

Dear students,

Below given are the Assignment #1 submission guidelines of each team:

Due Date: 04:00PM of 23.03.2024

\*If necessary wherever required add more fields

|  |  |
| --- | --- |
| Title of the Minor Project: | Cricket Player Analytics |
| Name of Team Player – 1 | Jangam Monesh |
| Roll No of Team Player - 1 | 1601-21-771-039 |
| Name of Team Player – 2 | Md.Mushtaq |
| Roll No of Team Player - 2 | 1601-21-771-050 |
| Name of Team Player – 3 | M.GopiPrashanthRaju |
| Roll No of Team Player - 3 | 1601-21-771-051 |
| Title of the Data Set | Cricksheet\_json\_files |
| **Description of Dataset:**  Size of the dataset  URL of the dataset  Summary / Description and Domain of each attribute  No. of missing values in each attribute | **SIZE**  83 json files containing meta info and ball to ball information  **URL:**  https://cricsheet.org/downloads/  **Summary:**  Meta Information:  Data Version: Version of the dataset.  Created: Date when the dataset was created.  Revision: Revision number of the dataset.  Match Information:  Date: Date of the match.  Event: Event name, match number, and group.  Gender: Gender of players (male).  Match Type: Type of match (ODI).  Match Type Number: Unique identifier for the match type.  City: City where the match took place.  Venue: Venue where the match was held.  Season: Season of the match.  Teams: Names of the teams playing the match.  Team Type: Type of teams (international).  Balls per Over: Number of balls per over.  Overs: Total number of overs per innings.  Toss: Toss decision and winner.  Officials:  Match Referees: Referees for the match.  TV Umpires: TV umpires for the match.  Umpires: On-field umpires for the match.  Outcome:  Winner: Winning team.  By: Margin of victory (wickets, runs, etc.).  Player Information:  Players: List of players for each team.  Innings:  Team: Name of the team batting.  Overs: Detailed information about each over, including deliveries.  **No. of missing values in each attribute**  No Missing Values |
|  |  |
| **Please mention if you have created the dataset or added some more fields to the dataset.**  Give detailed description of your contribution with resources from which the data is curated | **Contribution Description:**  We compiled the dataset by extracting data from CSV files in a ball-by-ball manner, capturing crucial details such as runs scored by batsmen, balls played, runs given by bowlers, balls bowled, and extras awarded. Additionally, we enhanced the dataset by including fields related to the performance of both batsmen and bowlers in each match, thereby providing comprehensive ball-by-ball statistics for analysis.  **Resources Used:**  CSV Files: We utilized CSV files containing ball-by-ball information for each of the 83 cricket matches between India and Australia from 2003 to 2023.  Cricket Match Archives: Our dataset is grounded in real cricket matches between India and Australia, sourced from match archives and online scorecards spanning two decades.  Data Extraction Tools: Leveraging scripting or programming languages such as Python, we meticulously extracted and processed the data from the CSV files, ensuring accuracy and completeness.  Cricket Statistics Databases: We may have consulted cricket statistics databases or APIs to cross-reference and validate the information, maintaining the dataset's reliability.  Overall, our collaborative effort involved gathering, processing, and structuring the ball-by-ball data to create an exhaustive dataset suitable for in-depth cricket match analysis and research. |
|  |  |
| **Description of Task#1** | Ball to Ball data conversion from the existing json file |
| Required Pre-processing | Conversion from json to csv |
| Attributes involved | Batter,Baller,runsperball,totalscore,extras,wickets,win margin,winner,tos winner,decision |
| R-Code with necessary comments | # Load required library  library(jsonlite)  # install.packages(hash)  jsonData <- read\_json(path = "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/2003/65244.json")  print(jsonData)  #Mushtaq's json path  # file\_path <- "D:/Minor\_Project-II/Data/IND vs AUS/ODI/2003/65244.json"  #Defining function  convert\_to\_csv <- function(file\_path){  jsonData <- read\_json(path = file\_path)  balls <- 0  flag <- TRUE  batter\_vector<-c()  baller\_vector <- c()  balls\_vector <- c()  runsperball\_vector <- c()  extra\_vector <- c()  country\_vector <- c()  Innings\_vector <- c()  toss <- c()  toss\_winner <- c()  Innings <- c(1, 2)  toss\_decision <- c()  match\_winner <- c()  margin <- c()  total\_score=c()  wickets=c()  cum\_wickets=0  for (inn in Innings) {  cummulative\_score=0  flag <- TRUE  balls<-0  cumm\_wickets=0  while (balls <= 300 && flag) {  # print(toss\_decision)  tryCatch({  over\_index <- balls %/% 6 + 1  if (over\_index > length(jsonData$innings[[inn]]$overs)) {  flag <- FALSE  break  }  over\_balls <- length(jsonData$innings[[inn]]$overs[[over\_index]]$deliveries)  currball <- 1  extras\_count=0  while (currball <= over\_balls) {  ball\_index <- (balls %/% 6) \* 6 + currball # Calculate the current ball index  toss <- c(toss, jsonData$info$toss$winner)  match\_winner <- c(match\_winner, jsonData$info$outcome$winner)  batter <- jsonData$innings[[inn]]$overs[[over\_index]]$deliveries[[currball]]$batter  bowler <- jsonData$innings[[inn]]$overs[[over\_index]]$deliveries[[currball]]$bowler  runs <- jsonData$innings[[inn]]$overs[[over\_index]]$deliveries[[currball]]$runs$batter  team <- jsonData$innings[[inn]]$team  batter\_vector<-c(batter\_vector,batter)  baller\_vector <- c(baller\_vector, bowler)  balls\_vector <- c(balls\_vector, balls%/%6 + 0.1\*(currball-extras\_count))  cummulative\_score=cummulative\_score+jsonData$innings[[inn]]$overs[[over\_index]]$deliveries[[currball]]$runs$total  runsperball\_vector <- c(runsperball\_vector, runs)  country\_vector <- c(country\_vector, team)  Innings\_vector <- c(Innings\_vector, inn)  toss\_winner<-c(toss\_winner,toss)  toss\_decision <- c(toss\_decision, jsonData$info$toss$decision)  if (!is.null(jsonData$innings[[inn]]$overs[[over\_index]]$deliveries[[currball]]$extras)) {  if(is.null(jsonData$innings[[inn]]$overs[[over\_index]]$deliveries[[currball]]$extras$legbyes))  {  extras\_count=extras\_count+1  }  extra\_vector <- c(extra\_vector, TRUE)  } else {  extra\_vector <- c(extra\_vector, FALSE)  }  if (!is.null(jsonData$info$outcome$by$wickets)) {  margin <- c(margin, paste(jsonData$info$outcome$by$wickets, " wickets"))  } else {  margin <- c(margin, paste(jsonData$info$outcome$by$wickets, " runs"))  }  total\_score=c(total\_score,cummulative\_score)  if(!is.null(jsonData$innings[[inn]]$overs[[over\_index]]$deliveries[[currball]]$wickets)){  cum\_wickets=cum\_wickets+1  wickets=c(wickets,cum\_wickets)  }  else{  wickets=c(wickets,0)  }    currball <- currball + 1  }    balls <- balls + 6  }, error = function(e) {  flag <- FALSE  })  }  }  # print(total\_score)  # print(wickets)  df <- data.frame(  baller = baller\_vector,  batter = batter\_vector,  balls = balls\_vector,  runsperball = runsperball\_vector,  extra = extra\_vector,  total\_score=total\_score,  wickets=wickets,  country = country\_vector,  innings = Innings\_vector,  toss\_decision = toss\_decision,  toss\_winner=match\_winner,  margin=margin,  match\_winner = match\_winner  )  # write.csv(df,"/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/2003/Conversion\_demo.csv")  # Mushtaq's path  #write.csv(df,"C:\Users\janga\OneDrive\Desktop\Minor\_Project-II/Data/IND vs AUS/ODI/2003/Conversion\_demo.csv")  # print(df)  return( df)  }  # convert\_to\_csv(file\_path) |
| Output | A csv file |
|  |  |
| **Description of Task#2** | **Extraction of runs scored by each player** |
| Required Pre-processing | Generation of another csv |
| Attributes involved | Batter,balls,runs,Bowler,balls |
| R-Code with necessary comments | player\_scores<- function(df){  # Load required library  library(jsonlite)  library(hash)  # Read JSON data  # df <- read.csv("/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/2003/CSVs/Conversion\_demo.csv")  # df<- read.csv("D:/Minor\_Project-II/Data/IND vs AUS/ODI/2003/CSVs/Conversion\_demo.csv")  # Initialize hash  h <- hash()  h\_bowler <- hash()  prev\_inn=0  for(i in 1:nrow(df)) {  row\_data <- df[i,]    # Update runs for batter  if(!(row\_data$batter %in% names(h))) {    h[[row\_data$batter]] <- list(runs=row\_data$runsperball,out=0,balls=0,country=row\_data$country)  } else {  h[[row\_data$batter]][[1]] <- h[[row\_data$batter]][[1]] + row\_data$runsperball  }  if(!row\_data$extra){  h[[row\_data$batter]][[3]]=h[[row\_data$batter]][[3]]+1  }  # Update runs and wickets for bowler  prev\_score <- 0    if(i != 1 && row\_data$innings==prev\_inn) {  prev\_score <- df[i - 1,]$total\_score  }  prev\_inn=row\_data$innings  if(!(row\_data$baller %in% names(h\_bowler))) {  if(row\_data$country=="Australia"){  h\_bowler[[row\_data$baller]] <- list(score = 0, wickets = 0,extra=0,country="India")  }  else{  h\_bowler[[row\_data$baller]] <- list(score = 0, wickets = 0,extra=0,country="Australia")  }  }  if(row\_data$extra){  h\_bowler[[row\_data$baller]][[3]]=h\_bowler[[row\_data$baller]][[3]]+1  }  if(row\_data$wickets != 0) {  h\_bowler[[row\_data$baller]][[2]] <- h\_bowler[[row\_data$baller]][[2]] + 1  h[[row\_data$batter]][[2]]=1  }    h\_bowler[[row\_data$baller]][[1]] <- h\_bowler[[row\_data$baller]][[1]]+row\_data$total\_score - prev\_score  # print(h\_bowler)  }  h\_df <- as.data.frame(t(sapply(h, unlist)))  h\_df1 <- data.frame(player = rownames(h\_df), h\_df, row.names = NULL)  # write.csv(h\_df, "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/2003/players\_details.csv", row.names = FALSE)  # write.csv(h\_df, "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/BatsmanScores.csv", row.names = FALSE)  h\_df <- as.data.frame(t(sapply(h\_bowler, unlist)))  h\_df2 <- data.frame(player = rownames(h\_df), h\_df, row.names = NULL)  # write.csv(h\_df, "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/2003/players\_details.csv", row.names = FALSE)  # write.csv(h\_df, "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/BowlerStats.csv", row.names = FALSE)  return (list(h\_df1,h\_df2))  } |
| Output | CSV file |
|  |  |
| **Description of Task#3** | Extraction of wickets taken by the bowler |
| Required Pre-processing | Extract the performance of a bowler from the ball by ball data |
| Attributes involved | Bowler,balls,runs,wickets |
| R-Code with necessary comments | player\_scores<- function(df){  # Load required library  library(jsonlite)  library(hash)  # Read JSON data  # df <- read.csv("/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/2003/CSVs/Conversion\_demo.csv")  # df<- read.csv("D:/Minor\_Project-II/Data/IND vs AUS/ODI/2003/CSVs/Conversion\_demo.csv")  # Initialize hash  h <- hash()  h\_bowler <- hash()  prev\_inn=0  for(i in 1:nrow(df)) {  row\_data <- df[i,]    # Update runs for batter  if(!(row\_data$batter %in% names(h))) {    h[[row\_data$batter]] <- list(runs=row\_data$runsperball,out=0,balls=0,country=row\_data$country)  } else {  h[[row\_data$batter]][[1]] <- h[[row\_data$batter]][[1]] + row\_data$runsperball  }  if(!row\_data$extra){  h[[row\_data$batter]][[3]]=h[[row\_data$batter]][[3]]+1  }  # Update runs and wickets for bowler  prev\_score <- 0    if(i != 1 && row\_data$innings==prev\_inn) {  prev\_score <- df[i - 1,]$total\_score  }  prev\_inn=row\_data$innings  if(!(row\_data$baller %in% names(h\_bowler))) {  if(row\_data$country=="Australia"){  h\_bowler[[row\_data$baller]] <- list(score = 0, wickets = 0,extra=0,country="India")  }  else{  h\_bowler[[row\_data$baller]] <- list(score = 0, wickets = 0,extra=0,country="Australia")  }  }  if(row\_data$extra){  h\_bowler[[row\_data$baller]][[3]]=h\_bowler[[row\_data$baller]][[3]]+1  }  if(row\_data$wickets != 0) {  h\_bowler[[row\_data$baller]][[2]] <- h\_bowler[[row\_data$baller]][[2]] + 1  h[[row\_data$batter]][[2]]=1  }    h\_bowler[[row\_data$baller]][[1]] <- h\_bowler[[row\_data$baller]][[1]]+row\_data$total\_score - prev\_score  # print(h\_bowler)  }  h\_df <- as.data.frame(t(sapply(h, unlist)))  h\_df1 <- data.frame(player = rownames(h\_df), h\_df, row.names = NULL)  # write.csv(h\_df, "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/2003/players\_details.csv", row.names = FALSE)  # write.csv(h\_df, "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/BatsmanScores.csv", row.names = FALSE)  h\_df <- as.data.frame(t(sapply(h\_bowler, unlist)))  h\_df2 <- data.frame(player = rownames(h\_df), h\_df, row.names = NULL)  # write.csv(h\_df, "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/2003/players\_details.csv", row.names = FALSE)  # write.csv(h\_df, "/Users/morampudigopiprashanthraju/Desktop/DataScience/Minor\_Project-II/Data/IND vs AUS/ODI/BowlerStats.csv", row.names = FALSE)  return (list(h\_df1,h\_df2))  } |
| Output | Csv file with bowler stats |
|  |  |
| **Description of Task#4** | Extraction of player details through API’s |
| Required Pre-processing | Genertation of all the players details involved in the match |
| Attributes involved | Player,Country,Batting-hand,Bowling-Style |
| R-Code with necessary comments | library(jsonlite)  base\_dir <- "D:/Minor\_Project-II/Data/IND vs AUS/ODI"  year\_folders <- list.files(base\_dir)  #Reading the Aussies's CSV file  players\_details <- read.csv("D:/Minor\_Project-II/Data/IND vs AUS/ODI/players\_details.csv")  for (year in year\_folders){  json\_files <- list.files(paste(base\_dir, year, sep = "/"))  for (json\_file in json\_files){  if (json\_file == "CSVs"){  next()  }  file\_path <- paste(base\_dir, year, json\_file, sep = "/")  print(file\_path)  json\_data <- read\_json(path=file\_path)  aplayers <- json\_data$info$players$Australia  for (player in aplayers){  if (player %in% players\_details$player){  next()  }  else{  players\_details <- rbind(players\_details, data.frame(player=player, country="Australia", batting="None", bowling\_style="None"))  }  }  iplayers <- json\_data$info$players$India  for (player in iplayers){  if (player %in% players\_details$player){  next()  }  else{  players\_details <- rbind(players\_details, data.frame(player=player, country="India", batting="None", bowling\_style="None"))  }  }  }  }  write.csv(players\_details,"D:/Minor\_Project-II/Data/IND vs AUS/ODI/players\_details.csv")  # # write(aus\_players, file = file\_path) |
| Output | A csv file with the details |
|  |  |
| **Description of Task#5** | Plotting of a Kernel Density Plot |
| Required Pre-processing | We had to preprocess the BatsmanScores.csv and BowlerStats.csv and add new fields in like Batting\_Avg, Strike\_Rate and Bowling\_Avg, Eonomy in respective files. |
| Attributes involved | Player, Batting\_Avg, Strike\_Rate for the batsman analysis and Player, Bowling\_Avg, Economy for the bowler analysis |
| R-Code with necessary comments | library(ggplot2)  #For the batsman analysis  data <- read.csv("D:/Minor\_Project-II/Data/IND vs AUS/ODI/BatsmanScores.csv")  aus\_players\_strike\_rate <- data.frame( strike\_rate = numeric())  aus\_players\_batting\_avg <- data.frame( batting\_avg = numeric())  ind\_players\_strike\_rate <- data.frame(strike\_rate = numeric())  ind\_players\_batting\_avg <- data.frame(batting\_avg = numeric())  for (i in 1:nrow(data)) {  if (data$country[i] == "Australia") {  aus\_players\_strike\_rate <- rbind(aus\_players\_strike\_rate, data.frame(strike\_rate = data$strike\_rate[i]))  aus\_players\_batting\_avg <- rbind(aus\_players\_batting\_avg, data.frame(batting\_avg = data$batting\_average[i]))  }  else{  ind\_players\_strike\_rate <- rbind(ind\_players\_strike\_rate, data.frame(strike\_rate = data$strike\_rate[i]))  ind\_players\_batting\_avg <- rbind(ind\_players\_batting\_avg, data.frame(batting\_avg = data$batting\_average[i]))  }  }  aus\_players\_strike\_rate$Group <- "Australia"  ind\_players\_strike\_rate$Group <- "India"  combined\_data <- rbind( aus\_players\_strike\_rate, ind\_players\_strike\_rate)  # Create a KDE plot of strike rates  strike\_rate\_plot<-ggplot(combined\_data, aes(x = strike\_rate, fill = Group, color = Group)) +  geom\_density(alpha = 0.7) +  labs(title = "Kernel Density Estimation Plot of Strike Rates",  x = "Strike Rate",  y = "Density") +  scale\_fill\_manual(values = c("yellow","blue")) +  scale\_color\_manual(values = c("yellow","blue")) +  theme\_minimal()  aus\_players\_batting\_avg$Group <- "Australia"  ind\_players\_batting\_avg$Group <- "India"  combined\_data <- rbind( aus\_players\_batting\_avg, ind\_players\_batting\_avg)  # Create a KDE plot of strike rates  batting\_avg\_plot<-ggplot(combined\_data, aes(x = batting\_avg, fill = Group, color = Group)) +  geom\_density(alpha = 0.7) +  labs(title = "Kernel Density Estimation Plot of Strike Rates",  x = "Batting Average",  y = "Density") +  scale\_fill\_manual(values = c("yellow","blue")) +  scale\_color\_manual(values = c("yellow","blue")) +  theme\_minimal()  library(gridExtra)  grid.arrange(strike\_rate\_plot, batting\_avg\_plot, nrow = 1)  #For the bowler analysis  data <- read.csv("D:/Minor\_Project-II/Data/IND vs AUS/ODI/BowlerStats.csv")  aus\_players\_economy <- data.frame( economy = numeric())  aus\_players\_bowling\_avg <- data.frame( bowling\_avg = numeric())  ind\_players\_economy <- data.frame(economy = numeric())  ind\_players\_bowling\_avg <- data.frame(bowling\_avg = numeric())  for (i in 1:nrow(data)) {  if (data$country[i] == "Australia") {  aus\_players\_economy <- rbind(aus\_players\_economy, data.frame(economy = data$economy[i]))  aus\_players\_bowling\_avg <- rbind(aus\_players\_bowling\_avg, data.frame(bowling\_avg = data$bowling\_average[i]))  }  else{  ind\_players\_economy <- rbind(ind\_players\_economy, data.frame(economy = data$economy[i]))  ind\_players\_bowling\_avg <- rbind(ind\_players\_bowling\_avg, data.frame(bowling\_avg = data$bowling\_average[i]))  }  }  aus\_players\_economy$Group <- "Australia"  ind\_players\_economy$Group <- "India"  combined\_data <- rbind( aus\_players\_economy, ind\_players\_economy)  # Create a KDE plot of strike rates  economy\_plot<-ggplot(combined\_data, aes(x = economy, fill = Group, color = Group)) +  geom\_density(alpha = 0.7) +  labs(title = "Kernel Density Estimation Plot of Economies",  x = "Economy",  y = "Density") +  scale\_fill\_manual(values = c("yellow","blue")) +  scale\_color\_manual(values = c("yellow","blue")) +  theme\_minimal()  aus\_players\_bowling\_avg$Group <- "Australia"  ind\_players\_bowling\_avg$Group <- "India"  combined\_data <- rbind( aus\_players\_bowling\_avg, ind\_players\_bowling\_avg)  # Create a KDE plot of strike rates  bowling\_avg\_plot<-ggplot(combined\_data, aes(x = bowling\_avg, fill = Group, color = Group)) +  geom\_density(alpha = 0.7) +  labs(title = "Kernel Density Estimation Plot of Bowling Averages",  x = "Bowling Average",  y = "Density") +  scale\_fill\_manual(values = c("yellow","blue")) +  scale\_color\_manual(values = c("yellow","blue")) +  theme\_minimal()  library(gridExtra)  grid.arrange(economy\_plot, bowling\_avg\_plot, nrow = 1) |
| Output | # Load necessary libraries  library(ggplot2)  # library(dplyr)  library(plotly)  # Read the dataset  data <- read.csv("D:/Minor\_Project-II/Data/IND vs AUS/ODI/BowlerStats.csv")  print(data)  # 1. Violin Plot of Batting Averages and Bowling Averages  p <- ggplot(data, aes(x = country, y = strike\_rate, fill = country)) +  geom\_violin(trim = FALSE) +  labs(title = "Distribution of Strike Rates by Country") +  scale\_fill\_manual(values = c("Australia" = "yellow", "India" = "blue")) + # Customize fill colors  theme\_minimal()  print(p)  # ggplot(data, aes(x = country, y = economy, fill = country)) +  # geom\_violin(trim = FALSE) +  # labs(title = "Distribution of Ecomonies by Country") +  # theme\_minimal()  # 2. Heatmap of Batting and Bowling Performance Metrics  heatmap\_data <- select(data, player, runs, out, balls, country, batting\_average, strike\_rate)  # # Convert the data to a dataframe  heatmap\_data <- as.data.frame(heatmap\_data)  # # Set the 'player' column as row names  rownames(heatmap\_data) <- heatmap\_data$player  heatmap\_data$player <- NULL  # # Convert any character columns to numeric (if needed)  heatmap\_data <- mutate\_if(heatmap\_data, is.character, as.numeric)  # # Plot the heatmap  heatmap(as.matrix(heatmap\_data), Rowv = NA, Colv = NA, col = cm.colors(256), scale = "column", margins = c(5, 10))  # 3. Joint Distribution Plot of Batting and strike rates  ggplot(data, aes(x = bowling\_average, y = economy, color = country)) +  geom\_point() +  geom\_smooth(method = "lm", se = FALSE) +  labs(title = "Joint Distribution of Bowling Averages and Economies") +  scale\_color\_manual(values = c("India" = "blue", "Australia" = "yellow")) + # Differentiate colors for both teams  theme\_minimal()  # # 4. Clustered Bar Chart of Player Statistics by Country  # Create a ggplot object  p <- ggplot(data, aes(x = player, y = wickets, fill = country)) +  geom\_bar(stat = "identity", position = "dodge") +  labs(title = "Wickets Taken by Player and Country", x = "Player", y = "Runs") +  scale\_fill\_manual(values = c("Australia" = "yellow", "India" = "blue")) +  theme\_minimal()  Add axis labels  # Convert the ggplot object to a plotly object  p <- ggplotly(p, tooltip = c("player", "wickets")) # Set tooltip to show player's name and runs  print(p)  # # 5. Interactive Bubble Chart of Player Performance Metrics  plot\_ly(data, x = ~score, y = ~wickets, size = ~economy,  color = ~country, text = ~player, hoverinfo = "text",  mode = "markers", type = "scatter") %>%  layout(title = "Bubble Chart of Bowler Performance Metrics",  xaxis = list(title = "Runs Conceded"),  yaxis = list(title = "wickets"),  showlegend = TRUE) |
|  |  |
| **Description of Task#6** |  |
| Required Pre-processing |  |
| Attributes involved |  |
| R-Code with necessary comments |  |
| Output |  |
|  |  |
| **List of Websites referred** | Cricsheet.com for datasets and many random websites for formulae for calculations in the codes. |
| **Details of Relevant Research papers referred, please present in tabular form the below details for each referred paper (minimum 10 – 12 papers):**   1. Paper Title 2. URL 3. Year of publication 4. Conference / Journal details 5. Dataset used 6. Description of the dataset 7. URL of the dataset 8. Tasks Carried out. 9. Algorithms used for each task 10. Results obtained for each Task as reported in the paper. 11. Gaps reported in the paper | |  |  | | --- | --- | | Paper Title | Predicting the Outcome of ODI Cricket Matches: A Team Composition Based Approach | | Year of publication | 2016 | | Conference details | IEEE | | Dataset used | **Source:** ESPN Cricinfo [reference 13]  **Timeframe:** 2010-2014 ODI matches (excluding rain-interrupted/tied matches) | | Description of the dataset | **Team data**  Teams competing  Toss winner.  **Venue**  **Player data (for all participating players in each match)**  Career statistics (matches played, batting innings, bowling innings, etc.)  Recent performance (last 4 matches) | | Tasks | Predicting the winner of an ODI cricket match | | Algorithms used | **Batsmen Score:**  Combines career statistics (batting average, centuries, fifties) and recent form (average runs in last 4 matches)  **Bowler Score:**  Combines career statistics (bowling average, wickets taken, economy)  Does not consider recent performance due to lack of data.  **Team Strength**  **Relative Strength (S(A/B)):** Batting Strength of Team A \* Bowling Strength of Team B - Batting Strength of Team B \* Bowling Strength of Team A (where Team A and B are the competing teams) | | Results obtained | Achieved better accuracy than other binary classifiers (SVM, Random Forest, Logistic Regression, Decision Tree)  Outperformed two baseline models:  Model 1: Team winning the toss wins the match (56% accuracy)  Model 2: Team with higher relative strength wins the match (63% accuracy)  Achieved team-wise winning accuracy between 68% and 70% (comparable to the best reported accuracy in the literature) | | Gaps reported | Dataset limitations  Lacks data on external factors like weather and ground design  Player performance data only considers matches played between 2010-2014.  The inability to compare the model against previous models due to different underlying datasets used (e.g., no data on day/night matches or in-match state) |  |  |  | | --- | --- | | Paper Title | Predicting Optimal Cricket Team using Data Analysis | | Year of publication | 2021 | | Conference details | 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) | | Dataset used | Not specified | | Description of the dataset | Cricket player and team statistics  Includes batting, bowling, and wicket-keeping parameters | | Tasks | **Data Collection:**  Gathered data on player and team statistics from various sources (Kaggle, ESPNcricinfo, Howstat, Cricmetric)  **Data Preprocessing:**  Defined parameters for batting, bowling, all-rounders, and wicketkeepers.  Calculated derived attributes (consistency, hard-hitting ability) for deeper insights. | | Algorithms used | Players are selected based on the highest batting average. | | Results obtained | Assigned weights to different player attributes based on venue and importance.  Developed formulae to calculate a resultant score for players on different pitches (e.g., Bengaluru, Chennai, Mumbai). | | Gaps reported | Specific details and size of the dataset are not available.  Algorithms used for weight calculation and resultant scores are not mentioned.  Performance metrics (e.g., MAE) to evaluate the model are not reported.  Ignores factors like player form, injuries, and team dynamics. |  |  |  | | --- | --- | | Paper Title | Cricket Player Profiling: Unraveling Strengths and Weaknesses Using Text Commentary Data | | Year of publication | 2023 | | Conference details | IEEE | | Dataset used | Text commentary of T20 International matches (2008-2019) | | Description of the dataset | Text commentary data containing information on batsman names, scores, dismissals (if any), type of delivery (length), and where the ball was hit on the field (ground region). Source: ESPN Cricinfo (not explicitly mentioned in the summary, but implied) | | Tasks | Identify batsman strengths and weaknesses.  Calculate region-wise strike rate | | Algorithms used | The paper mentions using regular expressions to identify patterns in the commentary text. However, the specific algorithms used for data processing and analysis aren't explicitly mentioned in the summary. | | Results obtained | Specific results related to batsman strengths and weaknesses or strike rates by region aren't provided in the summary. | | Gaps reported | The paper suggests limitations in the dataset as it only focuses on T20 Internationals.  The authors propose extending the analysis to include other cricket formats (Test and ODI).  They also suggest calculating bowler-specific strike rates for batsmen. |  |  |  | | --- | --- | | Paper Title | Cricket Team Prediction Using Machine Learning Techniques | | Year of publication | 2011 | | Conference details | Fr. Conceicao Rodrigues College of Engineering, Mumbai, Maharashtra, India | | Dataset used | Not explicitly mentioned, but authors suggest scraping data from websites like ESPN Cricinfo. | | Description of the dataset | Cricket player statistics including batting and bowling averages, strike rates, etc. | | Tasks | Predict the best 11 players for a team | | Algorithms used | Data scraping (web scraping)  Random Forest Classifier | | Results obtained | The system outputs a recommended team of 11 players based on their historical data and factors like the opposition team and venue. | | Gaps reported | **Data limitations:**  Authors acknowledge the lack of data on factors like weather and ground design.  Dataset used only considers historical data up to 2017.  **Model improvements:**  The paper proposes incorporating additional parameters like player fitness and location-based performance.  Experimenting with other machine learning algorithms like XGBoost is suggested.. |  |  |  | | --- | --- | | Paper Title | Analysis and Winner Prediction of Cricket Match | | Year of publication | 2022 | | Conference details | International Journal of Research Publication and Reviews (Volume 3, Issue 4, pp 1626-1632) | | Dataset used | Not specified in the paper | | Description of the dataset | IPL match data including details like teams, toss winners, scores, and winners.  Specific features used for modelling are not mentioned. | | Tasks | **Task 1: Exploratory Data Analysis (EDA)**  Analyzed win/loss rate based on toss win.  Compared win percentage of different teams across venues.  **Task 2:** Winner Prediction - Score based method | | Algorithms used | Linear Regression  Random Forest Regression  Decision Tree | | Results obtained | Random Forest achieved the highest R-squared score (0.7516) for score-based prediction.  Random Forest achieved the highest accuracy (88.23%) for toss-based prediction. | | Gaps reported | Specific features employed for model training are not mentioned.  Performance metrics (MAE) are only reported for score-based prediction using linear regression.  The paper doesn't discuss limitations of the models or potential improvements. |  |  |  | | --- | --- | | Paper Title | Enhancing Cricket Performance Analysis with Human Pose Estimation and Machine Learning | | Year of publication | 2023 | | Conference details | Sensors (MDPI) [Journal] | | Dataset used | Cricket video dataset | | Description of the dataset | Contains videos featuring eight different batsman strokes: pull, cut, cover drive, straight drive, backfoot punch, on drive, flick, and sweep.  Each video frame has 17 key points extracted related to the batsman's body pose using MediaPipe library. | | Tasks | **Task:** Stroke prediction for batsman | | Algorithms used | Random Forest (RF)  Support Vector Machine (SVM)  K-Nearest Neighbors (KNN)  Decision Tree (DT)  Logistic Regression (LR)  Long Short-Term Memory (LSTM) | | Results obtained | Random Forest achieved the highest accuracy (99.77%) for stroke prediction.  k-fold validation of the RF model yielded 95.0% accuracy with a standard deviation of 0.07. | | Gaps reported | The paper doesn't mention the size of the video dataset.  The dataset itself is not publicly available. |  |  |  | | --- | --- | | Paper Title | Analysis and Winner Prediction of Cricket Match | | Year of publication | 2022 | | Conference details | International Journal of Research Publication and Reviews (Volume 3, Issue 4, pp 1626-1632) | | Dataset used | Not specified in the paper | | Description of the dataset | IPL match data including details like teams, toss winners, scores, and winners.  Specific features used for modelling are not mentioned. | | Tasks | **Task 1: Exploratory Data Analysis (EDA)**  Analyzed win/loss rate based on toss win.  Compared win percentage of different teams across venues.  **Task 2:** Winner Prediction - Score based method | | Algorithms used | Linear Regression  Random Forest Regression  Decision Tree | | Results obtained | Random Forest achieved the highest R-squared score (0.7516) for score-based prediction.  Random Forest achieved the highest accuracy (88.23%) for toss-based prediction. | | Gaps reported | Specific features employed for model training are not mentioned.  Performance metrics (MAE) are only reported for score-based prediction using linear regression.  The paper doesn't discuss limitations of the models or potential improvements. |  |  |  | | --- | --- | | Paper Title | Extracted Information on IPL Match Result Prediction (Praveen Banasode et al., 2021) | | Year of publication | 2020 | | Conference details | IOP Conference Series: Materials Science and Engineering (2021) | | Dataset used | **Source:** Scraped data from https://stats.espncricinfo.com (2008-2019 seasons)  **Accessed through:** https://kaggle.com [mentioned but not explicitly stated as the source] | | Description of the dataset | Match data (win/loss, toss winner, venue, etc.)  Player data (runs scored, bowling economy, wickets taken, etc.) | | Tasks | **Data Collection:** Gather past IPL match data.  Data Preprocessing: Not explicitly mentioned, but likely cleaning and handling missing values (if any).  **Player Performance Analysis:**  Calculate individual player stats (runs scored, bowling economy, etc.) for each season and across all seasons.  Identify player strengths and weaknesses (e.g., favourite venues, performance against specific teams/bowlers).  **Team Analysis:** Analyse team performance (win/loss) considering factors like toss outcome and batting first.  **Match Result Prediction:** Not a core focus, but the analysis provides insights that could be used for prediction (e.g., win probability based on toss and batting order). | | Algorithms used | **Tasks 1 & 2:** Not mentioned.  **Tasks 3 & 4:** Likely basic statistical functions and calculations (e.g., sum, average). | | Results obtained | **Data Collection:** 756 rows (matches) and 18 data attributes collected for the 2008-2019 seasons.  **Player Performance Analysis:**  Insights on individual player performance across various metrics (runs scored, bowling economy, wicket types).  Identification of player strengths against specific teams/bowlers and preferred venues.  **Team Analysis:** Findings on toss win impact on match outcome and batting first advantage (higher win percentage). | | Gaps reported | The specific algorithms used for data processing and analysis are not mentioned.  The paper doesn't explicitly address how the collected data is used for match result prediction.  Factors beyond historical data (e.g., player injuries, team morale) are not considered for prediction. |  |  |  | | --- | --- | | Paper Title | Analysing the performance of the Indian Cricket Team using Weighted Association Rule Mining | | Year of publication | 2018 | | Conference details | 2018 International Conference on Computing, Power and Communication Technologies (GUCON) | | Dataset used | ODI cricket matches played by India against Sri Lanka and South Africa from 2008 to 2017 | | Description of the dataset | **Source:** Data likely collected from ESPNcricinfo (https://www.espncricinfo.com/) (URL not explicitly mentioned in the paper).  **Type:** Transactional data  **Features:**  **Match-related attributes:** Toss result (Win/Loss), Ground condition (Home/Away), Innings order (Bat first/Second), Match outcome (Win/Loss) | | Tasks | **Data Collection:** ODI match data for India, Sri Lanka, and South Africa (2008-2017).  Data Preprocessing: (Not explicitly mentioned, but likely cleaning and handling missing values if any).  **Association Rule Mining:** Identify frequent patterns between match attributes and outcome.  **Weight Assignment:** Experts assign weights to attribute-value pairs based on perceived significance.  **Weighted Association Rule Evaluation:** Calculate Weighted Support and Confidence to evaluate the strength of the association rules. | | Algorithms used | **Task 1 & 2:** Not mentioned in the paper.  **Task 3:** Association rule mining algorithm (specific algorithm not mentioned).  **Task 4 & 5:** Not applicable (expert weighting and calculations). | | Results obtained | **Data Collection:** Specific details not provided.  **Data Preprocessing**: Not mentioned.  **Association Rule Mining:** Not explicitly presented, but the paper mentions identifying patterns like "toss win, bat first at home" potentially leading to higher win probability.  **Weight Assignment:** Weights assigned to attributes (e.g., toss win might be more important than batting first) - specific values not provided.  **Weighted Association Rule Evaluation:** The paper discusses findings based on weighted scores, but specific values are not presented. | | Gaps reported | Specific association rules and their weighted scores are not provided.  The method for selecting experts and resolving weight disagreements is not mentioned.  The study doesn't consider the impact of team composition (e.g., player form, injuries).  The data is limited to matches against two teams during a specific period. |  |  |  | | --- | --- | | Paper Title | A DEA Model for Selection of Indian Cricket Team Players | | Year of publication | 2019 | | Conference details | IEEE (Institute of Electrical and Electronics Engineers) | | Dataset used | Cricket player statistics for batting and bowling performance. | | Description of the dataset | Likely includes individual player data for a specific period (not mentioned in the paper).  **Batting parameters:** Runs scored, Strike Rate, Batting Average, Centuries (100s), Fifties (50s), Boundaries (4s & 6s), Sixes (6s).  **Bowling parameters:** Overs bowled, Maiden Overs (Mdns), Runs Conceded, Wickets Taken, Bowling Average, Economy Rate, Strike Rate, Five-wicket hauls (5s) & Ten-wicket hauls (10s) (for Test matches). | | Tasks | **Data Collection:** Batting and bowling statistics of Indian cricket players were collected, possibly from sources like ESPNcricinfo (https://www.espncricinfo.com/). The paper doesn't mention the specific source of the data.  **Data Preprocessing:** The data was likely preprocessed to ensure consistency and handle missing values (if any). The paper doesn't elaborate on the preprocessing steps.  Efficiency Score Calculation: Data Envelopment Analysis (DEA) was used to calculate efficiency scores for each player based on batting and bowling parameters.  **Inputs:** Runs, Strike Rate, Batting Average, etc. for batsmen; Overs, Maidens, Runs Conceded, Wickets Taken, etc. for bowlers.  **Outputs:** Not explicitly mentioned in the paper, but likely a weighted sum of these parameters.  **Team Selection:** Players were selected for a test match based on their DEA efficiency scores, with some consideration for expert advice (e.g., bowler with better batting strike rate being preferred). | | Algorithms used | **Task 1 & 2:** Not mentioned in the paper.  **Task 3:** Data Envelopment Analysis (DEA) | | Results obtained | DEA efficiency scores were obtained for each player.  The team selected based on the DEA model was like the team selected by the Board of Control for Cricket in India (BCCI) for a test match series. | | Gaps reported | The paper doesn't specify the exact source of the data or the details of the preprocessing steps.  The specific DEA model formulation (e.g., input/output weights) is not provided.  The weightage given to expert advice in team selection is not quantified. | |

NOTE: The content that you submit can be used for your minor project documentation.