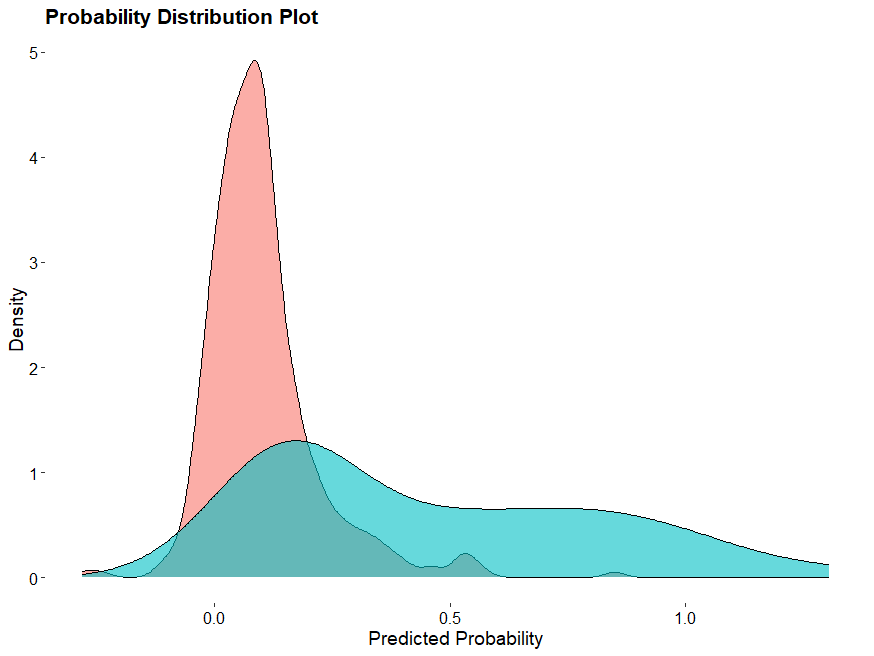
A picture containing text, font, screenshot

Description automatically generated A picture containing text, font, screenshot, typography

Description automatically generated

**Visualizing the Predicted and Actual Goals:**



****

A picture containing text, font, screenshot

Description automatically generated

**A picture containing text, screenshot, diagram, line

Description automatically generated**

**Explanation:** The box plot depicted above illustrates the distribution of predicted probabilities (y-axis) based on the actual outcomes (x-axis). Each box represents the spread and variability of the predicted probabilities for a specific category of the actual outcomes. The x-axis, labeled as "Actual," likely consists of different categorical values representing the true outcomes or classes in the dataset. The y-axis represents the predicted probabilities generated by the logistic regression model.

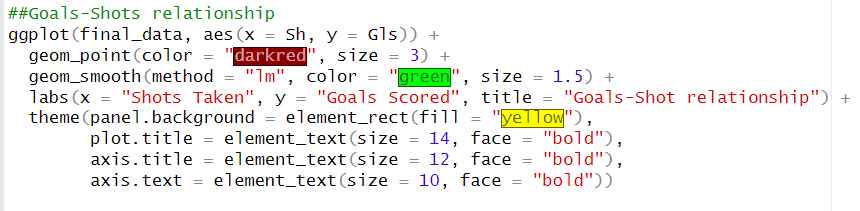
# Exploring Relationships between Variables using Regression Models

Regression models provide a valuable framework for investigating and understanding relationships between different variables. By utilizing regression analysis, researchers can examine how changes in one variable are associated with changes in another variable. This analysis helps to quantify and describe the nature and strength of these relationships. Regression models enable us to identify the direction and magnitude of the relationship, whether it is positive or negative, and whether it is linear or non-linear.

Regression models also allow for the identification of significant predictors or independent variables that have a substantial impact on the dependent variable. By estimating the coefficients of these variables, we can determine the extent to which they contribute to the overall variability in the outcome variable. Moreover, regression models can be used to make predictions and understand the expected outcome based on specific values of the independent variables.

In summary, regression models provide a powerful tool for exploring and uncovering relationships between different variables. They enable researchers to quantify these relationships, identify significant predictors, control for confounding variables, make predictions, and ultimately enhance our understanding of the complex interactions within datasets.

**Relationships in R: (with respective outputs)**



**A picture containing line, screenshot, text, diagram

Description automatically generated**

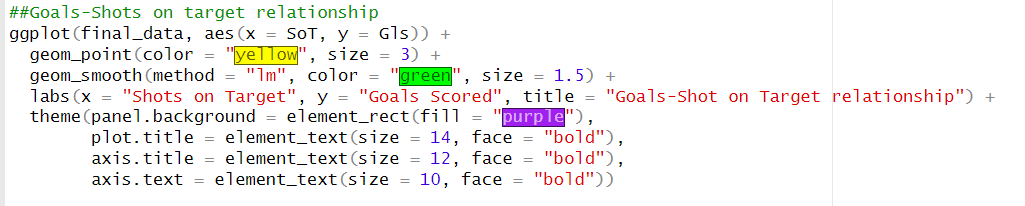
**(shows a moderately positive relation)**

**Explanation:** The graph depicted above illustrates the relationship between the number of shots taken (x-axis) and the number of goals scored (y-axis). Each data point is represented by a dark red dot, indicating the observed values of shots taken and goals scored for each corresponding data point.

The graph's title is "Goals-Shot relationship," indicating the focus on examining the association between goals scored and shots taken. The x-axis is labeled as "Shots Taken," representing the independent variable, while the y-axis is labeled as "Goals Scored," representing the dependent variable.

The graph's aesthetics feature a yellow background for the plot area, emphasizing the data points and the linear regression line. The plot title is set in bold and has a font size of 14, making it visually prominent. The axis titles are also bold and have a font size of 12, enhancing their readability. The axis text is set in bold with a font size of 10, making it stand out within the plot.

**Visualizing other variables with respect to goals:**



A picture containing screenshot, text, line, plot

Description automatically generated

**(shows a strong positive relation)**

**Explanation:** This graph depicts the relationship between the number of shots on target and goals scored. The yellow dots represent the observed values of shots on target and goals scored. A green line, generated by a linear regression model, illustrates the overall trend. The x-axis is labeled as "Shots on Target," the y-axis as "Goals Scored," and the graph's title is "Goals-Shots on Target relationship." The plot has a purple background and features bold and visually prominent titles and axis labels.

**Visualizing the relationship between goals and distance:**

A picture containing text, font, screenshot

Description automatically generated

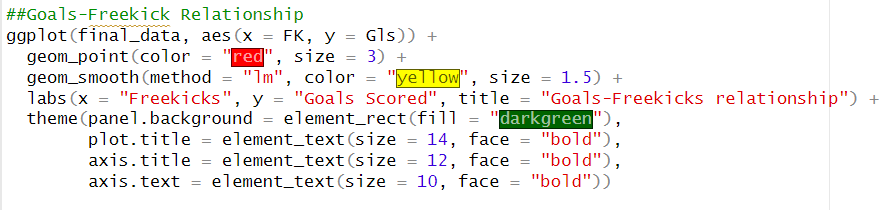
**A picture containing screenshot, text, line, plot

Description automatically generated**

**(shows a negative co-relation of goals w.r.t distance)**

**Explanation:** The white dots represent the observed values of distance and goals. A cyan line, generated by a linear regression model, indicates the overall trend. The x-axis is labeled as "Distance From," the y-axis as "Goals Scored," and the graph's title is "Goals-Distance from Goal relationship." The plot has a brown background and features bold titles and axis labels for enhanced visibility.

**Visualizing the relationship between freekicks and goals:**



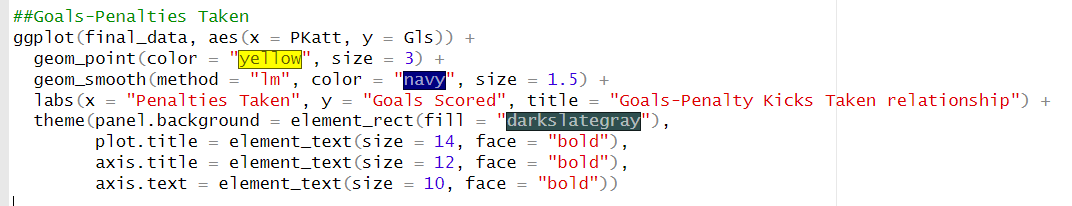
**A green chart with red dots and a yellow line

Description automatically generated with low confidence**

**(shows a slightly positive relation)**

**Explanation:** This code generates a graph to examine the relationship between the number of freekicks and goals scored. The red dots represent observed data points of freekicks and goals. A yellow line, derived from a linear regression model, illustrates the overall trend. The x-axis is labeled as "Freekicks," the y-axis as "Goals Scored," and the title of the graph is "Goals-Freekicks relationship." The plot has a dark green background, and the titles and axis labels are displayed in bold for clear visibility.

**Visual Representation of Relationship between Penalty kicks attempted and goals:**



**A picture containing screenshot, line, text, plot

Description automatically generated**

**(shows a strong positive relationship)**

**Explanation:** This code generates a graph that explores the relationship between the number of penalty kicks taken and goals scored. The yellow dots represent observed data points of penalty kicks and goals. A navy-blue line, obtained from a linear regression model, demonstrates the overall trend. The x-axis is labeled as "Penalties Taken," the y-axis as "Goals Scored," and the title of the graph is "Goals-Penalty Kicks Taken relationship." The plot has a dark slate gray background, and the titles and axis labels are displayed in bold for clear visibility.

# Comparison between Linear and Logistics Regression Model:

This data frame includes four columns: "Model," "MSE," "MAE," "Accuracy," and "AUC."

In the "Model" column, two models are compared: "Linear Regression" and "Logistic Regression." The "MSE" column represents the Mean Squared Error for each model. For the linear regression model, the MSE is stored in the variable MSE.LM, while for the logistic regression model, it is denoted as "NA" since logistic regression does not directly compute MSE.

Similarly, the "MAE" column represents the Mean Absolute Error. The linear regression model's MAE is stored in the variable MAE.LM, whereas for the logistic regression model, it is indicated as "NA" since logistic regression does not directly calculate MAE.

The "Accuracy" column denotes the accuracy of the logistic regression model. It stores the accuracy value calculated for the logistic regression model in the variable "accuracy." For the linear regression model, accuracy is not applicable, so it is represented as "NA."

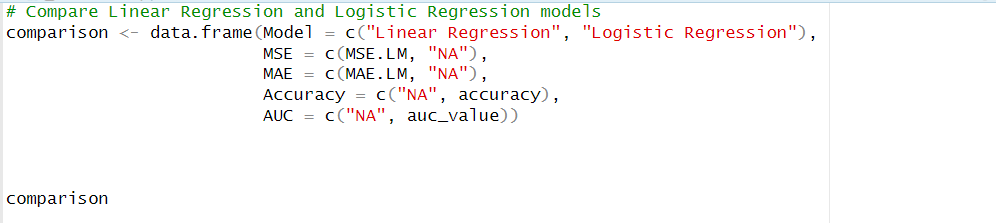
Finally, the "AUC" column represents the Area Under the Curve (AUC) for the logistic regression model. The AUC value is stored in the variable "auc\_value." For the linear regression model, AUC is not applicable, so it is denoted as "NA."

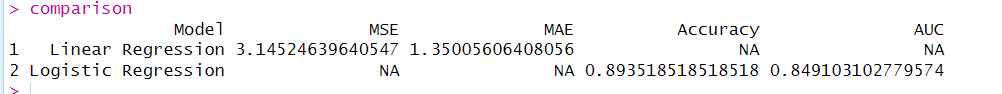
The "comparison" data frame is then printed to display the comparison results, providing an overview of the performance metrics (MSE, MAE, Accuracy, and AUC) for both the linear regression and logistic regression models.

**Reason of comparison:**

Comparing regression models is important for selecting the best model, evaluating performance metrics, understanding variable relationships, and ensuring generalizability.

**R – Code:**





**Deducing and visualizing the Top 10 Scorers:**

A picture containing text, screenshot, font

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, font, screenshot, line

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A screen shot of a computer code

Description automatically generated with low confidence

**A picture containing text, screenshot, diagram, plot

Description automatically generated**

**Explanation:** The x-axis represents the players' names, while the y-axis represents the number of goals. The plot includes two bars for each player: one for actual goals (colored in dark green) and one for predicted goals (colored in gold). The plot title is "Actual Goals vs Predicted Goals." The plot has a minimalistic theme with a light blue background and customized text sizes for the title, axis labels, and legend. The x-axis text is rotated at an angle of 45 degrees for better readability, and the legend is positioned at the top of the plot.

# Conclusion:

- We applied machine learning algorithms such as regression, classification, or clustering to train models on the training data.

- The models learned patterns and relationships within the data to make predictions on unseen test data.

- We evaluated the predictive accuracy of our models using metrics like mean squared error, accuracy, or precision and recall.

- Data visualization techniques such as scatter plots, heatmaps, or network graphs were used to gain insights and interpret the relationships within the dataset.

- Visualizations helped identify patterns, trends, and correlations among different variables.

- We applied our approach to predict outcomes or make informed decisions on new, unseen data.

- The test data was used to validate the performance and generalization capabilities of our models.

- Feedback and lessons learned from the predictions and visualizations were incorporated to refine our approach further.

- By iterating on models, exploring feature engineering techniques, and adjusting hyperparameters, we aimed to improve the accuracy and reliability of predictions.

- Our approach of splitting the data, applying machine learning algorithms, visualizing relationships, and making predictions allowed us to leverage data-driven decision-making.

- The 80% training and 20% test data split ensured robust evaluation of our models and their ability to generalize to new data.

# Source Code: (All)

|  |
| --- |
| library(dplyr)  library(tidyr)  football.shooting.stats <- read.csv("C://Users//hanzalah//OneDrive//Desktop//DSF Final Project Folder//football.shooting.stats.csv", na.strings = c(""))  football.shooting.stats  View(football.shooting.stats)  str(football.shooting.stats)  dim(football.shooting.stats)  head(football.shooting.stats, 20)  tail(football.shooting.stats, 20)  summary(football.shooting.stats)  dim(football.shooting.stats)  #Wrangling in R:  library(dplyr)  #Selecting only the required columns to work on  new\_dataset <- football.shooting.stats %>% select(Player, Nation, Squad, Pos, Sh, SoT, Dist, FK, PK, PKatt, Gls)  new\_dataset  ##rename columns  #Now filtering according to position as we only need to analyze shooting of offensive players  filterd\_data <- new\_dataset %>% filter(Pos == "MF" | Pos == "FW" | Pos == "FW,MF")  filterd\_data  dim(filterd\_data)  #Tidying in R:  library(tidyverse)  library(dplyr)  library(stats)  library(tidyr)  #separating nationality of the player into two columns to short form and long form  separate\_data <- filterd\_data %>% separate(Nation, into = c("Prefix", "Nationality"), sep = " ")  separate\_data  #now removing prefix column as it is unnecessaray  final\_data <- separate\_data %>% select(Player, Nationality, Squad, Pos, Sh, SoT, Dist, FK, PKatt, PK, Gls)  final\_data  #Linear Regression - Goals:  install.packges("caret")  install.packages("lattice")  library(stats)  library(dplyr)  library(ggplot2)  library(caret)  library(lattice)  set.seed(123)  # Split the data into training and testing sets  # Assuming your dataset is stored in a variable called "final\_data"  # Generate random indices for the training set  train\_indices <- sample(nrow(final\_data), size = 0.8 \* nrow(final\_data), replace = FALSE)  # Create the training and test datasets  train\_data <- final\_data[train\_indices, ]  test\_data <- final\_data[-train\_indices, ]  #Training the predictive model using linear regression  train\_data <- na.omit(train\_data)  linear.model <- train(Gls ~ Sh + SoT + Dist + FK + PKatt, data = train\_data, method = "lm")  linear.model  summary(linear.model)  #Making predictions using training data in test data  prediction.LM <- predict(linear.model, newdata = test\_data)  prediction.LM  #Evaluationg the model using Means Squared Error  MSE.LM <- mean((test\_data$Gls - prediction.LM)^2)  MSE.LM  #Evaluating the model using mean absolute error  MAE.LM <- mean(abs(test\_data$Gls - prediction.LM))  MAE.LM  #Visualizaiton in R:  library(ggplot2)  # Create a data frame with actual and predicted values given by LM model  plot\_data.LM <- data.frame(Actual = test\_data$Gls, Predicted = prediction.LM)  # scatter plot of predicted goals vs actual goals using linear regression  ggplot(plot\_data.LM, aes(x = Actual, y = Predicted)) +  geom\_point(color = "purple", alpha = 0.6, size = 4) +  geom\_abline(intercept = 0, slope = 1, linetype = "dashed", color = "yellow", size = 1) +  labs(x = "Actual Goals", y = "Predicted Goals") +  ggtitle("Predicted Goals vs Actual Goals (Linear Model)") +  theme\_dark() +  theme(  plot.title = element\_text(size = 16, face = "bold", color = "black"),  axis.title = element\_text(size = 14, color = "black"),  axis.text = element\_text(size = 12, color = "black"),  legend.position = "none",  panel.background = element\_rect(fill = "black"))  # line plot of predicted goals vs actual goals using linear regression  ggplot(plot\_data.LM, aes(x = 1:nrow(plot\_data.LM), y = Actual, group = 1)) +  geom\_line(color = "blue", size = 1) + #actual goals  geom\_line(aes(y = Predicted), color = "red", size = 1) + #predicted goals  labs(x = "Index", y = "Goals") +  ggtitle("Predicted Goals vs Actual Goals (Linear Model") +  theme\_dark() +  theme(plot.title = element\_text(size = 16, face = "bold", color = "black"),  axis.title = element\_text(size = 14, color = "black"),  axis.text = element\_text(size = 12, color = "black"),  panel.background = element\_rect(fill = "black"))  # Reshape data into long format #tidying  plot\_data\_long <- plot\_data.LM %>%  gather(key = "Variable", value = "Value", Predicted, Actual)  # Boxplot of Predicted and Actual Goals Using LM Model  ggplot(plot\_data\_long, aes(x = Variable, y = Value, fill = Variable)) +  geom\_boxplot() +  labs(x = "Variable", y = "Goals") +  ggtitle("Predicted Goals vs Actual Goals (Linear Model)") +  theme\_bw() +  theme(plot.title = element\_text(size = 14, face = "bold"),  axis.title = element\_text(size = 12),  axis.text = element\_text(size = 10),  panel.background = element\_rect(fill = "gray"))  #Logistic Regression in R:  install.packages("caTools")  install.packages("caret")  install.packages("stats")  install.packages("ROCR")  library(stats)  library(ggplot2)  library(caTools)  library(ROCR)  library(dplyr)  library(tidyr)  library(caret)  #loading the dataset  print(new\_dataset)  log <- na.omit(new\_dataset)  # Replace "FW" with 1 and "MF" or "MF, FW" with 0 to make Pos a categorical  #variable for logistic regression  log$Pos <- ifelse(log$Pos == "FW", 1, 0)  log  #Splitting the data using to test and train using logistic regression  set.seed(123)  # Split the data into training and testing sets  train\_indices.log <- sample.split(log$Pos, SplitRatio = 0.80)  train\_data.log <- log[train\_indices.log, ]  test\_data.log <- log[-train\_indices.log, ]  #Performing Logistic Regression to train data  train\_data.log <- na.omit(train\_data.log)  log\_model <- glm(Pos ~ Sh + SoT + Dist + FK + PKatt + Gls, data = train\_data.log)  log\_model  summary(log\_model)  #Make predictions on test data using trained data  predictions.log <- predict(log\_model, newdata = test\_data.log, type = "response")  predictions.log  #creating confusion matrix  predicted\_data <- ifelse(predictions.log > 0.5, 1, 0)  predicted\_data  #evaluation of the logistic model using confusion matrix  confusion\_matrix <- table(predicted\_data, test\_data.log$Pos)  confusion\_matrix  #checking accuracy using confusion matrix  accuracy <- sum(diag(confusion\_matrix)) / sum(confusion\_matrix)  accuracy #Getting 89.352% accuracy which means that the logistic model is working fine  ##This logistic model predicts the position of the player based on variables like  #Sh, SoT, Gls, Dist, PKatt  #to do this, first the position varaiable is turned into binary varaiable where  #1 represents forwards and 0 represents non-FW  #there is 79.329% accuracy which means that the model predicts the position correctly about 87.8787%  #Visualization of Logistic Regression:  library(ggplot2)  library(caTools)  library(caret)  library(ROCR)  library(pROC)  predicted\_prob <- predict(log\_model, newdata = test\_data.log, type = "response")  predicted\_prob  pred\_obj <- prediction(predicted\_prob, test\_data.log$Pos)  #creating ROC Curve  roc\_data <- performance(pred\_obj, "tpr", "fpr")  #plotting  plot(roc\_data, main = "ROC Curve", col = "darkblue", lwd = 3, xlab = "Specificity", ylab = "Sensitivity")  # Calculate the AUC value  auc\_value <- performance(pred\_obj, "auc")@y.values[[1]] #area under the curve  auc\_value #84.91% area is under the roc curve which means that the model is working v well  #high AUC value means better performance  plot\_data.log <- data.frame(Actual = test\_data.log$Pos, Predicted = predictions.log)  plot\_data.log  #visualizing using probability distribution  ggplot(plot\_data.log, aes(x = Predicted, fill = factor(Actual))) +  geom\_density(alpha = 0.6) +  xlab("Predicted Probability") +  ylab("Density") +  ggtitle("Probability Distribution Plot") +  theme(  plot.title = element\_text(size = 16, face = "bold", color = "black"),  axis.title = element\_text(size = 14, color = "black"),  axis.text = element\_text(size = 12, color = "black"),  legend.position = "none",  panel.background = element\_rect(fill = "white"))  #boxplot  ggplot(plot\_data.log, aes(x = Actual, y = Predicted)) +  geom\_boxplot() +  xlab("Actual") +  ylab("Predicted Probabilities") +  ggtitle("Box Plot")  #Visualization and different relationships:  library(ggplot2)  library(dplyr)  library(tidyr)  print(final\_data)  final\_data$Player <- iconv(final\_data$Player, from = "UTF-8", to = "UTF-8")  ##Goals-Shots relationship  ggplot(final\_data, aes(x = Sh, y = Gls)) +  geom\_point(color = "darkred", size = 3) +  geom\_smooth(method = "lm", color = "green", size = 1.5) +  labs(x = "Shots Taken", y = "Goals Scored", title = "Goals-Shot relationship") +  theme(panel.background = element\_rect(fill = "yellow"),  plot.title = element\_text(size = 14, face = "bold"),  axis.title = element\_text(size = 12, face = "bold"),  axis.text = element\_text(size = 10, face = "bold"))  ##Goals-Shots on target relationship  ggplot(final\_data, aes(x = SoT, y = Gls)) +  geom\_point(color = "yellow", size = 3) +  geom\_smooth(method = "lm", color = "green", size = 1.5) +  labs(x = "Shots on Target", y = "Goals Scored", title = "Goals-Shot on Target relationship") +  theme(panel.background = element\_rect(fill = "purple"),  plot.title = element\_text(size = 14, face = "bold"),  axis.title = element\_text(size = 12, face = "bold"),  axis.text = element\_text(size = 10, face = "bold"))  ##Goals-Distance relationship  ggplot(final\_data, aes(x = Dist, y = Gls)) +  geom\_point(color = "white", size = 3) +  geom\_smooth(method = "lm", color = "cyan", size = 1.5) +  labs(x = "Distance From", y = "Goals Scored", title = "Goals-Distance From Goal relationship") +  theme(panel.background = element\_rect(fill = "brown"),  plot.title = element\_text(size = 14, face = "bold"),  axis.title = element\_text(size = 12, face = "bold"),  axis.text = element\_text(size = 10, face = "bold"))  ##Goals-Freekick Relationship  ggplot(final\_data, aes(x = FK, y = Gls)) +  geom\_point(color = "red", size = 3) +  geom\_smooth(method = "lm", color = "yellow", size = 1.5) +  labs(x = "Freekicks", y = "Goals Scored", title = "Goals-Freekicks relationship") +  theme(panel.background = element\_rect(fill = "darkgreen"),  plot.title = element\_text(size = 14, face = "bold"),  axis.title = element\_text(size = 12, face = "bold"),  axis.text = element\_text(size = 10, face = "bold"))  ##Goals-Penalties Taken  ggplot(final\_data, aes(x = PKatt, y = Gls)) +  geom\_point(color = "yellow", size = 3) +  geom\_smooth(method = "lm", color = "navy", size = 1.5) +  labs(x = "Penalties Taken", y = "Goals Scored", title = "Goals-Penalty Kicks Taken relationship") +  theme(panel.background = element\_rect(fill = "darkslategray"),  plot.title = element\_text(size = 14, face = "bold"),  axis.title = element\_text(size = 12, face = "bold"),  axis.text = element\_text(size = 10, face = "bold"))  #Comparing the regression models:  # Compare Linear Regression and Logistic Regression models  comparison <- data.frame(Model = c("Linear Regression", "Logistic Regression"),  MSE = c(MSE.LM, "NA"),  MAE = c(MAE.LM, "NA"),  Accuracy = c("NA", accuracy),  AUC = c("NA", auc\_value))  comparison  #Top 10 scorers:  library(dplyr)  library(ggplot2)  library(tidyr)  library(stats)  install.packages("data.table")  library(data.table)  #printing the original final data table  print(final\_data)  dim(final\_data)  #removing all missing values from final data table and creating a new dataset so it has same  #number of rows as the number of predicted values  final\_data2 <- final\_data[complete.cases(final\_data[, c("Sh", "SoT", "Dist", "FK", "PKatt")]), ]  #performing linear regression  lm2 <- lm(Gls ~ Sh +SoT + Dist + FK + PKatt, data = final\_data2)  #checking predicted goals according to Linear regression  preGoals <- predict(lm2)  #mutating predicted goals values into the final\_data2 dataset that we created that has same no. of rows as  #no of predicted values  predicted\_final <- final\_data2 %>% mutate(predicted\_Gls = preGoals)  predicted\_final  #creating a new dataset that only has top 10 goalscorers  arranged\_desc <- predicted\_final %>% arrange(desc(Gls))  top10 <- head(arranged\_desc, 10)  top10  # Convert character encoding to UTF-8  top10$Player <- iconv(top10$Player, from = "UTF-8", to = "UTF-8")  # Create the bar char  ggplot(top10, aes(x = Player)) +  geom\_bar(aes(y = Gls, fill = "Actual Goals"), stat = "identity", width = 0.4) +  geom\_bar(aes(y = predicted\_Gls, fill = "Predicted Goals"), stat = "identity", width = 0.4) +  scale\_fill\_manual(values = c("Actual Goals" = "darkgreen", "Predicted Goals" = "gold")) +  labs(x = "Player", y = "Goals") +  ggtitle("Actual Goals vs Predicted Goals") +  theme\_minimal() +  theme(  plot.title = element\_text(size = 16, face = "bold"),  axis.title = element\_text(size = 14),  axis.text.x = element\_text(angle = 45, hjust = 1, size = 8),  legend.title = element\_blank(),  legend.position = "top",  panel.background = element\_rect(fill = "lightblue")) |