M.Sc Project
Report

To study the ODI cricket performance of players in different countries (2019-2023)

Submitted in partial fulfillment of
the requirements for the award of the degree of

Master of Science
in
Statistics

Submitted by

Roll Numbers	Names of Students
P03NK21S0417	Mr, DIVAKARA, K,N.
P03NK21S0456	Mr, LOHITH, B,N.
P03NK21S0766	Ms, VANDANAPRIYA, M.

Under the guidance of

Dr. Suresh, R.

Department of Statistics
Bangalore University
Bengaluru, Karnataka, India – 560056

Department of Statistics

Bangalore University

Certificate

This is to certify that this is a bonafide record of the project presented by the student whose name is given below in the year 2023 in partial fulfillment of the requirements of the Master degree of Statistics.

Roll Numbers	**Names of Students**
P03NK21S0417	Mr DIVAKARA, K,N.
P03NK21S0456	Mr LOHITH, B,N.
P03NK21S0766	Ms VANDANAPRIYA, M.

Project Guide

Chairperson

Declaration

We here by declare that the matter embodied in this project entitled ”To study the ODI cricket performance of players in different countries (2019-2023)” submitted to the Department of Statistics, Bangalore University in partial fulfillment of the requirements for the award of Master’s Degree in Statistics, is the result of our studies and this project has been composed by us under the guidance and supervision of Dr. Suresh, R., Assistant Professor, Department of Statistics, Bangalore University, during 2022-2023.

we also declare that this project has not been previously formed the basis for the award of any degree, diploma, associateship, fellowship, etc. of any university or institution.

Place: Bengaluru
Date:

Mr Divakara, K,N.
Mr Lohith, B,N.
Ms Vandanapriya, M.

Acknowledgments

It gives great pleasure to express my sincere gratitude to my project guide **Dr. Suresh, R**. Assistant Professor, Department of Statistics, Bangalore University, Bangalore, with whose help and supervision the present work has been carried out.

I profoundly thank Bangalore University and the chairperson **Professor Dr. Parmeshwar V Pandit**, Department of Statistics for providing the necessary facilities to carry out this project.

I wish to thank all my teachers for their encouragement and support.

I thank to the non-teaching staff of the Department of Statistics for their help and co-operation in carrying out this work.

Words are inadequate to express my gratitude to all the respondents who have enormously helped in providing the required data.

I also convey my heartful gratitude to my parents and friends for the love, encouragement and support provided to me in the course of studies. I thankful to all people involved directly or indirectly without whose affection and help this work would have been impossible.

Mr, DIVAKARA, K,N.
Mr, LOHITH, B,N.
Miss, VANDANAPRIYA, M.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is Cricket?

Cricket is believed to have begun possibly as early as the 13th century as a game in which country boys bowled at a tree stump or at the hurdle gate into a sheep pen. This gate consisted of two uprights and a crossbar resting on the slotted tops; the crossbar was called a bail and the entire gate a wicket. The fact that the bail could be dislodged when the wicket was struck made this preferable to the stump, which name was later applied to the hurdle uprights. Early manuscripts differ about the size of the wicket, which acquired a third stump in the 1770s, but by 1706 the pitch—the area between the wickets—was 22 yards long. The ball, once presumably a stone, has remained much the same since the 17th century. Its modern weight of between 5.5 and 5.75 ounces (156 and 163 grams) was established in 1774.
Cricket is played with a bat and ball and involves two competing sides (teams) of 11 players. The field is oval with a rectangular area in the middle, known as the pitch, that is 22 yards (20.12 metres) by 10 feet (3.04 metres) wide. Two sets of three sticks, called wickets, are set in the ground at each end of the pitch. Across the top of each wicket lie horizontal pieces called bails. The sides take turns at batting and bowling (pitching); each turn is called an "innings" (always plural). Sides have one or two innings each, depending on the prearranged duration of the match, the object being to score the most runs. The bowlers, delivering the ball with a straight arm, try to break (hit) the wicket with the ball so that the bails fall. This is one of several ways that the batsman is dismissed, or put out. A bowler delivers six balls at one wicket (thus completing an "over"), then a different player from his side bowls six balls to the opposite wicket. The batting side defends its wicket..
There are two batters up at a time, and the batsman being bowled to (the

striker) tries to hit the ball away from the wicket. A hit may be defensive or offensive. A defensive hit may protect the wicket but leave the batsmen no time to run to the opposite wicket. In that case the batsmen need not run, and play will resume with another bowl. If the batsman can make an offensive hit, he and the second batsman (the non-striker) at the other wicket change places. Each time both batsmen can reach the opposite wicket, one run is scored. Providing they have enough time without being caught out and dismissed, the batsmen may continue to cross back and forth between the wickets, earning an additional run for each time both reach the opposite side. There is an outside boundary around the cricket field. A ball hit to or beyond the boundary scores four points if it hits the ground and then reaches the boundary, six points if it reaches the boundary from the air (a fly ball). The team with the highest number of runs wins a match. Should both teams be unable to complete their number of innings before the time allotted, the match is declared a draw. Scores in the hundreds are common in cricket.

Matches in cricket can range from informal weekend afternoon encounters on village greens to top-level international contests spread over five days in Test matches and played by leading professional players in grand stadiums. A One Day International (ODI) is a form of limited overs cricket, played between two teams with international status, in which each team faces a fixed number of overs, currently 50, with the game lasting up to 9 hours. The first ever international cricket game was between the US and Canada in 1844. The match was played at the grounds of the St George's Cricket Club in New York.

1.1.1 Introduction of Cricket in India:

Cricket was introduced to the Indian subcontinent by British sailors in the 18th century, and the first cricket club was established in 1792. India's men's national cricket team played its first international match on 25 June 1932 in a Lord's Test against England becoming the sixth team to be granted Test cricket status. India had to wait until 1952, almost twenty years, for its first Test victory. In its first fifty years of international cricket, success was limited, with only 35 wins in 196 Tests. The team, however, gained strength in the 1970s with the emergence of players like Sunil Gavaskar, Gundappa Viswanath, Kapil Dev, and the Indian spin quartet.

1.1.2 About Indian Men's Cricket Team:

In men's limited-overs cricket, India made its ODI and T20I debuts in 1974 and 2006, respectively. The team has won five major ICC tournaments, winning the Cricket World Cup twice (1983 and 2011), the ICC T20 World Cup once (2007) and the ICC Champions Trophy twice (2002 and 2013) and have also finished as runners-up in the World Cup once (2003), the T20 World Cup once (2014), and the Champions Trophy twice (2000 and 2017). The team were also part of ICC World Test Championship finals in the first two editions (2019–21, 2021–23). It was the second team after the West Indies to win the World Cup and the first team to win the World Cup on home soil after winning the 2011 Cricket World Cup.They have also won the Asia Cup seven times, in 1984, 1988, 1990–91, 1995, 2010, 2016 and 2018, whilst finishing runners-up thrice (1997, 2004, 2008). The team also won the 1985 World Championship of Cricket, defeating Pakistan in the final. Other achievements include winning the ICC Test Championship Mace five times and the ICC ODI Championship Shield once. The team is currently(25-07-2023) ranked third in ICC Men's Rankings with a total of points 3,807 and a rating of 115.

1.1.3 About Indian Womens Cricket Team:

The Indian women's national cricket team, also known as the Women in Blue, represents the country of India in international women's cricket. The team is currently(25/07/2023) ranked fourth in ICC Women's Rankings with a total of 7,662 points and a rating of 111. The Indian team is one of the top teams today in Asia and the world. It has shown tremendous improvements in the last few years and has proved their mettle on various occasions. Women's Cricket Association of India was formed in 1973. Indian women's cricket team played their first Test match against West Indies in 1976 at M.Chinnaswamy Stadium, Bangalore. Their first Test victory came two years later against the same opponent they debuted at the Moin-ul-Haq Stadium in Patna. Indian team played their first ODI against England in 1978 at Eden Gardens, Kolkata. Their first T20I match was also against England in 2006 at the County Cricket Ground, Derby. Women's Cricket Association of India was merged with the Board of Control for Cricket in India (BCCI) in 2006. India has reached the final of ICC World Cup twice in 2005 and 2017 but failed to get hold of the trophy in both the cases. India has made to the semi-finals on three other occasions, in 1997, 2000, and 2009. India also reached to the semi-finals of the first two editions of ICC Women's World Twenty20 in 2009 and 2010.

1.1.4 Review of the Literature

An extensive online search produced very few articles related to player's performance in the game of cricket. Here we discuss the role of machine learning tools in the analysis of the performances of players in the ODI format of Cricket. In Our analysis uses principle component analysis to rank the players based on their performance attributes like runs, average, strike rate, wickets, economy, etc. Grouping players based on similar characteristics and playing styles by using cluster analysis and classifying the players as batsmen bowlers and all-rounders with the help of classification techniques(Random Forest, Neural Networks, Support vector machine) then by using these techniques we determine which features have impact more on players performance.

1.1.5 Preliminary work done by investigators

- The main reason behind choosing this topic is how the players performs in ODI format comparing to other formats.
- The topic provides information about the how will players perform in upcoming years based on their past years performance.

1.1.6 Objectives

- To rank the batters and bowlers.
- Grouping players based on their performance attributes.
- To discriminate and classify players as all-rounders, batters, or bowlers based on their performance attributes.
- To compare the overall performance of a player across different continents.
- To compare the overall performance of a player across asia.
- To predict the outcome of an ODI cricket match using machine learning techniques.
- To analyze the association between match result and other attributes by calculating odds ratios.

1.1.7 Research Methodology

Study design:
The study design being used are:

DESCRIPTIVE DESIGNS – Performing analyses is crucial, to gaining an understanding of the distribution and traits of the data. It involves calculating statistics like the median standard deviation and range, for relevant variables.

OBSERVATIONAL ANALYTICAL DESIGNS – The data have been collected from www.espncricinfo.com and www.cricmetric.com.

Sampling: The sample size for the very project is 135. The sample size so chosen such that it is adequate to apply all relevant statistical techniques so that the sampling errors can be reduced as much as possible. Variables used are:

1. Independent Variables – Innings,runs,balls faced,outs,average,strike-rate,highest score,fifties,hundreds,fours,sixes,dot percentage,wickets,economy,bowling average,bowling strike rate,fifers,bowling dot-percentage,catches,stumping.

2. Dependent Variable – Player Type.

Data collection: Out of total of 135 data, 70 were collected from www.espncricinfo.com and 65 were collected from www.cricmetric.com.
Data analysis: The statistical method used are machine learning(supervised and unsupervised). The software used are R software,python.

1.1.8 Data description

This project is carried out using secondary data. The dataset provides information on overall batting, bowling and fielding details of 15 players each from 9 countries- India, Australia, New Zealand, West Indies, England, South Africa, Sri Lanka, Bangladesh, and Pakistan played from 2019-2023. The details are collected from www.espncricinfo.com and www.cricmetric.com . To study the batting and bowling performance of players, important measures of batting statistics such as innings, runs, balls faced, outs, average, strike rate, highest score, 50's, 100's, 4's, 6's, and dot percentage are considered. Similarly bowling statistics such as innings, overs, runs, wickets, economy, average, strike rate, 5W haul, best bowling innings, 4's, 6's, dot percentage. And Fielding statistics such as catches, and stumping are taken into consideration.

1.1.9 Software

The software being used are:

- R Software
- Python

Chapter 2

Methodologies

2.1 Introduction

This chapter includes the statistical methods which we have used in our project.These methods used will address the objectives of the study.we will used some statistical tools like machine learning techniques for classification,prediction,multivariate techniques,and MANOVA to assess the objectives of our study.

2.2 Principal Component Analysis(PCA)

Principal components are linear combinations of random or statistical variables which have special properties in terms of variances. For example, the first principal component is the normalized linear combination (the sum of squares of the coefficients being one) with maximum variance.In effect, transforming the original vector variable to the vector of principal components amounts to a rotation of coordinate axes to a new coordinate system that has inherent statistical properties.This choosing of a coordinate system is to be contrasted with the many problems treated previously where the coordinate system is irrelevant. The principal components turn out to be the characteristic vectors of the covariance matrix.Thus the study of principal components can be considered as putting into statistical terms the usual developments of characteristic roots and vectors (for positive semi-definite matrices).

2.2.1 Steps to Perform PCA for Rank the Players

Step-1:**Data Collection**
Collect a data set of cricket player statistics. Include attributes such as batting average, bowling average, strike rate, runs scored, wickets taken, and any other relevant performance metrics for each player.

Step-2:**Data Preprocessing**
Standardize the data to have a mean of 0 and a standard deviation of 1. This is crucial for PCA, as it is sensitive to the scale of the variables.

Standardization Formula:

$$Z = \frac{(X - \mu)}{\sigma}$$

Where:
• Z is the standardized value.
• X is the original value.
• μ is the mean of the variable.
• σ is the standard deviation of the variable.

Step-3:**Compute the Covariance Matrix**
Calculate the covariance matrix of the standardized data. The covariance matrix represents the relationships between different attributes.

Covariance Matrix Formula (for two variables X and Y):

$$Cov(X, Y) = \frac{1}{(N-1)} \sum_{i=1}^{N} (X_i - \bar{X})(Y_i - \bar{Y}) \tag{2.1}$$

where:
• X_i and Y_i are the data points for variables x and y.
• $\bar{X}$ and $\bar{Y}$ are the means of X and Y
• N is the number of data points.

we will create a covariance matrix that shows how the variables in your dataset are related to each other.
Step-4:**Eigenvalue Decomposition**
Compute the eigenvalues and eigenvectors of the covariance matrix(C). These eigenvectors represent the principal components (PCs), and the corresponding eigenvalues indicate the variance explained by each PC.

$$C\ v_i = \lambda\ v_i$$

Where:
- C is the Covariance Matrix.
- v_i is the ith eigenvector.
- λ_i is the ith eigenvalue.

Step-5:**Select Principal Components**
Sort the eigenvectors in descending order of their eigenvalues. We can choose to keep a certain number of principal components that explain most of the variance in the data.

Step-6:**Compute Principal Component Scores**
Here we consider only PC1 Scores for ranking,To calculate the scores for PC1 for each player we multiply their standardized data by the corresponding eigenvector.

For PC1:

$$PC1Score = Zv_1$$

Step-7:**Ranking the Players**
Rank the players based on their PC1 scores. Players with higher PC1 scores contribute more to the variance in the data along PC1, and by using Biplot we conclude that the most influential features are lie along PC1 axis.

To Identify the Most influencing feature in players ranking, we use Biplot. The vectors in the biplot represent the loadings of variables on the principal components, making it easier to understand the contributions of variables to each component.

2.3 Cluster Analysis

Cluster analysis is a data analysis technique used in various fields to group similar objects or data points together into clusters.The primary objective of cluster analysis is to uncover underlying patterns or structures in a dataset by identifying groups that share common characteristics. In the context of analyzing the performance of players in One Day International (ODI) cricket, cluster analysis can be a valuable tool to identify patterns and group players with similar playing styles, strengths, weaknesses, or performance characteristics.

2.3.1 Types of Cluster Analysis

1.Hierarchical Clustering: Hierarchical clustering is a method that creates a hierarchy of clusters by iteratively merging or dividing existing clusters. It forms a tree-like structure called a dendrogram that visually represents the arrangement of clusters at different levels.

Steps to Perform Hierarchical Cluster Analysis

Step-1:**Data Collection**
Collect relevant data on cricket player performance. This may include statistics like batting average, bowling average, strike rate, centuries, wickets taken, etc. You should have a dataset where each row represents a player, and each column represents a performance metric.

Step-2:**Data Preprocessing**
Normalize your data to ensure that all performance metrics are on the same scale. we can use z-score normalization or min-max scaling for this purpose.

Step-3:**Distance Metric**
Choose an appropriate distance metric to measure the dissimilarity between players. Common distance metrics include Euclidean distance, Manhattan distance, or Mahalanobis distance. The choice of distance metric depends on the nature of our data.

⋆ Euclidean distance is given by:

$$d(x,y) = \sqrt{(x-y)'(x-y)} = \sqrt{\sum_{j=1}^{p}(x_j - y_j)^2}. \tag{2.2}$$

Step-4:**Linkage Method**
Select a linkage method to determine how clusters are formed at each step of the hierarchical clustering process. Common linkage methods include single linkage, complete linkage, and average linkage.

The choice of linkage method determines how clusters are merged.
For example, in complete linkage, the distance between two clusters is defined

as the maximum distance between any two data points from each cluster.

The formula for complete linkage distance between clusters A and B is:

$$D(A, B) = max_{x \in A, y \in B} D(x, y)$$

Step-5: **Hierarchical Clustering**
Perform hierarchical clustering by taking features Batting Average,Strike rate, Wickets and Economy using the chosen distance metric and linkage method. This will create a hierarchical tree-like structure known as a dendrogram, which shows how players are grouped together at different levels of similarity.

Step-6:**Determine the Number of Clusters**
Decide how many clusters we want to create.We can use technique elbow method or silhouette score to help determine the optimal number of clusters.

Step-7:**Cutting the Dendrogram**
Based on the number of clusters we decided in the previous step, cut the dendrogram at the appropriate level to obtain the desired clusters of players.

Step-8:**Assign Players to Clusters**
Assign each player to the clusters based on the results of the dendrogram cutting.

Step-9:**Interpret the Clusters**
Analyze the clusters to interpret the characteristics of high-performing, moderate-performing, and low-performing players. This may involve looking at the average performance metrics within each cluster.

The dendrogram shows how the players are grouped based on the distances between their features.Players that are close to each other in the dendrogram have similar average and strike rates.The vertical height of the dendrogram's branches indicates the distance between clusters. Shorter branches suggest players with similar attributes.

2.K-means Clustering: K-means clustering is a partitioning method that divides data points into K clusters, where K is a user-defined parameter. It aims to minimize the variance within clusters and maximize the variance between clusters.

Steps to Perform K-Means Clustering.

Step-1:**Data Collection**

Gather the relevant data on cricket players' performance, including statistics like batting average, bowling average, strike rate, and other relevant metrics.

Step-2:**Data Preprocessing**

Normalize or standardize the data to ensure that all variables have the same scale.

Step-3:**Feature Selection**

Choose the relevant features (performance metrics) that will be used for clustering. These could be batting average, bowling average, strike rate, etc.

Step-4:**Choosing the Number of Clusters(K)**

Decide how many clusters we want to create. This can be done using various methods like the Elbow method, Silhouette method.

Step-5:**K-means Clustering Algorithm**

1. **Initialization**: Randomly initialize k cluster centroids: $C_1, C_2, ..., C_k$.
2. **Assignment**: For each data point X_i, calculate its distance to each cluster centroid C_j using the Euclidean distance formula:

   $$d(X_i, C_j) = \sqrt{\sum_{k=1}^{n} (X_{ik} - C_{jk})^2}. \tag{2.3}$$

   Assign X_i to the cluster with the nearest centroid:

   $$X_i \rightarrow argmin_j d(X_i, C_j) \tag{2.4}$$

3. **Update**:Recalculate the cluster centroids C_j as as the mean of all data points assigned to cluster j:

   $$C_j = \frac{1}{N} \sum_{i=1}^{N_j} X_i \tag{2.5}$$

   Where N_j is the number of data points in cluster j

4. **Convergence**:Repeat steps 2 and 3 until convergence, which is typically defined as no change in cluster assignments or a maximum number of iterations.

5. **Result**:The final cluster assignments determine which players belong to the "high performing," "moderate performing," and "low performing" categories.

2.4 Machine Learning Techniques for Classification

For Classification of players as Batsmen Bowler and All-rounder here we use machine learning techniques like SVM,Random Forest,Neural Networks,Decision Trees,Logistic Regression, Multinomial Regression.

2.4.1 Support Vector Machine

Support Vector Machine(SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well. Support Vectors are simply the coordinates of individual observation.

Steps to Perform SVM Classification

Step-1:**Data Preparation**
Collect a dataset of cricket players with features such as batting average (BA), bowling average (BOA), and other relevant statistics.Label each player as "Batsman," "Bowler," or "All-Rounder" based on their primary role.

Step-2:**Feature Selection**
Select relevant features from your dataset that are indicative of a player's batting and bowling abilities. These features could include batting average, bowling average, strike rate, and more.

Step-3:**Labeling**
Label the data points based on the class labels: batsman, bowler, and all-rounder(0,1,and 2). You may use historical data or expert knowledge to assign labels to each player.

Step-4:**Splitting the Dataset**
Split the dataset into training and testing sets. This helps evaluate the model's performance on unseen data.

Step-5:**SVM Model Selection**
Choose the type of SVM kernel that suits your data. Common choices include linear, polynomial, and radial basis function (RBF) kernels.

Step-6:**Formulate the Optimization Problem**
Our goal is to find the optimal hyperplane that maximizes the margin between classes while minimizing classification errors.
The optimization problem for a linear SVM is formulated as:
Maximize:

$$2\sum_{i=1}^{n}\alpha_i - \sum_{i=1}^{n}\sum_{i=1}^{j=1}\alpha_i\alpha_j y_i y_j(x_i, xj) \tag{2.6}$$

Subject to:

$$0 \leq \alpha_i \leq C, for i = 1, 2, ..., n$$

$$\textstyle\sum_{i=1}^{n}\alpha_i y_i = 0$$

Where:
• α_i are the Lagrange Multipliers.
• C is a regularization parameters that controls the trade-off between maximizing the margin and minimizing classification errors.
• n is the number of data points.

Step-7:**Solve the Optimization Problem**
By Using the optimization techniques for e.g.,Sequential Minimal Optimization to find the values of α_i.

Step-8:**Calculate the Support Vectors**
The support vectors are the data points corresponding to non-zero α_i

Step-9:**Prediction**
Use the trained SVM model to classify new cricket players into one of the three categories: batsman, bowler, or all-rounder based on their attributes.

Step-10: **Model Evaluation**
Evaluate the SVM model's performance on the testing data using metrics such as confusion matrix and accuracy.

2.4.2 Random Forest

A random forest is an ensemble machine learning algorithm that combines multiple decision trees to make more accurate predictions or classifications. It works by aggregating the results of many individual trees to reduce over fitting and improve overall model performance.

Steps to Perfoem Random Forest for Classification

Step-1: **Data Preparation**
Collect a dataset of cricket players with features such as batting average (BA), bowling average (BOA), and other relevant statistics. Label each player as "Batsman," "Bowler," or "All-Rounder" based on their primary role.

Step-2: **Feature Scaling**
Standardize or normalize the features to ensure they have similar scales.

Step-3: **Random Forest Algorithm**

1. **Bootstrapping**: Randomly select subsets (with replacement) of your dataset for each tree.

2. **Feature Selection**: For each tree, randomly select a subset of features at each split. This helps in decorrelating the trees and reducing over fitting.

3. **Decision Tree Construction**: Build a decision tree for each subset of data using a criterion like Gini impurity or entropy to determine the best split at each node.Here's a simplified version of how you calculate the impurity (Gini impurity) for a node:

$$Gini(t) = 1 - \sum_{i=1}^{C}[p(i)]^2 \tag{2.7}$$

Where:
• C is the number of classes (Batsman, Bowler, All-Rounder).

- p(i) is the proportion of samples in class i at the node.

4. **Voting**: For classification tasks, each tree predicts a class for a given input player's statistics.
   In the case of Random Forest, the final prediction is made by taking a majority vote among the predictions of all the trees.

Step-4:**Prediction**
Use the trained Random Forest model to make predictions for new or unseen player data.

Step-5:**Evaluation**
Evaluate the Random Forest model's performance using a validation dataset. Use evaluation metrics such as accuracy, precision, recall, and F1-score to assess the model's performance.

2.4.3 Multinomial Logistic Regression

Multinomial logistic regression is a statistical model used to predict the probability of an observation falling into one of several possible categories (classes) based on one or more predictor variables. It's particularly useful when dealing with categorical dependent variables with more than two categories. In Multinomial logistic regression, model consist of an observation class k (where k represents one of the possible classes) as a linear combination of predictor variables. The probability that an observation belongs to class k is then calculated using the soft-max function. The formula can be expressed

$$P(Y=k|X) = \frac{exp(\beta_{0k} + \beta_{1k}X_1 + \beta_{2k}X_2 + ... + \beta_{pk} + X_p)}{\sum_{j=1}^{k} exp(\beta_{0j} + \beta_{1j}X_1 + ... + \beta_{pj}X_p)} \tag{2.8}$$

Where:

- $P(Y=k|X)$ is the probability of the observation being in class k given the predictor variables X.
- $\beta_{0k}, \beta_{1k}, ..., \beta_{pk}$ are the coefficients associated with class k for each predictor variable.
- $X_1, X_2, ..., X_p$ are the predictor variables .
- K is the total number of classes.

Steps to Perform Multinomial Logistic Regression

Step-1: **Data Collection and Preprocessing**:
Gather a dataset that includes features for each cricket player (e.g., batting average, bowling average, wickets taken, etc.) and their corresponding class labels (batsman, bowler, all-rounder).

Step-2: **Data Exploration and Visualization**
• Explore the dataset to understand its characteristics and identify any missing values or outliers.
• Visualize the data to gain insights into the relationships between different features and the target classes.

Step-3: **Data Splitting**
Split your dataset into training and testing sets. A common split is 70% for training and 30% for testing.

Step-4: **Feature Scaling (Optional)**
Depending on data and algorithm, we may need to scale or standardize our features to ensure they have similar scales.

Step-5: **Model Building**
• Perform Multinomial Logistic Regression using a suitable library or programming language (e.g.,Python with libraries like scikit-learn).
• The mathematical formula for the Multinomial Logistic Regression model can be expressed as follows: For each class 'i' (batsman, bowler, all-rounder):

$$p(Y = i|X) = \frac{e^{(X\beta_i)}}{\sum_{j=1}^{k} e^{(X\beta_j)}} \tag{2.9}$$

Where:
• P(Y = i/X) is the probability that the instance belongs to class i.
• X represents the feature vector for a given instance.
• β_j represents the parameter vector associated with class i.
• K is the total number of classes.

Step-6: **Model Training**
Fit the Multinomial Logistic Regression model to the training data.
Step-7: **Prediction**
Once the model is trained we can use it to make predictions on new or unseen data.

Step-8: **Model Evaluation**
Evaluate the model's performance on the testing dataset using appropriate metrics accuracy.

2.4.4 Neural Networks

A Neural Network is a computational model inspired by the structure and function of the human brain. It consists of interconnected nodes, called neurons, organized in layers. Neural networks are used for various machine learning tasks, including pattern recognition, regression, and classification.

Steps to Perform Neural Networks Classification

Step-1: **Data Collection and Preprocessing**
• Collect a dataset containing information about cricket players, including statistics like batting average, bowling average, runs scored, wickets taken, etc.
• Preprocess the data by normalizing or standardizing the features, handling missing values, and encoding categorical variables.

Step-2; **Data Splitting**
Split the dataset into a training set, validation set, and test set. Typically, an 80-10-10 or 70-15-15 split is used.

Step-3: **Feature Selection**
Select the relevant features for classification.

Step-4: **Neural Network Architecture**
• Design the neural network architecture. For a classification task, a common choice is a feedforward neural network (also known as a multi-layer perceptron).
• Define the input layer with neurons equal to the number of features.
• Add one or more hidden layers with activation functions (common choices are ReLU, Sigmoid, or Tanh).
• Define the output layer with neurons equal to the number of classes (3 in this case - batsman, bowler, all-rounder). Use a softmax activation function to get class probabilities.

Step-5: **Loss Function**
Choose a suitable loss function for classification. Cross-entropy loss is commonly used for multi-class classification:

Cross-Entropy Loss (Categorical Cross-Entropy):

$$L(y, \hat{y}) = (-\sum(y_i * \log(\hat{y}_i))) \tag{2.10}$$

Where:
- $L(y, \hat{y})$ is the loss.
- y_i is the true class probability.
- $\hat{y}_i$ is the predicted class probability.

Step-6: **Optimizer**
Select an optimizer algorithm to minimize the loss function. Common choices include Adam, SGD(Stochastic Gradient Descent), or RMSprop.
Step-7: **Train the Model**
- Train the neural network using the training data.
- Use backpropagation to update the model's weights and biases.

Step-8: **Model Validation**
Monitor the model's performance on the validation set during training to avoid overfitting. Early stopping can be applied if the validation loss stops improving.

Step-9: **Prediction**
Use the trained model to classify new cricket players into one of the three categories: batsman, bowler, or all-rounder.

Step-10: **Model Evalution**
Evaluate the trained model on the test set to assess its performance and here we use accuracy as the metrics for classification.

2.4.5 Decision Trees

A decision tree is a supervised machine learning algorithm used for both classification and regression tasks. It resembles a tree-like structure where each internal node represents a feature (or attribute), each branch represents a decision rule, and each leaf node represents an outcome or a class label.

Steps for Decision Tree Classification

Step-1: **Data Collection**
Data Collection: Gather a dataset that includes information about cricket players, such as their batting averages, bowling averages, number of runs scored, number of wickets taken, and any other relevant features. Ensure that the dataset is labeled with the corresponding player types (batsman, bowler, or all-rounder).

Step-2: **Data Preprocessing**
- Handle missing data:we can Replace or remove missing values in the dataset.
- Encode categorical variables: Convert categorical variables like player country or team into numerical values using techniques like one-hot encoding.
- Split the dataset into a training set and a testing set for model evaluation.

Step-3: **Decision Tree Algorithm**
Select a decision tree algorithm like CART, or Random Forest.

Step-4: **Feature Selection**
Determine which features (batting average, bowling average, etc.) are relevant for classifying players. we can use techniques like information gain or Gini impurity to rank feature importance.

Step-5: **Training the Decision Tree**
- Feed the training data into the Decision Tree algorithm.
- The algorithm will recursively split the data based on the selected features to create a tree structure. • At each node of the tree, it selects the feature that provides the best separation of the data according to a criterion (e.g., Gini impurity or entropy).

Step-6: **Stopping Criteria**
The algorithm continues to split nodes until a stopping criterion is met. Common stopping criteria include a maximum depth for the tree or a minimum number of samples required to split a node.

Step-7: **Pruning**
After the tree is built, we can perform pruning to remove branches that do not significantly improve classification performance. Pruning helps prevent overfitting.

Step-8: **Prediction**

Once the Decision Tree is trained, we can use it to classify players into one of the categories (batsman, bowler, or all-rounder) by traversing the tree based on their feature values.

Step-9: **Model Evaluation**
Evaluate the model's performance using the testing dataset and here we use confusion matrix and accuracy as the metrics for classification.

2.5 Multivariate Technique for Classification

2.5.1 Linear Discriminant Analysis

Linear Discriminant Analysis is a dimensionality reduction technique that is commonly used for supervised classification problems. It is used for modelling differences in groups i.e. separating two or more classes. It is used to project the features in higher dimension space into a lower dimension space.

Steps to Perform Linear Discriminant Analysis

Step-1: **Data Collection and Preparation**
• Firstly we collect data about cricket players, including features such as batting average, bowling average, strike rate, runs scored, wickets taken, etc.
• Label each player as a batsman, bowler, or all-rounder based on their playing style.

Step-2: **Data Preprocessing**
Standardize or normalize the data to ensure that all features have the same scale.

Step-3: **Compute Class Means**
Calculate the mean vectors for each class (batsman, bowler, all-rounder).

$$\mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_{ij} \tag{2.11}$$

Where:
• μ_i is the mean vector for class i.
• n_i is the number of samples in class i.

- x_{ij} is the jth sample in class i.

Step-4: **Compute Scatter Matrices**
Calculate the within-class scatter matrix S_w and the between-class scatter matrix S_b.

Within-Class Scatter Matrix:

$$S_w = \sum_{i=1}^{C} \sum_{j=1}^{n_i} (x_{ij} - \mu_i)(x_{ij} - \mu_i)^T \tag{2.12}$$

Between-Class Scatter Matrix:

$$S_b = \sum_{i=1}^{C} n_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{2.13}$$

Where:

- C is the number of classes.
- μ is the overall mean vector.

Step-5: **Compute Eigenvectors and Eigenvalues**
Calculate the eigenvalues and corresponding eigenvectors of the matrix $S_w^{-1} S_b$.

Step-6: **Select Linear Discriminants**
Sort the eigenvalues in descending order and choose the top k eigenvectors to form a transformation matrix W.

step-7: **Project Data**
Project the original data onto the new subspace using the transformation matrix W.

$$Y = XW$$

Where:

- Y is the transformed data.
- X is the original data.
- W is the transformation matrix.

Step-8: **Train a Classifier**:
- Use the transformed data Y to train a classifier such as Logistic Regression,

Support Vector Machine, or k-Nearest Neighbors.

Step-9: **Make Predictions**
Use the trained classifier to make predictions for new, unseen data.

2.6 Machine Learning Techniques for Prediction

For Predicting of an match outcome here we use Machine learning techniues like SVM,Random Forest,Neural Networks,Logistic Regression,Naive Bayes,Gradient Boosting,K-NN.

2.6.1 K-Nearest Neighbors (KNN)

The k nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

Steps to Perform K-NN Algorithm for Prediction

Step-1: **Data Preparation**
• Collect and preprocess our cricket match data. Ensure that it includes features like Toss (1 for winning, 0 for losing), Batting First or Second (1 for batting first, 0 for batting second), and Home or Away (1 for home, 0 for away).
• Split our data into a training set and a testing set for model evaluation.

Step-2: **Choose the Number of Neighbors (K)**
Determine the value of K, which represents the number of nearest neighbors to consider when making a prediction. This is a hyperparameter that you can tune for optimal performance.

Step-3: **Calculate Distance**
For each data point in the testing set, calculate the Euclidean distance (or any other distance metric) between that point and all data points in the training set.
The Euclidean distance between two points (x1, y1) and (x2, y2) is given by:

$$d = \sqrt{((x2 - x1)^2 + (y2 - y1)^2)} \tag{2.14}$$

Step-4: **Find k Nearest Neighbors**
Sort the calculated distances in ascending order and select the top k data points with the smallest distances.

Step-5: **Make Predictions**
• For classification (e.g., predicting the match outcome as 'Win' or 'Lose'), count the number of neighbors in each class. The class with the majority of neighbors becomes the predicted class for the test data point.

Formula for classification:
Prediction=Majority class among the K nearest neighbors.

• For regression (e.g., predicting a numerical score), calculate the average or weighted average of the target values of the k nearest neighbors as the prediction.

Formula for regression:

$Prediction = \frac{1}{K} \sum_{i=1}^{k} y_i$
Where y_i is the target value of the i-th neighbor.

2.6.2 Naive Bayes Classifier

The Naive Bayes classifier is a popular supervised machine learning algorithm used for classification tasks such as text classification. It belongs to the family of generative learning algorithms, which means that it models the distribution of inputs for a given class or category. This approach is based on the assumption that the features of the input data are conditionally independent given the class, allowing the algorithm to make predictions quickly and accurately.

Steps To Perform Naive Bayes for Prediction

Step-1:**Data Collection**
Collect historical cricket match data with features such as toss result (win/lose), batting order (first/second), and home/away status, along with the corresponding match outcomes (win/lose).

Step-2:**Data Preprocessing**
Clean and pre-process the data, handling missing values and encoding categorical variables (e.g., toss result, batting first/second,match won/loss, home/away) into numerical values (e.g., 0 or 1).

Step-3:**Splitting Data**
Split the dataset into a training set and a testing set. The training set will be used to train the Naive Bayes model, and the testing set will be used to evaluate its performance.

Step-4:**Calculate Priors**
• Calculate the prior probabilities of winning and losing for the entire dataset. These are the probabilities that a team will win or lose regardless of any specific features. we can calculate them as follows:
• P(win) = (number of matches won) / (total number of matches)
• P(lose) = (number of matches lost) / (total number of matches)

Step-5:**Calculate Likelihoods**
• Calculate the likelihoods of each feature given the class (win or lose). For Naive Bayes, we assume that the features are conditionally independent, so we calculate the likelihoods independently for each feature. For example, we calculate the likelihood of winning given that the toss was won as follows:
• P(toss win — win) = (number of matches won with toss win) / (total number of matches won)
• P(toss win — lose) = (number of matches lost with toss win) / (total number of matches lost)
Similarly, calculate likelihoods for other features like batting order and home/away status.

Step-6: **Calculate Posteriors**
• Use Bayes' theorem to calculate the posterior probabilities for each class (win or lose) given the features. This involves multiplying the prior probabilities and the likelihoods for each feature and then normalizing. For example, for winning:
• P(win — features) ∞ P(win) * P(toss win — win) * P(batting order — win) * P(home away — win) Normalize the probabilities to sum to 1.

Step-7: **Prediction**
• Given a new set of features (toss result, batting first,match won/loss, home/away), calculate the posterior probabilities for each class (win or lose) using the formula from Step 6. The class with the highest posterior proba-

bility is our prediction.
Mathematically, the prediction can be represented as:
• Predicted Class = argmax(P(class) * P(feature 1 — class) * P(feature 2 — class) * ... * P(feature n — class))
Where:
• P(class) is the prior probability of the class (win or lose).
• P(feature i — class) is the likelihood of feature i given the class.

Steps to perform SVM for Prediction

Step-1: Data Collection
• Gather historical cricket match data, including features such as toss outcome (Toss), batting order (Batting First or Batting Second), and home or away status (Home or Away).
• Label each match outcome as a binary variable (e.g., 1 for win and 0 for loss).

Step-2: Data Preprocessing
• Encode categorical variables like "Toss" and "Home or Away" using one-hot encoding or label encoding .
• Normalize or standardize numerical features if needed.

Step-3: Split Data
Split your data set into a training set and a testing/validation set. This helps evaluate the model's performance on unseen data.

Step-4: SVM Model Selection
• Choose the appropriate type of SVM model based on your classification problem (e.g., linear SVM or kernel SVM).
• Formulate the SVM optimization problem:
Minimize:

$$0.5 * ||w||^2 + C * \sum[max(0, 1 - y_i * (w * x_i + b))] \tag{2.15}$$

subject to:
$y_i * (w * x_i + b) \geq 1$ for all training samples(x_i, y_i)

Where:
• w is the weight vector.
• b is the bias term.
• C is the regularization parameter.

• (x_i, y_i) are the training data points and labels.

Step-5: **Model Training**
Train the SVM model using the training dataset. The goal is to find the hyperplane that maximizes the margin between the two classes while minimizing classification errors.

Step-6: **Model Prediction**
Use the trained SVM model to make predictions on the testing dataset or new data.

Step-7: **Evaluate Model Performance**
Calculate performance metrics such as accuracy, precision, recall, F1-score, and ROC AUC to assess the model's predictive accuracy.

2.6.3 Neural Networks

Steps to perform Neural Networks for Prediction

Step-1: **Data Collection and Preparation**
• Gather historical cricket match data, including features such as Toss, batting(Batting First or Batting Second), and home or away.
• each match outcome as a binary variable (e.g., 1 for win and 0 for loss).

Step-2: **Data Preprocessing**
• Encode categorical variables like ”Toss” and ”Home or Away” using label encoding.
• Normalize or standardize numerical features if needed.
• Split our dataset into a training set, and a testing set.

Step-3: **Neural Network Architecture**
Design a neural network architecture suitable for our binary classification problem. A simple feedforward neural network with one or more hidden layers should suffice for this task.

Step-4: **Forward Propagation**
• Define the mathematical formulas for forward propagation. These include:
• Activation function for hidden layers (e.g., ReLU): $a_i = max(0, z_i)$, where a_i is the activation and z_i is the weighted sum of inputs.
• Output layer activation function (e.g., sigmoid for binary classification): $y_{pred} = 1/(1 + exp(-z))$, where y_{pred} is the predicted probability.

Step-5: **Loss Function**
Choose a suitable loss function for binary classification, such as binary cross-entropy loss:

$$L(y, y_{pred}) = -[y * log(y_{pred}) + (1 - y) * log(1 - y_{pred})]$$

Step-6: **Back propagation**
Implement the back propagation algorithm to compute gradients with respect to model parameters (weights and biases).

Step-7: **Training**
• Train the neural network model using the training dataset by iteratively updating the model parameters. Monitor the loss on the validation set to prevent overfitting.
• Training continues until the model converges or reaches a predefined number of epochs.

Step-8: **Prediction**
Use the trained neural network model to make predictions on the testing dataset.

2.6.4 Random Forest

Steps to Perform Random Forest for Prediction

Step-1:**Data Collection and Preparation**
Gather historical cricket match data, including features such as toss outcome (Toss), batting (Batting First or Batting Second), and home or away. Label each match outcome as a binary variable (e.g., 1 for win and 0 for loss).

Step-2:**Pre processing**
Encode categorical variables like ”Toss” and ”Home or Away” using one-hot encoding or label encoding. Split your dataset into a training set and a testing/validation set.

Step-3:**Random Forest Setup**
Choose the number of trees (n estimators) for your Random Forest.
Define other hyperparameters like maximum tree depth (max depth), minimum samples required to split a node (min samples split), and minimum samples required in a leaf node (min samples leaf).

Step-4:**Random Subset Selection**
For each tree in the Random Forest, randomly select a subset of the training data (with replacement, called bootstrapping). This creates diversity among the trees.

Step-5: **Decision Tree Training**
Train a decision tree on each of the randomly selected subsets from step 4.

Step-6: **Prediction**
• To make a prediction for a new data point, run it through each decision tree in the Random Forest.
• For classification tasks (like predicting match outcomes), each tree's prediction is treated as a vote, and the majority class is chosen as the final prediction.

2.6.5 Logistic Regression

Steps to Perform Logistic Regression for Prediction

Step-1: **Data Collection and Preparation**
Gather historical cricket match data, including features such as toss outcome (Toss win/lose), batting(Batting First or Batting Second), and home or away. Label each match outcome as a binary variable (e.g., 1 for win and 0 for loss).

Step-2: **Data Preparation**
Encode categorical variables like "Toss" and "Home or Away" using one-hot encoding or label encoding. Split your dataset into a training set and a testing/validation set.

Step-3: **Logistic Regression Model**
Formulate the logistic regression model, which models the probability of winning (class 1) as a function of the features.
The logistic regression model is represented as:

$$P(Y = 1|X) = 1/(1 + e^{-z}) \tag{2.16}$$

where:
•P(Y=1—X) is the probability of winning, X is the input feature vector, and z is the linear combination of features and model parameters:
z = b0 + b1 * x1 + b2 * x2 + ... + bn * xn
where
b0, b1, b2, ..., bn are the model coefficients, and x1, x2, ..., xn are the feature

values.

Step-4:**Loss Function**
Choose a suitable loss function for logistic regression, typically the log-likelihood or cross-entropy loss:

$$Loss(Y, P(Y=1|X)) = -[Y*log(P(Y=1|X))+(1-Y)*log(1-P(Y=1|X))] \tag{2.17}$$

Step-5: **Model Training**
•Train the logistic regression model using the training dataset.
• Use optimization algorithms like gradient descent or its variants to find the optimal coefficients (b0, b1, b2, ..., bn) that minimize the loss function.

Step-6:**Prediction**
• Use the trained logistic regression model to make predictions on the testing dataset.
• To classify an observation, compare the predicted probability (P(Y=1—X)) to a decision threshold (e.g., 0.5). If P(Y=1—X) $\geq$ 0.5, classify it as a win; otherwise, classify it as a loss.

2.7 Comparing overall performance of a player across different countries and continents using MANOVA

2.7.1 MANOVA

Multivariate Analysis of Variance (MANOVA) is a statistical technique used to compare the means of multiple dependent variables across two or more groups . In our case, we want to compare the overall performance of a cricket player across different countries using various performance metrics as dependent variables.

Steps to Perform MANOVA for Country Comparison

Step-1: **Data Collection**
Collect data on the performance metrics (runs, wickets, average, strike rate, bowling strike rate, bowling average, bowling dot percentage, catches) of cricket players across different countries. Each player's data should be associated with the country they played for.

Step-2: **Data Preparation**
Organize the data into a table where each row represents a player, and columns represent the different performance metrics and the country they played for.

Step-3: **Hypothesis**
Formulating the hypothesis. In this case, we might want to test whether there are significant differences in the performance metrics (dependent variables) across different countries (independent variable).

Step-4: **MANOVA Analysis**
Now, let's perform the MANOVA analysis. The MANOVA model can be expressed mathematically as follows:

$$Y = X\beta + \epsilon \tag{2.18}$$

Where:
• Y is a vector of the dependent variables (performance metrics) for all players.
• X is the design matrix that includes dummy variables for the categorical independent variable (Country).
• β is a vector of coefficients representing the mean differences between countries.
• ϵ is a vector of error terms.

The MANOVA model tests whether there are significant differences between the groups (countries) in terms of the performance metrics.

Step-5: **Interpretation**
Interpreting the results of the MANOVA. Identify which performance metrics show significant differences across countries and which countries differ significantly from each other.

Steps to Perform MANOVA for Continents Comparison

Step-1: **Data Collection**
Gather the data for cricket players, including their performance metrics (runs, wickets, average, etc.) and the continent they belong to.

Step-2: **Data Preparation**
Ensure that our data is clean, with missing values handled appropriately. Create a data matrix where each row represents a player and each column

represents a performance metric. Assign numerical codes to continents (e.g., 1 for Asia, 2 for Africa, etc.) if they are not already in numeric format.

Step-3: **Hypothesis Formulation**
Formulating our null hypothesis (H0) and alternative hypothesis (H1). For example, H0 could state that there is no significant difference in cricket performance across continents, while H1 could state that there is a significant difference.

Step-4:**MANOVA Model**
The MANOVA model is represented as follows:

$$Y = X\beta + \epsilon \tag{2.19}$$

where:
• Y represents the vector of dependent variables (performance metrics).
• X represents the design matrix, including categorical predictors (continent) and possibly covariates.
• β represents the vector of unknown parameters.
• ϵ represents the vector of error terms.

Step-5:**Interpretation**
Interpreting the results of the MANOVA. Identify which performance metrics show significant differences across continents and which countries differ significantly from each other.

2.8 Odds Ratio

Definition:The **odds ratio (OR)** is a statistic that quantifies the strength of the association between the two events, A and B it is symmetry.
Two events are independent iff OR=1. The odds of one event are the same in either the presence or absence of the other event.
If OR $>$ 1, A and B are associated(correlated) in the sense that, compared to the absence of B, the presence of B raises the odds of A, and symmetrically the presence of A raises the odds of B.
If OR $<$ 1, A and B are negatively correlated, and the presence of one event reduces the odds of the other event.

2.8.1 What is odds ratio formula?

Odds Ratio =(odds of the event in the exposed group) / (odds of the event in the non-exposed group).The odds ratio formula in mathematical form is as follows:
Consider the two events A and B.

$$\theta = \frac{\pi 1(1 - \pi 2)}{\pi 2(1 - \pi 1)} \tag{2.20}$$

where,
$\pi 1$ is probability of the success occurring at event A and
$\pi 2$ is probability of the success occurring at event B

Steps for Checking Association Between Outcome of a Match

Step-1: **Data Collection**
Collect data on ODI matches, including information on whether the team that won the toss also won the match. we will need a dataset that contains these two categorical variables for multiple matches.

Step-2: **Create a Contingency Table**
Create a 2x2 contingency table to summarize the relationship between the two categorical variables: toss result and match outcome. The table will look like this:

	Match Outcome	
Toss Result	Win	Loss
Win	A	B
Loss	C	D

- A: Number of times the team that won the toss also won the match.
- B: Number of times the team that won the toss lost the match.
- C: Number of times the team that lost the toss won the match.
- D: Number of times the team that lost the toss also lost the match.

Step-3: **Calculate the Odds Ratio (OR)**
The odds ratio (OR) is calculated using the following formula:

$$OR = \frac{A * D}{B * C} \tag{2.21}$$

Step-4: **Interpretation**
The odds ratio (OR) tells us whether there is an association between winning

the toss and winning the match.

• If OR ≥ 1, it suggests that winning the toss is associated with a higher likelihood of winning the match.
• If OR ≤ 1, it suggests that winning the toss is associated with a lower likelihood of winning the match.
• If OR $= 1$, there is no association between winning the toss and winning the match.

2.9 PLOTS

The following plots are used for assessing our study objectives:

2.9.1 Bi-plots

A bi-plot combines two fundamental graphical techniques: principal component analysis (PCA) and scatter plots. PCA is a dimensionality reduction method that transforms the original variables into a set of orthogonal (uncorrelated) variables called principal components. These components capture the most significant sources of variation in the data. A bi-plot takes the first two or more principal components and creates a scatter plot where each observation and variable are represented as points in a two-dimensional space.

In a bi-plot, the position of observations and variables in this reduced space provides valuable insights into the structure and relationships within the data. Observations are represented as points, and their proximity in the bi-plot indicates similarity or dissimilarity. Variables, on the other hand, are represented as vectors emanating from the origin, and their direction and length convey information about their contributions to the principal components. The angle between vectors reflects the correlation between variables, and the length of vectors indicates the magnitude of their influence on the principal components.

Uses of Bi-plot

Biplots are often used in conjunction with principal component analysis (PCA). The vectors in the biplot represent the loadings of variables on the principal components, making it easier to understand the contributions of variables to each component.

2.9.2 ROC

Receiver Operating Characteristic (ROC) plots are a fundamental tool in the field of statistics and machine learning, primarily used for evaluating and visualizing the performance of classification models. ROC plots provide valuable insights into a model's ability to distinguish between two or more classes, making them an essential component of model assessment and selection.

An ROC plot, short for Receiver Operating Characteristic plot, is a graphical representation that illustrates the trade-off between a classification model's true positive rate (sensitivity) and its false positive rate (1-specificity) across various classification thresholds. In essence, it helps us assess how well a model can discriminate between the positive and negative classes by plotting the relationship between true positive and false positive rates as the decision threshold for classifying instances changes.

The central idea behind ROC plots is to measure the model's performance across different decision thresholds, allowing us to visualize its ability to balance between correctly identifying positive instances and minimizing the misclassification of negative instances. This trade-off is essential in many real-world applications, such as medical diagnostics, fraud detection, and spam email filtering, where the cost of false positives and false negatives can vary significantly.

Uses of ROC

ROC plots are a valuable tool for comparing the performance of multiple classification models. By examining the area under the ROC curve (AUC), we can quantify and compare the discriminative power of different algorithms. A higher AUC generally indicates a better-performing model.

2.9.3 Dendrogram

Dendrograms, derived from the Greek words "dendron" meaning tree and "gramma" meaning drawing, are graphical representations that play a pivotal role in various fields, including biology, data science, and taxonomy. These intricate tree-like structures serve as powerful tools for visualizing and analyzing hierarchical relationships among entities, such as biological species, data points, or clusters of data. Dendrograms are not just aesthetically pleasing illustrations; they provide valuable insights into the organization and similarities within complex data sets.

A dendrogram typically appears as a branching diagram with a tree-like structure, where each branch, or ”dendrite,” represents a group or cluster of objects or data points. The entities grouped together at the tips of the branches are more closely related to each other than to entities further away in the tree. The height at which branches merge or split in the dendrogram reveals the degree of similarity or dissimilarity among the entities being analyzed.

Dendrograms can be constructed through various techniques, including hierarchical clustering algorithms like agglomerative and divisive clustering. These methods iteratively combine or divide entities based on their similarity, gradually building the dendrogram structure.

Uses of Dendrogram

Dendrograms are fundamental in clustering analysis, where they aid in identifying natural groupings within datasets. This is particularly valuable in market segmentation, customer profiling, and pattern recognition tasks.

2.9.4 Silhouette plot

A Silhouette plot is a graphical representation used in cluster analysis to assess the quality and appropriateness of a clustering solution. It provides insights into how well-defined and distinct the clusters are within a dataset. The plot is particularly useful when you want to determine the optimal number of clusters (k) for your data or evaluate the quality of a clustering algorithm's output.

Uses in cluster analysis

1. Identifying the optimal number of clusters (k): You can create Silhouette plots for different values of k and choose the one that yields the highest average Silhouette score as the optimal number of clusters.

2. Assessing the quality of a clustering solution: After clustering your data, you can use Silhouette plots to evaluate the overall quality of the

clusters. Higher average Silhouette scores indicate better-defined and more distinct clusters.

3. Identifying outliers and misclassified data points: Negative Silhouette scores can help you identify data points that may not belong to any cluster or are assigned to the wrong cluster.

Interpretation

1. The Silhouette plot typically ranges from -1 to 1. A high Silhouette score (close to 1) indicates that the data point is well-matched to its own cluster and poorly matched to neighboring clusters, suggesting a good clustering.
2. A score near 0 suggests that the data point is on or very close to the decision boundary between two neighboring clusters.
3. A negative score (below -0.5) indicates that the data point may have been assigned to the wrong cluster.

2.10 Accuracy

Accuracy is a commonly used metric in machine learning and statistics to evaluate the performance of classification models. It measures the proportion of correctly classified instances out of the total number of instances in a dataset. In other words, accuracy tells you how often the model's predictions are correct.
Mathematically, accuracy is calculated using the following formula, based on the confusion matrix:

$$Accuracy = \frac{Number\, of\, Correct\, Predictions}{Total\, Number\, of\, Predictions} \tag{2.22}$$

In a confusion matrix, which is commonly used to assess the performance of a classification model, the key components are:

1. True Positives (TP): The number of instances that were correctly classified as positive (i.e., the model predicted them as positive, and they are actually positive).
2. True Negatives (TN): The number of instances that were correctly classified as negative (i.e., the model predicted them as negative, and they are actually negative).

3. False Positives (FP): The number of instances that were incorrectly classified as positive (i.e., the model predicted them as positive, but they are actually negative).

4. False Negatives (FN): The number of instances that were incorrectly classified as negative (i.e., the model predicted them as negative, but they are actually positive).

Chapter 3

Results Pertaining to Our Study

3.1 Introduction

This chapter includes the results which are obtained in our project. These results will address the objectives of our study. We have used following statistical tools like machine learning techniques for classification, prediction, multivariate techniques, MANOVA and obtained the following results.

3.2 Ranking of Players Using Principal Component Analysis

The multivariate principal component analysis as we discussed in section 2.2 used to rank players:

3.2.1 Ranking of men cricketers - all-rounders, batters and bowlers)

Ranking of men all-rounders

The results of principal component analysis pertaining to the ranking of men all-rounders are presented below.

Rank	Player Name	Rank	Player Name
1	Naseem Shah	31	Mustafizur Rahman
2	Jayden Seales	32	K Zondo
3	Jhye Richardson	33	Ish Sodhi
4	Mitchel Swepson	34	Pat Cummins
5	Faheem Ashraf	35	Taskin Ahmed
6	Haris Rauf	36	Dunith Wellalge
7	Wiaan Mulder	37	Keshav Maharaj
8	Jasprit Bumrah	38	Marco Jansen
9	Mark Wood	39	Hasan Ali
10	Anritch Nortje	40	Ben Duckett
11	Lockie Fergusion	41	Ashton Agar
12	Lungi Ngidi	42	Abdullah Shafique
13	Jofra Archer	43	Josh Inglis
14	Matt Henry	44	Nurul Hasan
15	Shoriful Islam	45	Mitchel Starc
16	Lahiru Kumara	46	Chris Woakes
17	Tim Southee	47	Sarfaraz Ahmed
18	Mossadek Hossain	48	Andile Phehlukwayo
19	Kasun Rajitha	49	Odean Smith
20	Rayman Reifer	50	Washington Sunder
21	Soumya Sarkar	51	Harry Brook
22	Trent Boult	52	Romario Shepherd
23	Kuldeep Yadav	53	Ben Stokes
24	Mohammad Wasim	54	Angelo Mathwes
25	Kagiso Rabada	55	Shardul Thakur
26	Mahesh Theekshana	56	Shadab Khan
27	Usama Mir	57	Mathew Wade
28	Mohammad Shami	58	Agha Salman
29	Shaheen Afridi	59	Alzaari Joseph
30	Josh Hazlewood	60	Tom Blundell

Rank	Player Name	Rank	Player Name
61	Sam Curran	99	Aiden Markram
62	Axar Patel	100	Dawid Malan
63	Zak Crawley	101	Dhananjaya De Silva
64	Cameron Green	102	Dasun Shanaka
65	Akeal Hosein	103	Dimuth Karunaratne
67	Mitchell Santner	104	Mohammed Rizwan
68	Joe Root	105	Devon Conway
69	Glenn Phillips	106	Rishab Pant
70	Roston Chase	107	Mahmudullah
71	Haris Sohail	108	Charith Asalanka
72	Rovman Powell	109	Shamarh Brooks
73	Chamika Karunaratne	110	Temba Bavuma
74	Ravindra Jadeja	111	Jonny Bairstow
75	Mohammad siraj	112	Heinrich Klassen
76	Darren Bravo	113	David Miller
77	Moen Ali	114	Vander Dussen
78	Shimron Hetmeyer	115	Imam Ul-haq
79	Dinesh Chandimal	116	David warner
80	Jason Holder	117	Steven Smith
81	K Verreynne	118	Jos Buttler
82	Michael Bracewell	119	Alex Carey
83	Afif Hossain	120	Kusal Mendis
84	Mehidy Hasan Miraz	121	Quinton De Kock
85	Philip Salt	122	Pathum Nissanka
86	Mitchel Marsh	123	Shubhman Gill
87	Sam Billing	124	Shreyas Iyyer
88	Hussain Shanto	125	K L Rahul
89	Kusal Perrera	126	Tamim Iqbal
90	Ravichandran Ashwin	127	Mushfiqur Rahim
91	Kyle Mayers	128	Kane Williamson
92	Marnus Labuchagne	129	Virat Kohli
93	Travis Head	130	Liton Das
94	Finn Allen	131	Nicholas Pooran
95	Henry Nicholls	132	Babar Azam
96	Shakib Al Hasan	133	Rohit Sharma
97	Will Young	134	Tom Latham
98	Wanindu Hasaranga	135	Daryl Mitchell

Table 3.1: **ODI ranking of men all-rounders using principal component analysis**

Interpretation: From the above table, We observe that among the 135 players "Naseem Shah" was the top-ranked all-rounder, and followed by Jayden Seals, Jhye Richardson, and so on.

Countrywise top ranked all-rounders are given below:

Country	**Player Name**
Pakistan	Naseem Shah
India	Jasprit Bumrah
Australia	Jhye Richardson
England	Mark Wood
NewZealand	Lockie Ferguson
West Indies	Jayden Seals
South Africa	Wiam Mulder
Srilanka	Lahiru Kumara
Bangladesh	Shoriful Islam

Table 3.2: **Top ranked all-rounders**

Interpretation: From the above table we can observe that "Jasprit Bumrah" was the top ranked all-rounder for india.

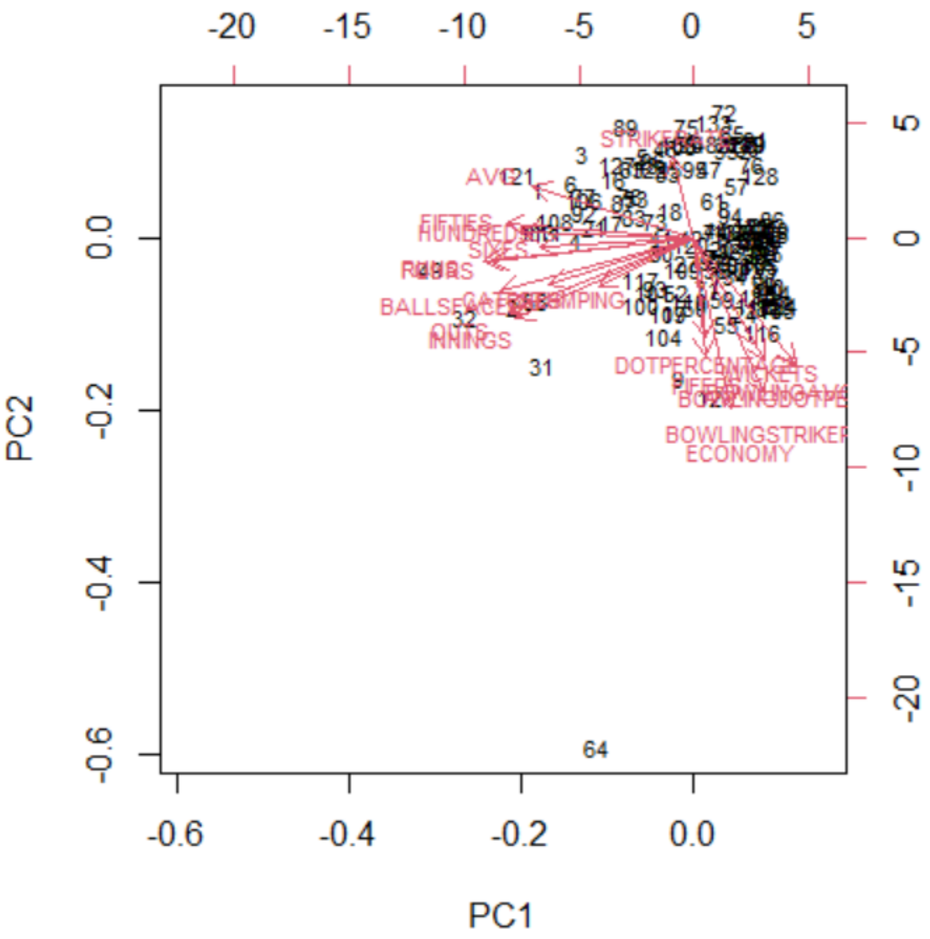

Figure 3.1: **Bi-plot of Men All-Rounders**

Interpretation: From the above plot we can see that Stumping, Avg, Fifties, Hundreds, Balls Faced, Outs, Innings, Fours and Sixes are negatively correlated with PC1 and the most influencing feature for ranking is Runs.

Ranking of men batters

The results of principal component analysis pertaining to the ranking of men batters are presented below.

Rank	Player Name	Rank	Player Name
1	Wiaan Mulder	41	Travis Head
2	Abdullah Shafique	42	Mehidy Hasan Miraz
3	Sarfaraz Ahmed	43	Hussain Shanto
4	Josh Inglis	44	Mohammed Rizwan
5	Ben Duckett	45	Rishab Pant
6	Hasan Ali	46	Kyle Mayers
7	Faheem Ashraf	47	Devon Conway
8	Harry Brook	48	Daryl Mitchell
9	Odean Smith	49	Aiden Markram
10	Zak Crawley	50	Marnus Labuchagne
11	Kasun Rajitha	51	Dawid Malan
12	Lahiru Kumara	52	Heinrich Klassen
13	Tom Blundell	53	Jonny Bairstow
14	Mathew Wade	54	Shakib Al Hasan
15	Angelo Mathwes	55	David Miller
16	Shadab Khan	56	Dhananjaya De Silva
17	Ben Stokes	57	Shamarh Brooks
18	Andile Phehlukwayo	58	Dasun Shanaka
19	Shimron Hetmeyer	59	Temba Bavuma
20	Romario Shepherd	60	Vander Dussen
21	Sam Curran	61	Charith Asalanka
22	Rovman Powell	62	David warner
23	Darren Bravo	63	Alex Carey
24	Dinesh Chandimal	64	Steven Smith
25	Agha Salman	65	Quinton De Kock
26	Cameron Green	66	Shubhman Gill
27	Glenn Phillips	67	Kusal Mendis
28	Mitchell Santner	68	Pathum Nissanka
29	Joe Root	69	Mushfiqur Rahim
30	Phil Salt	70	K L Rahul
31	K Verreynne	71	Shreyas Iyyer
32	Roston Chase	72	Tamim Iqbal
33	Sam Billing	73	Liton Das
34	Kusal Perrera	74	Jos Buttler
35	Moen Ali	75	Virat Kohli
36	Michael Bracewell	76	Nicholas Pooran

Rank	Player Name	Rank	Player Name
37	Finn Allen	77	Kane Williamson
38	Will Young	78	Rohit Sharma
39	Henry Nicholls	79	Tom Latham
40	Jason Holder	80	Shai Hope

Table 3.3: **ODI ranking of men batters using principal component analysis**

Interpretation: From the above table, We can observe that among the 80 players "Wiaan Mulder" is the top-ranked Batter, and followed by Abdullah Shafique, Sarfaraz Ahmed, and so on.

Countrywise top ranked batters are given below:

Country	Player Name	Rank
India	Rishab Pant	45
Pakistan	Abdullah Shafique	2
Australia	Josh Inglis	4
NewZealand	Tom Blundell	13
England	Ben Ducket	5
West Indies	Odean Smith	9
South Africa	Wiaan Mulder	1
Srilanka	Kasun Rajitha	11
Bangladesh	Mehady Hasan Miraj	42

Table 3.4: **Top ranked batters**

Interpretation: From the above table we can observe that "Rishab Pant" was the top ranked batter for india.

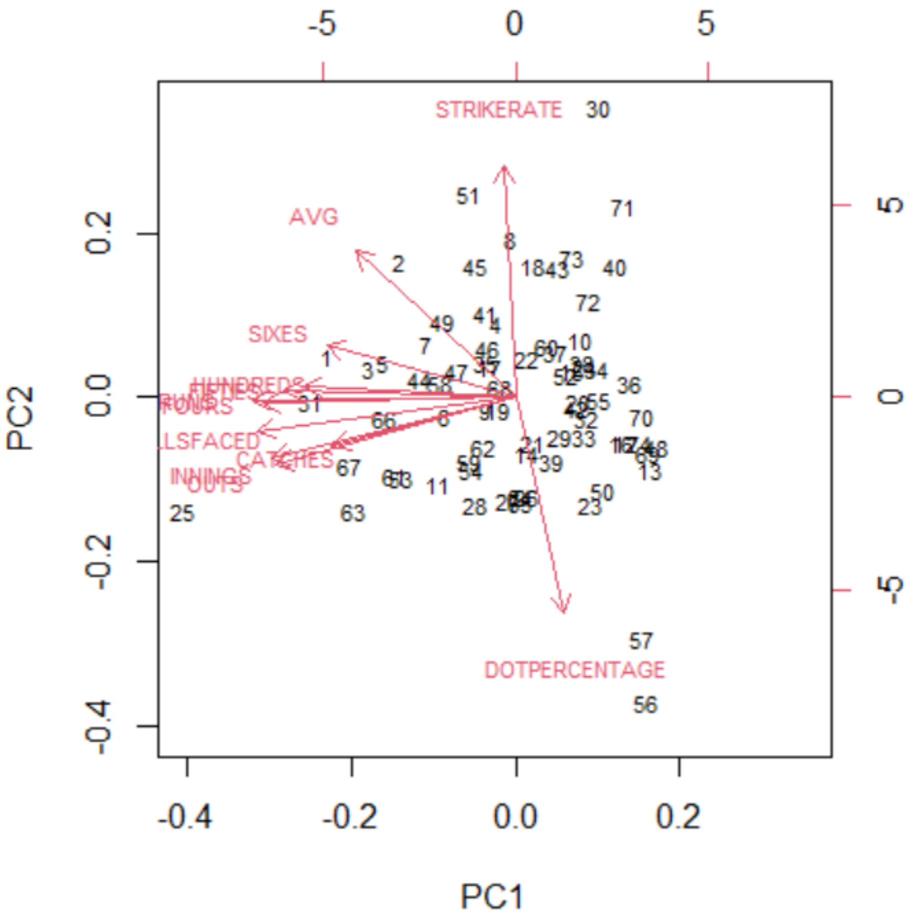

Figure 3.2: **Bi-plot of Men Batters**

Interpretation: From the above plot we can say that Avg, Fifties, Hundreds, BallsFaced, Outs, Innings, Fours and Sixes are negatively correlated with PC1 and the most influencing feature for menbatting ranking is Runs.

Ranking of men bowlers

The results of principal component analysis pertaining to the ranking of men bowlers are presented below.

Rank	Player Name	Rank	Player Name
1	Wanindu Hasaranga	23	Trent Boult
2	Ravichandran Ashwin	24	Mohammad Shami
3	Mohammad siraj	25	Dunith Wellalge
4	Mitchel Marsh	26	Ish Sodhi
5	Afif Hossain	27	Kuldeep Yadav
6	Michael Bracewell	28	Mohammad Wasim
7	Mitchell Santner	29	Usama Mir
8	Akeal Hosein	30	Mossadek Hossain
9	Alzaari Joseph	31	Matt Henry
10	Axar Patel	32	Tim Southee
11	Shardul Thakur	33	Shoriful Islam
12	Washington Sunder	34	Lungi Ngidi
13	Mitchel Starc	35	Jasprit Bumrah
14	Taskin Ahmed	36	Anritch Nortje
15	Josh Hazlewood	37	Lockie Fergusion
16	Keshav Maharaj	38	Faheem Ashraf
17	Ashton Agar	39	Naseem Shah
18	Hasan Ali	40	Haris Rauf
19	Mustafizur Rahman	41	Jhye Richardson
20	Pat Cummins	42	Jayden Seales
21	Shaheen Afridi	43	Mitchel Swepson
22	Mahesh Theekshana		

Table 3.5: **ODI ranking of men bowlers using principal component analysis**

Interpretation: From the above table, We conclude that among the 43 players "Wanindu Hasaranga" is the Top ranked bowler, and followed by Ravichandran Ashwin, Mohammad Siraj, and so on.

Countrywise top ranked bowlers are given below:

Country	**Player Name**	**Rank**
India	Ravichandran Ashwin	2
Australia	Mitchel Marsh	4
Pakistan	Hasan Ali	18
Bangladesh	Mustafizur Rahman	19
NewZealand	Michael Bracewell	6
South Africa	Keshav Maharaj	16
Srilanka	Wanindu Hasaranga	1
West Indies	Akeal hossian	5

Table 3.6: **Top ranked bowlers**

Interpretation: From the above table we can observe that ”Ravichandran Ashwin” was the top ranked bowler for india.

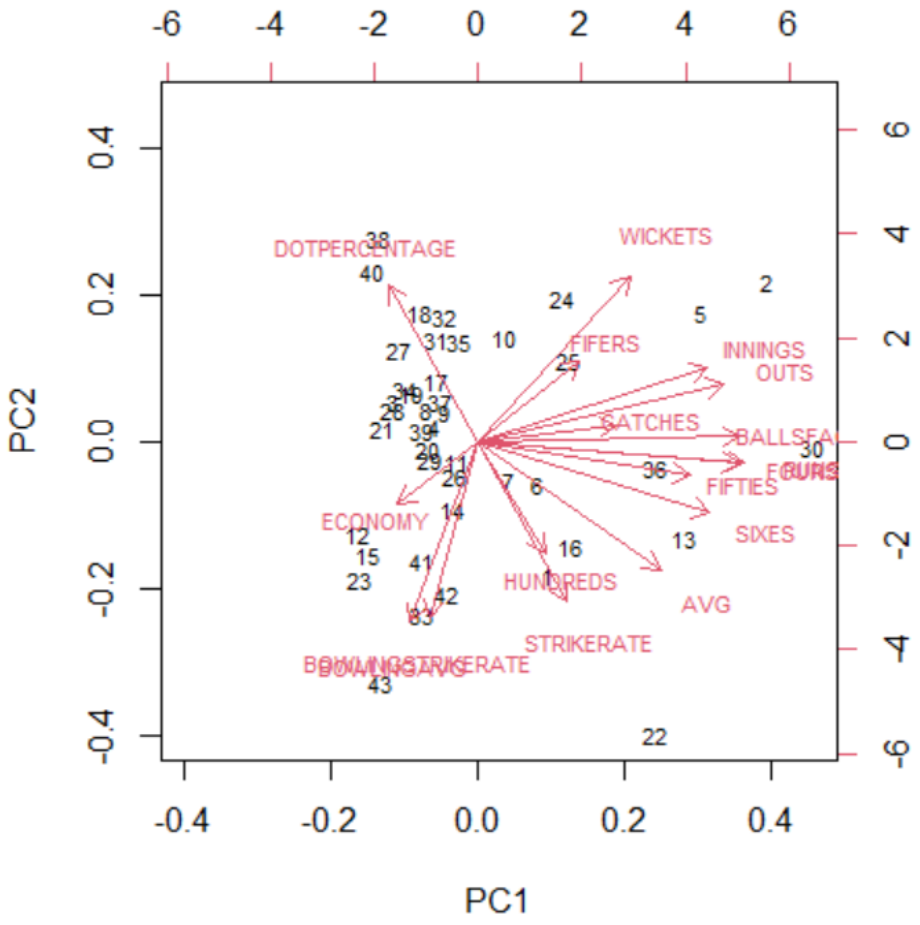

Figure 3.3: Bi-plot of Men Bowlers

Interpretation: From the above plot we can see that Wickets, Fifers, Catches, are positively correlated and Bowling Avg and Bowling Strike rate are negative correlated with PC1 and the most influencing feature for men bowlers ranking is Bowling Strike rate.

3.2.2 Ranking of women cricketers - all-rounders, batters and bowlers)

Ranking of women all-rounders

The results of principal component analysis pertaining to the ranking of women all-rounders are presented below.

Rank	Player Name	Rank	Player Name
1	Inoshi Priyadharshani	67	Sneh Rana
2	Radha Yadav	68	Kate Cross
3	Raisibe Ntozakhe	69	P Litchfield
4	Tayla Vlaeminck	70	Aaliyah Alleyne
5	Hannah Darlington	71	Shamima Sultana
6	Eden Carson	72	Rumana Ahmed
7	K Garth	73	Anushka Sanjeewani
8	Renuka Singh	74	A Sutherland
9	Ghulam Fatima	75	N Klerk
10	Molly Penfold	76	Lata Mondal
11	Fran Jonas	77	Ritu Moni
12	Sadia Iqbal	78	A Steyn
13	D Tucker	79	Kavisha Dilhari
14	A Wellington	80	Nat Scivar-Brunt
15	N Mlaba	81	Hayley Jensen
16	Lauren Bell	82	Sidra Nawaz
17	Isabella Gaze	83	Sophie Ecclestone
18	Nuzhat Tasnia	84	F Sana
19	Rabeya Khan	85	Jemimah Rodrigues
20	Freya Kemp	86	T Brits
21	Rajeshwari Gayakwad	87	Richa Ghosh
22	N Shangase	88	Emma Lamb
23	Freya Davies	89	Murshida Khatun
24	Achini Kulasuriya	90	Nilakshi de Silva
25	T Sekhukhune	91	T McGrath
26	Amanjot Kaur	92	Harshitha Samarawickrama
27	A King	93	Pooja Vastrakar
28	Sobhana Mostary	94	Chinelle Henry
29	Sugandika Kumari	95	Lauren Down
30	Rosemary Mair	96	Brooke Halliday
31	Ayesha Naseem	97	Javeria Khan
32	Meghna Singh	98	Chedean Nation
33	Cherry Ann Fraser	99	Kyshona Knight
34	Jahanara Alam	100	Sharmin Akther
35	Sarah Glenn	101	Nigar Sultana

Rank	Player Name	Rank	Player Name
36	A Khaka	102	Shafali Verma
37	Georgia Plimmer	103	Rashada Williams
38	Taniya Bhatia	104	Rashada Williams
39	Alice Davidson-Richards	105	Deepti Sharma
40	S Molineux	106	M kapp
41	Udeshika Prabodhani	107	A Gardener
42	Sheneta Grimmond	108	Chamari Athapaththu
43	Shamilila Connell	109	Nida Dar
44	Fahima Khatun	110	Amy jones
45	S Jafta	111	Fargana Hoque
46	Karishma Ramharack	112	Bismah Maroof
47	Nashra Sandhu	113	Sophia Dunkley
48	Bernadine Bezuidenhout	114	Muneeba Ali
49	Nahida Akter	115	Omaima Sohail
50	Inoka Ranaweera	116	E Perry
51	G Wareham	117	Aliya Riaz
52	Hansima Karunaratne	118	Danni Wyatt
53	Prasadani Weerakkody	119	Maddy Green
54	Shakera Selman	120	Deandra Dottin
55	Sadaf Shamas	121	S Luus
56	Ama Kanchana	122	Beth Mooney
57	Salma Khatun	123	Heather Knight
58	Alice capsey	124	Hayley Mathews
59	Oshadi Ranasinghe	125	Harmanpreet Kaur
60	Jess Kerr	126	Smriti Mandanna
61	Shabika Gajnabi	127	Tammy Beaumont
62	Kainat Imtiaz	128	Amelia Kerr
63	Harleen Deol	129	Suzie Bates
64	N Carey	130	Stefanie Taylor
65	Hasini Perera	131	L Lee
66	Diana Baig	132	L Wolvaardt

Table 3.7: **ODI ranking of women all-rounders using principal component analysis**

Interpretation: From the above table, We observe that among the 132 players "Inoshi Priyadharshani" was the top ranked all-rounder,and followed by Radha yadav,Raisibe Ntozakhe, and so on.

Countrywise top ranked all-rounders are given below:

Country	**Player Name**	**Rank**
India	Radha Yadav	2
Australia	Hannah Darlington	5
New Zealand	Eden Carson	6
West Indies	Cherry Ann Freser	33
South Africa	Raisibe Niozakhe	3
Sri Lanka	Inoshi Priyadharshani	1
Bangladesh	Nuzhat Tasnia	18
England	Lauren Bell	16
Pakisthan	Ghulam Fathima	9

Table 3.8: **Top ranked all-rounders**

Interpretation: From the above table we can observe that ”Radha Yadav” was the top ranked all-rounder for india.

I

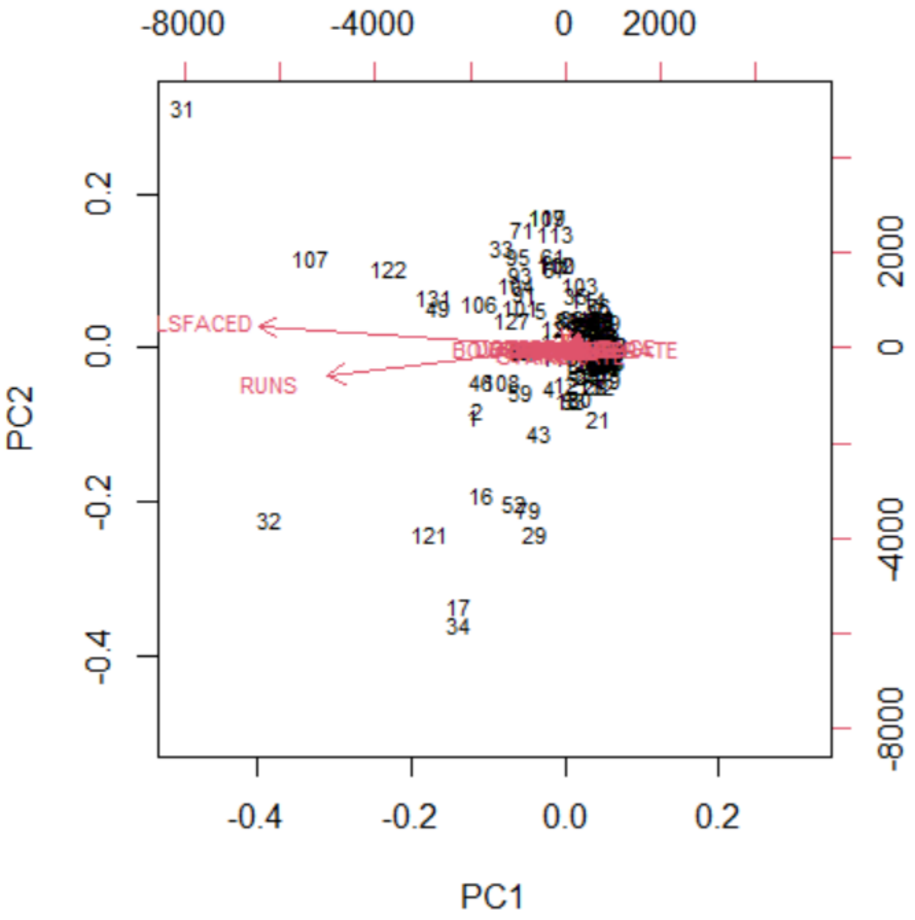

Figure 3.4: **Bi-plot of women all-rounders**

Interpretation: From the above plot we can see that Balls Faced and Runs are negatively correlated and outs, average, strike rate, fifties, hundreds, fours, sixes, dot percentage, wickets, economy, bowling avg, bowling strike rate, fifers, catches, stumping are positively correlated with PC1 and the most influencing feature for ranking is Balls Faced.

Ranking of women batters

The results of principal component analysis pertaining to the ranking of women batters are presented below.

Rank	**Player Name**	Rank	**Player Name**
1	A Khaka	39	Brooke Halliday
2	Taniya Bhatia	40	Javeria Khan
3	S Molineux	41	Chedean Nation
4	Shamilila Connell	42	Kyshona Knight
5	Nashra Sandhu	43	Sharmin Akther
6	G Wareham	44	Nigar Sultana
7	Shakera Selman	45	Shafali Verma
8	Ama Kanchana	46	Rashada Williams
9	Salma Khatun	47	Rashada Williams
10	Alice capsey	48	Deepti Sharma
11	Jess Kerr	49	M kapp
12	Harleen Deol	50	A Gardener
13	N Carey	51	Chamari Athapaththu
14	Hasini Perera	52	Nida Dar
15	Diana Baig	53	Amy jones
16	P Litchfield	54	Fargana Hoque
17	Aaliyah Alleyne	55	Bismah Maroof
18	Shamima Sultana	56	Sophia Dunkley
19	Anushka Sanjeewani	57	Muneeba Ali
20	A Sutherland	58	Omaima Sohail
21	N Klerk	59	Aliya Riaz
22	Lata Mondal	60	Danni Wyatt
23	Ritu Moni	61	Maddy Green
24	Kavisha Dilhari	62	Deandra Dottin
25	Nat Scivar-Brunt	63	S Luus
26	Hayley Jensen	64	Beth Mooney
27	Sidra Nawaz	65	Heather Knight
28	F Sana	66	Hayley Mathews
29	Jemimah Rodrigues	67	Harmanpreet Kaur
30	T Brits	68	Smriti Mandanna
31	Richa Ghosh	69	Alysa Healy
32	Emma Lamb	70	C Tryon
33	Murshida Khatun	71	Tammy Beaumont
34	Nilakshi de Silva	72	Amelia Kerr
35	T McGrath	73	Sophie Devina
36	Harshitha Samarawickrama	74	Suzie Bates
37	Pooja Vastrakar	75	Stefanie Taylor
38	Chinelle Henry	76	L Lee

Table 3.9: **ODI ranking of men batters using principal component analysis**

Interpretation: From the above table, We can observe that among the 76 players "A Khaka" was the Top ranked Batter,and followed by Taniya Bhatiya, S Molineux, and so on.

Countrywise top ranked batters are given below:

Country	**Player Name**	**Rank**
India	Taniya Bhatia	2
Australia	S Molineux	3
South Africa	A Khaka	1
England	Alice capsey	10
Bangladesh	Salma Khatun	9
Sri Lanka	Ama Kanchana	8
Pakistan	Nashra Sandhu	5
West Indies	Shamilila Connell	4
New Zealand	Jess Kerr	11

Table 3.10: **Top ranked batters**

Interpretation: From the above table we can observe that "Taniya Bhatia" was the top ranked bowler for india.

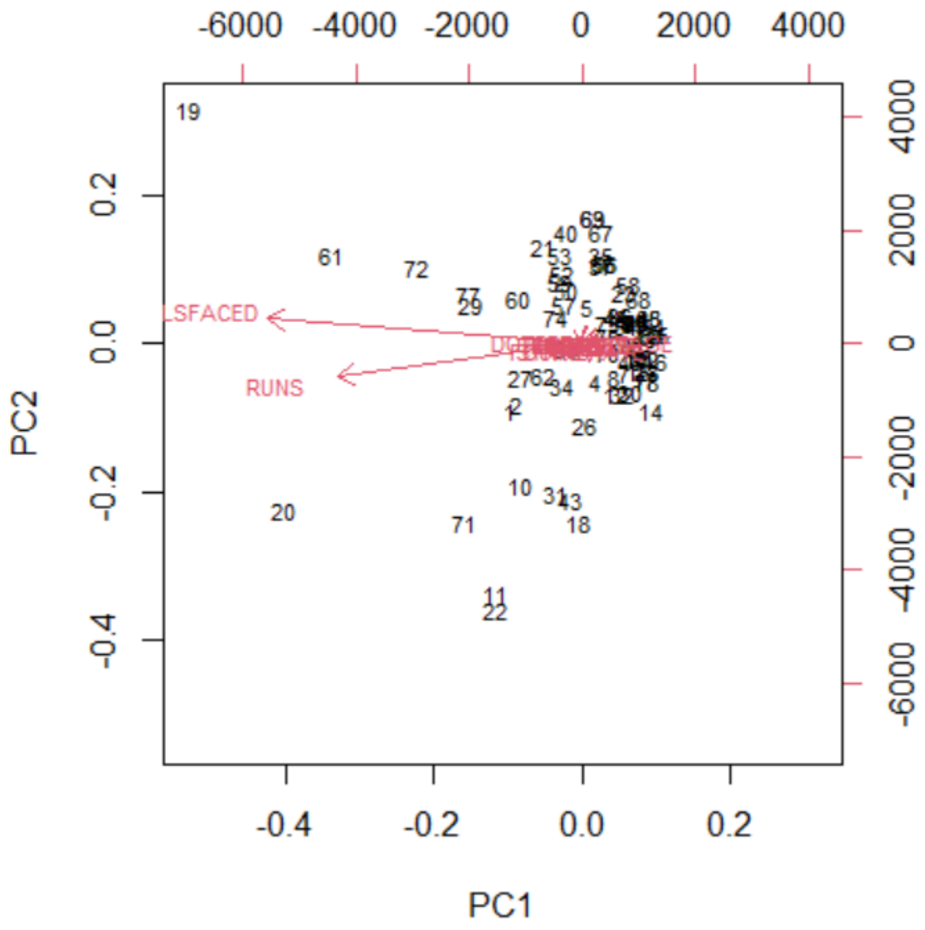

Figure 3.5: **Bi-plot of Women Batters**

Interpretation: From the above plot we can say that Avg, Fifties, Hundreds, Outs, Innings, Fours and Sixes are positively correlated with Runs and Balls Faced are negatively correlated with PC1 and the most influencing feature for Women's batting ranking is Balls Faced.

Ranking of women bowlers

The results of principal component analysis pertaining to the ranking of women bowlers are presented below.

Rank	Player Name	Rank	Player Name
1	Amanjot Kaur	34	A King
2	K Garth	35	G Wareham
3	Freya Kemp	36	Sneh Rana
4	Alice Davidson-Richards	37	N Carey
5	Ama Kanchana	38	Hayley Jensen
6	Nat Scivar-Brunt	39	N Klerk
7	Eden Carson	40	Shakera Selman
8	Chamari Athapaththu	41	Ritu Moni
9	Sugandika Kumari	42	Sophie Devina
10	Udeshika Prabodhani	43	Jess Kerr
11	Achini Kulasuriya	44	Chinelle Henry
12	Renuka Singh	45	Jahanara Alam
13	Lauren Bell	46	Shamilila Connell
14	Ghulam Fatima	47	Diana Baig
15	Freya Davies	48	Salma Khatun
16	S Molineux	49	F Sana
17	Fahima Khatun	50	M kapp
18	Rumana Ahmed	51	Nida Dar
19	Sarah Glenn	52	Nashra Sandhu
20	A Sutherland	53	Nahida Akter
21	Sadia Iqbal	54	Rajeshwari Gayakwad
22	Oshadi Ranasinghe	55	Kate Cross
23	A Wellington	56	A Khaka
24	Aaliyah Alleyne	57	Deepti Sharma
25	Sidra Nawaz	58	Amelia Kerr
26	T McGrath	59	Hayley Mathews
27	Karishma Ramharack	60	C Tryon
28	Inoka Ranaweera	61	Sophie Ecclestone
29	Omaima Sohail	62	A Gardener
30	E Perry	63	Stefanie Taylor
31	Meghna Singh	64	S Luus
32	Pooja Vastrakar	65	T Brits
33	T Sekhukhune	66	L Wolvaardt

Table 3.11: **ODI ranking of women bowlers using principal component analysis**

Interpretation: From the above table, We observe that among the 66 players "Amanjot Kaur" was the Top ranked Bowler,and followed by K Garth,Freya Kemp, and so on.

Countrywise top ranked bowlers are given below:

Country	**Player Name**	**Rank**
India	Amanjot Kaur	1
Australia	K Garth	2
South Africa	T Sekhukhune	33
England	Freya Kemp	3
Bangladesh	fahima Khatun	17
Sri Lanka	Ama Kanchana	5
Pakistan	Ghulam Fatima	14
West Indies	Aaliyah Alleyne	24
New Zealand	Eden Carson	7

Table 3.12: **Top ranked bowlers**

Interpretation: From the above table we can observe that "Amanjot Kuar" was the top ranked all-rounder for india.

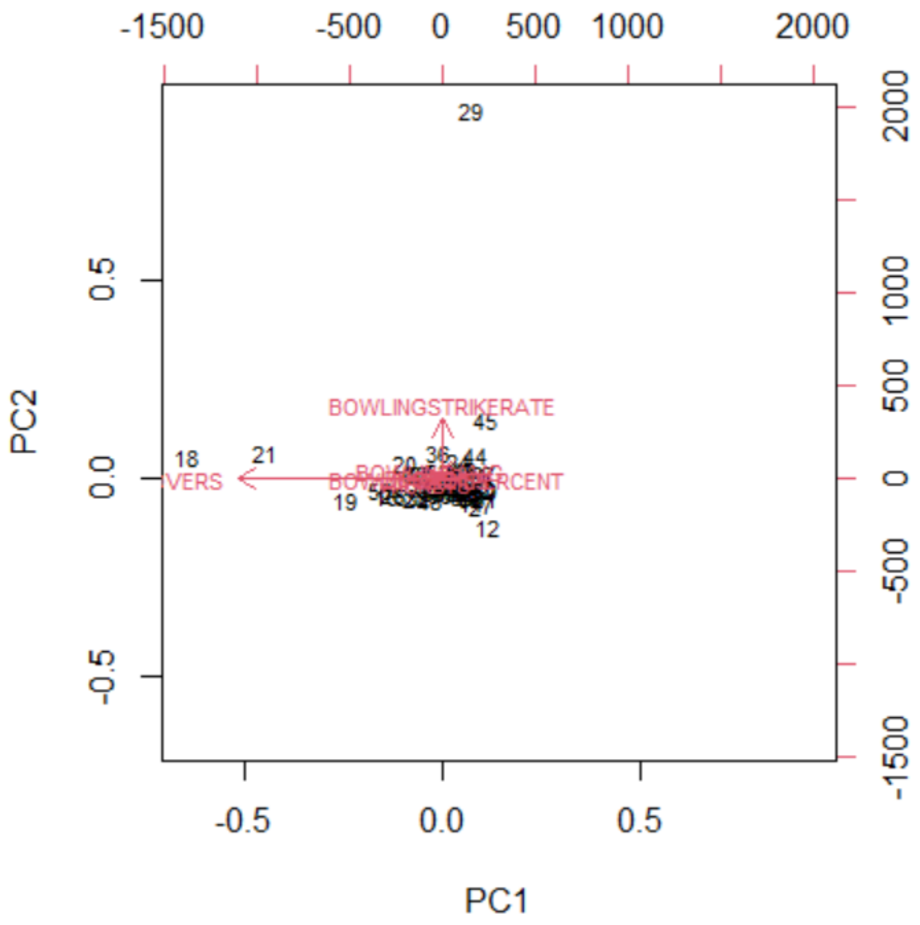

Figure 3.6: **Bi-plot of Women Bowlers**

Interpretation: From the above plot we can see that Wickets, Fifers Catches, economy, and Bowlingavg are positively correlated and Overs are negatively correlated with PC1 and the most influencing feature for men's bowling ranking is Overs.

3.3 Clustering of cricket players using cluster analysis

The methodology/statistical technique for cluster analysis as we discussed in section 2.3 used to cluster players:

3.3.1 Grouping of men cricketers using hierarchical clustering : all-rounders, batters and bowlers

Grouping of men all-rounders

The results of hierarchical cluster analysis pertaining to the grouping of men all-rounders are presented below.

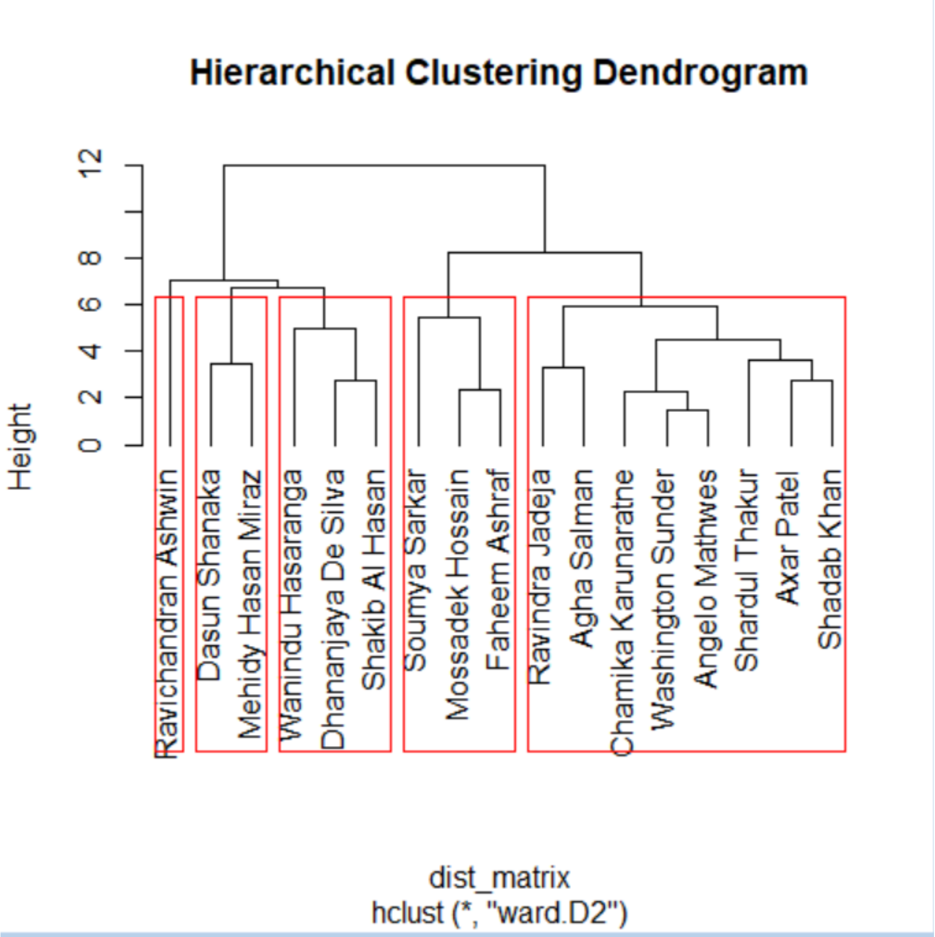

Figure 3.7: **Clustering of men all-rounders using hierarchical clustering**

Interpretation: The dendrogram shows how the players are grouped based on the distances between their features. Ravichandran Ashwin belongs to first cluster, Dasun Shanaka and Mehidy Hasan Miraz belongs to second cluster and so on. The vertical height of the dendrogram branches indicates the distance between clusters.

Grouping of men batters

The results of hierarchical cluster analysis pertaining to the grouping of men all-batters are presented below.

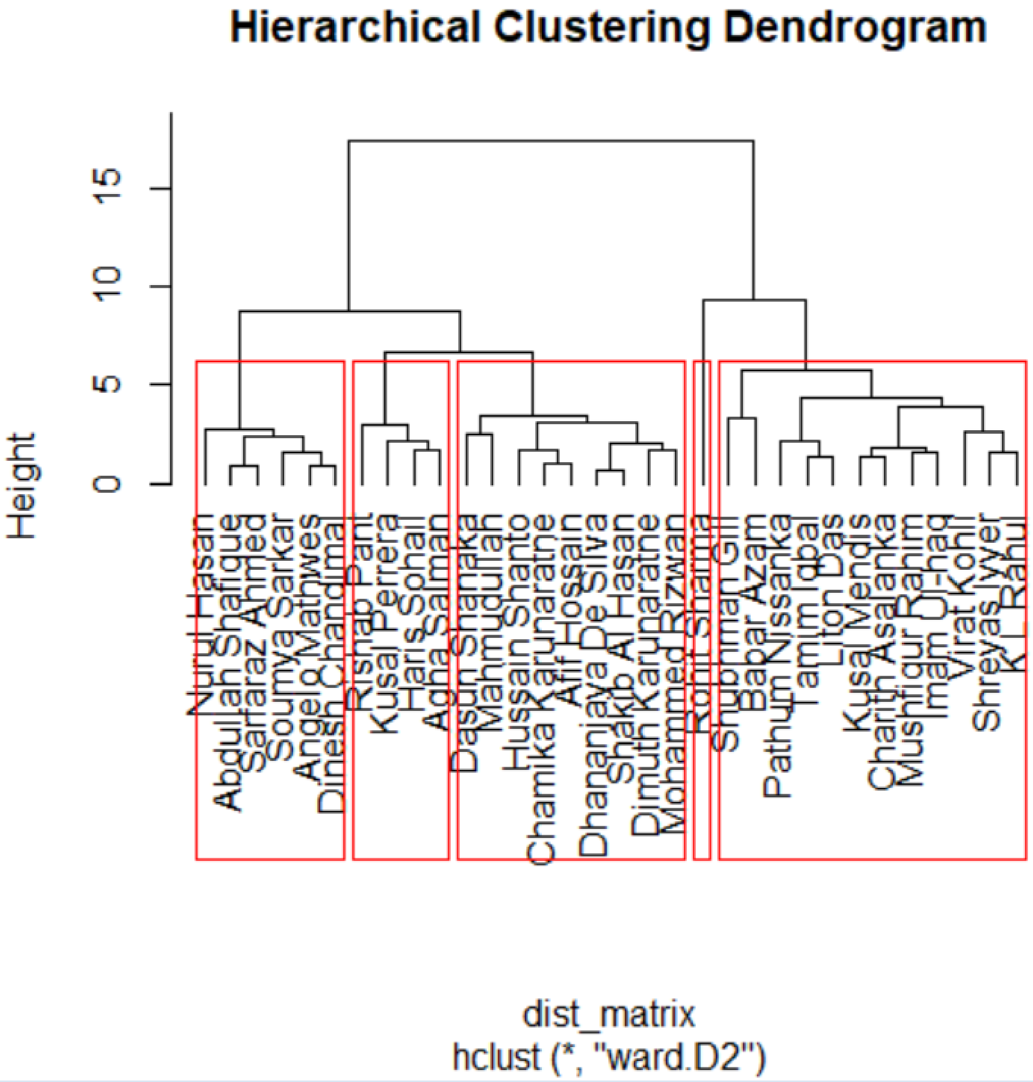

Figure 3.8: **Clustering of men batters using hierarchical clustering**

Interpretation: The dendrogram shows how the players are grouped based on the distances between their features. Nurul Hasan, Abdullah Shafique, Sarfaraz Ahmed, Soumya Sarkar, Angelo Mathews, and Dinesh Chandimal belongs to are first cluster, Rishab Pant, Kusal Perera, Haris Sohail, and

Agha Salman are belongs to second cluster, and so on. The vertical height of the dendrogram's branches indicates the distance between clusters.

Grouping of men bowlers

The results of hierarchical cluster analysis pertaining to the grouping of men bowlers are presented below.

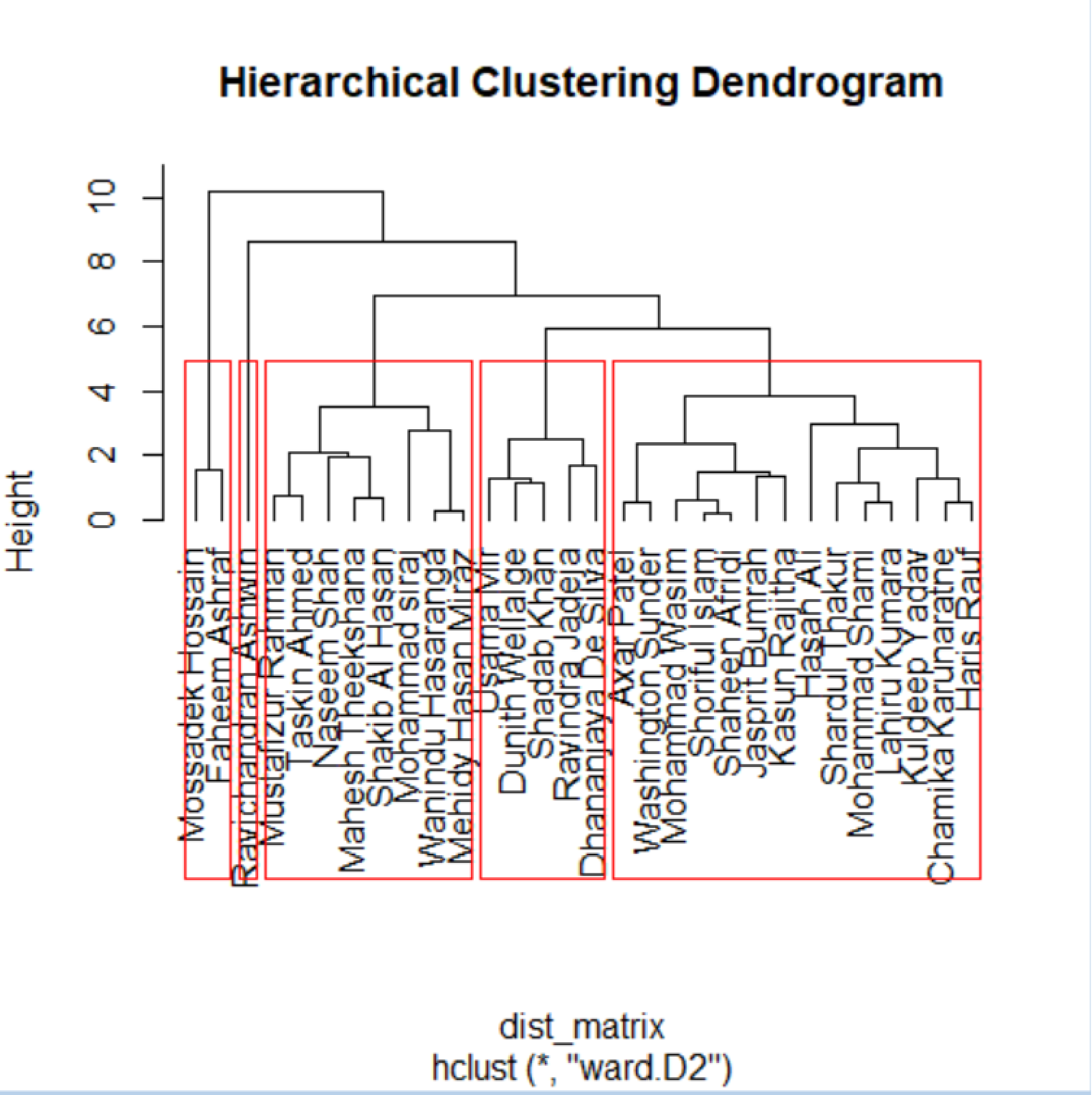

Figure 3.9: **Clustering of men bowlers using hierarchical**

Interpretation: The dendrogram shows how the players are grouped based on the distances between their features. Mosadek Hossain and Faheem Ashraf belongs to first cluster,Ravichandran Ashwin belongs to second cluster and so on. The vertical height of the dendrogram's branches indicates the distance between clusters.

3.3.2 Grouping of women cricketers using hierarchical clustering : all-rounders, batters and bowlers)

Grouping of women all-rounders

The results of hierarchical cluster analysis pertaining to the grouping of women all-rounders are presented below.

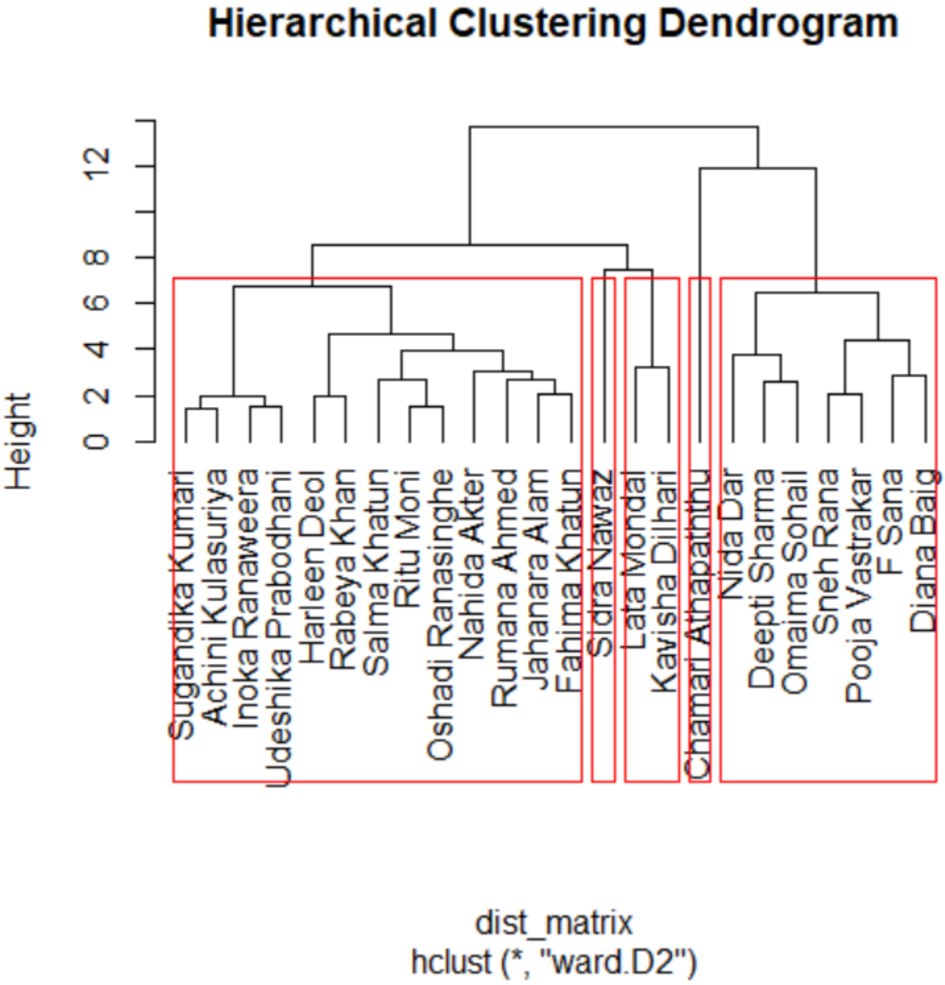

Figure 3.10: **Clustering of women all-rounders using hierarchical clustering**

Interpretation: The dendrogram shows how the players are grouped based on the distances between their features. Sugandika Kumari, Achini Kulasuriya, Inoka Ranaweera and so on belongs to first cluster, Sidra Nawaz belongs to second cluster and so on. The vertical height of the dendrogram branches indicates the distance between clusters.

Grouping of women batters

The results of hierarchical cluster analysis pertaining to the grouping of women batters are presented below.

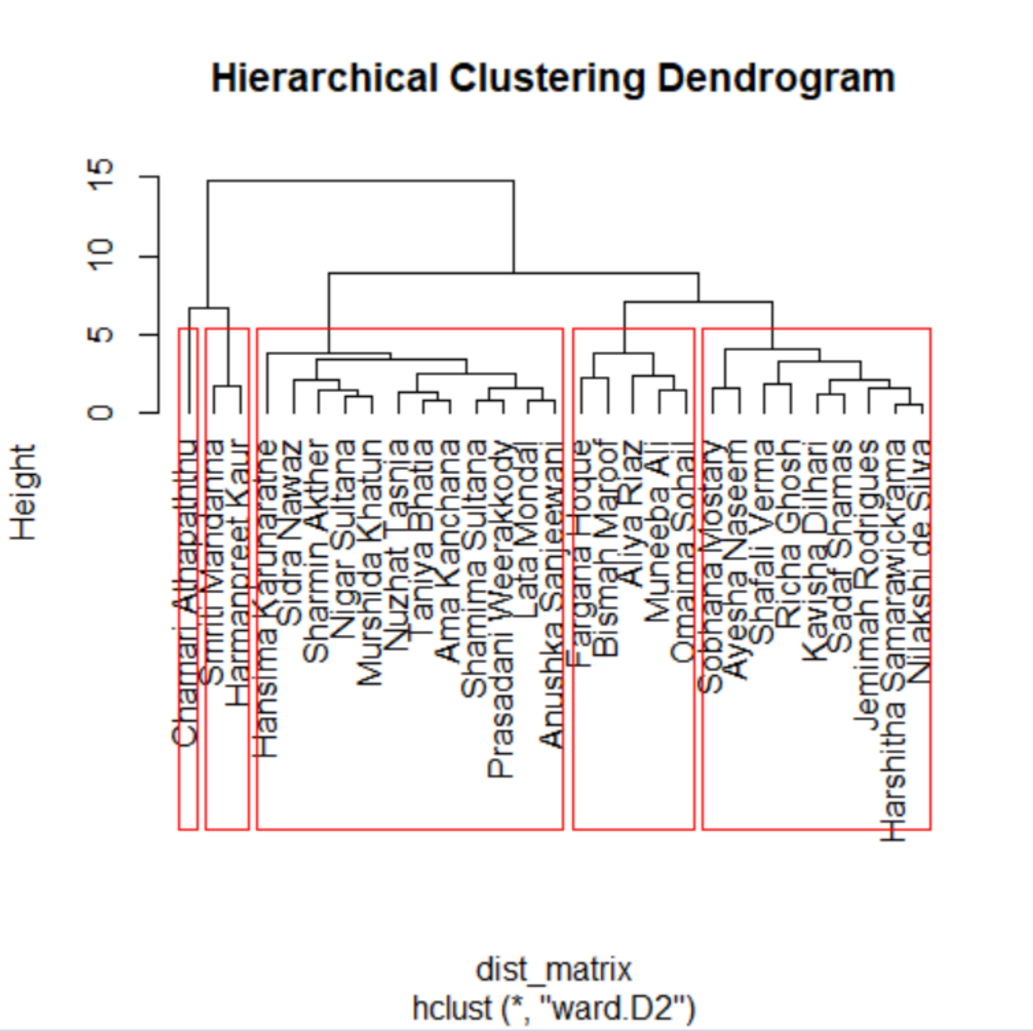

Figure 3.11: **Clustering of women batters using hierarchical clustering**

Interpretation: The dendrogram shows how the players are grouped based on the distances between their features (wickets and economy). Chamari Athapaththu belongs to first cluster, Smriti Mandanna, Harmanpreet Kaur belongs to second cluster and so on. The vertical height of the dendrogram's branches indicates the distance between clusters.

Grouping of men bowlers

The results of hierarchical cluster analysis pertaining to the grouping of men bowlers are presented below.

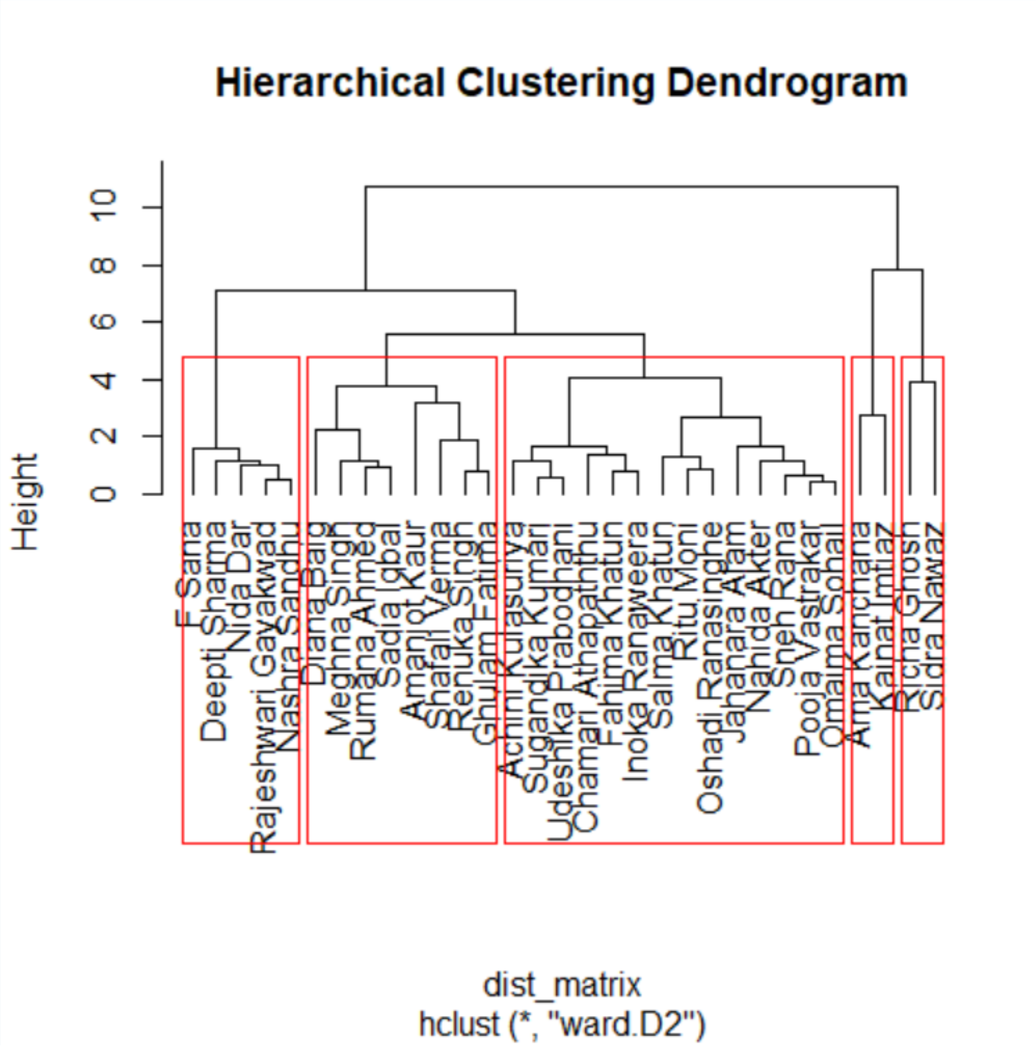

Figure 3.12: **Clustering of women bowlers using hierarchical clustering**

Interpretation: The dendrogram shows how the players are grouped based on the distances between their features . F Sana, Deepti Sharma, Nida Dar, Rajeshwari Gayakwad, and Nashra Sandhu belongs to are first cluster, Diana Biag, Meghna Singh, Rumana Ahmed, Sadia Iqbal, Amanjot Kaur, Shafali Verma, Renuka Singh, and Ghulam Fatima are belongs to second cluster, and so on. The vertical height of the dendrogram's branches indicates the distance between clusters.

3.3.3 Grouping of men cricketers using K-means clustering : all-rounders, batters and bowlers)

Grouping of men all-rounders

The results of K-means cluster analysis pertaining to the grouping of men all-rounders are presented below.

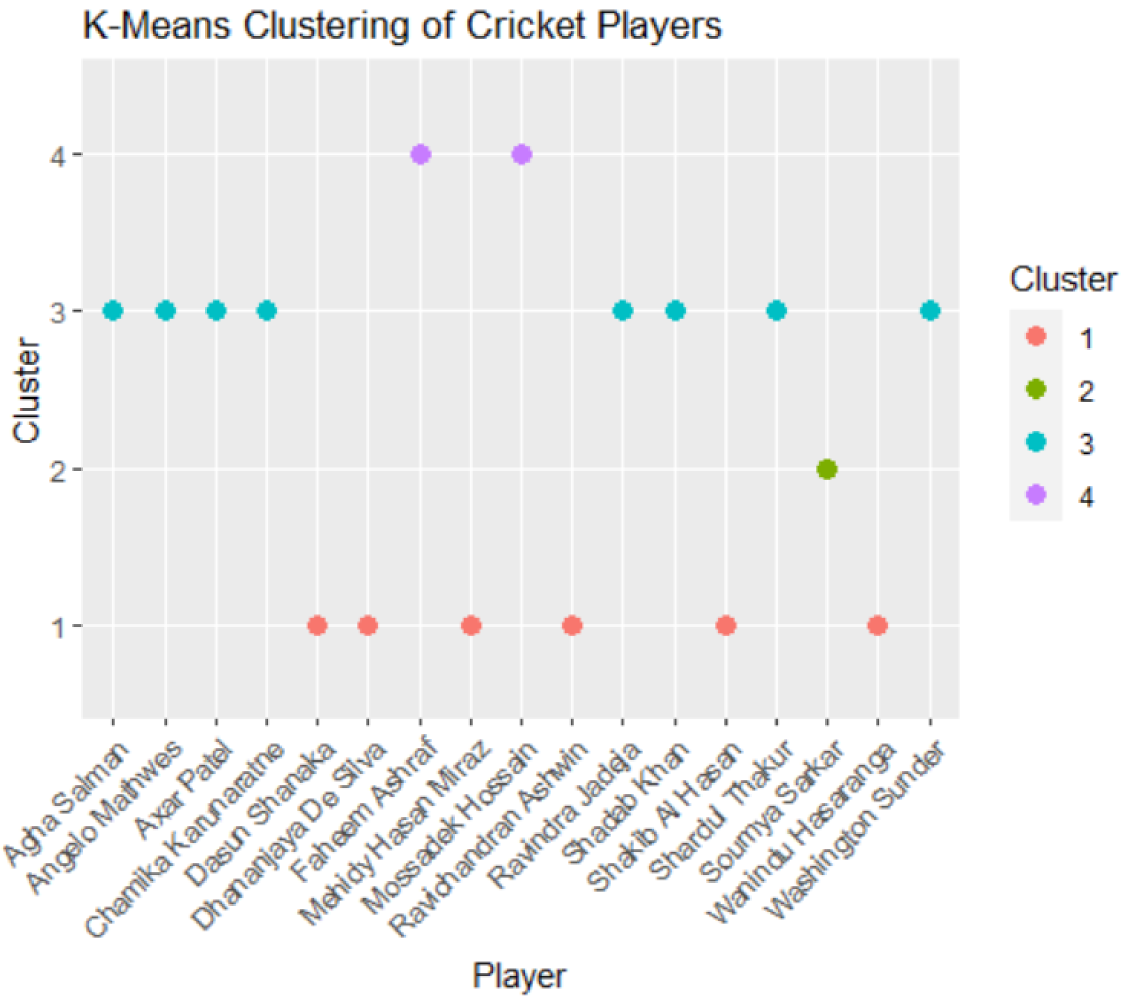

Figure 3.13: **Clustering of men all-rounders using k-means clustering**

Interpretation: The scatter plot shows how players are grouped into clusters based on their cricket statistics. Agha Salman, Angelo Mathews, Axar Patel, Chamika Karunaratne, Faheem Ashraf, Mosaddek Hossain, Ravindra Jadeja, Shadab Khan, Shardul Thakur, Sowmya Sarkar, Washington Sundar belongs to Cluster1 and Dhananjaya De Silva, Shakib Al Hasan, Wanindu Hasaranga belongs to cluster2 and so on Each point on the scatter plot represents a player, and the color indicates the cluster. Players in the same cluster are similar in terms of their cricket statistics.

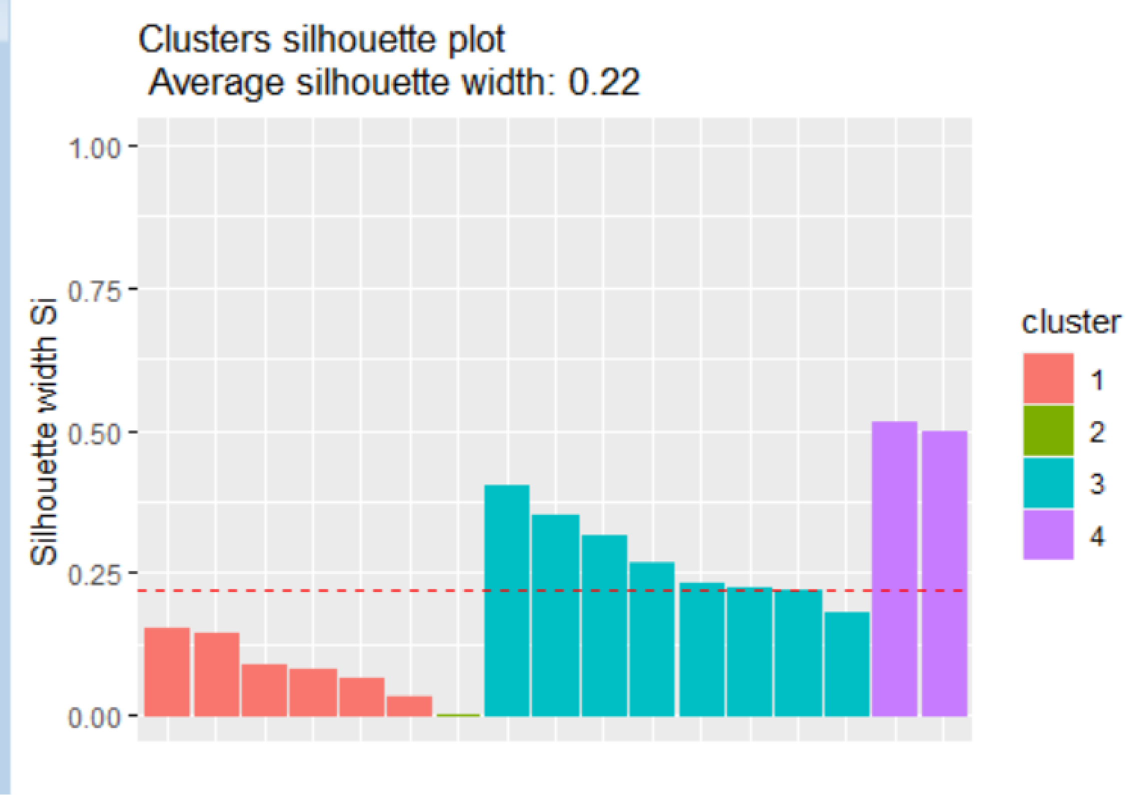

Figure 3.14: **Clustering of men all-rounders using silloutte plot**

Interpretation: Silhouette plot measures the quality of the clustering. Higher silhouette values indicate better separation between clusters. Interpret the silhouette plot: Higher average silhouette width suggests that the clusters are well-separated.

Grouping of men batters

The results of K-means cluster analysis pertaining to the grouping of men batters are presented below.

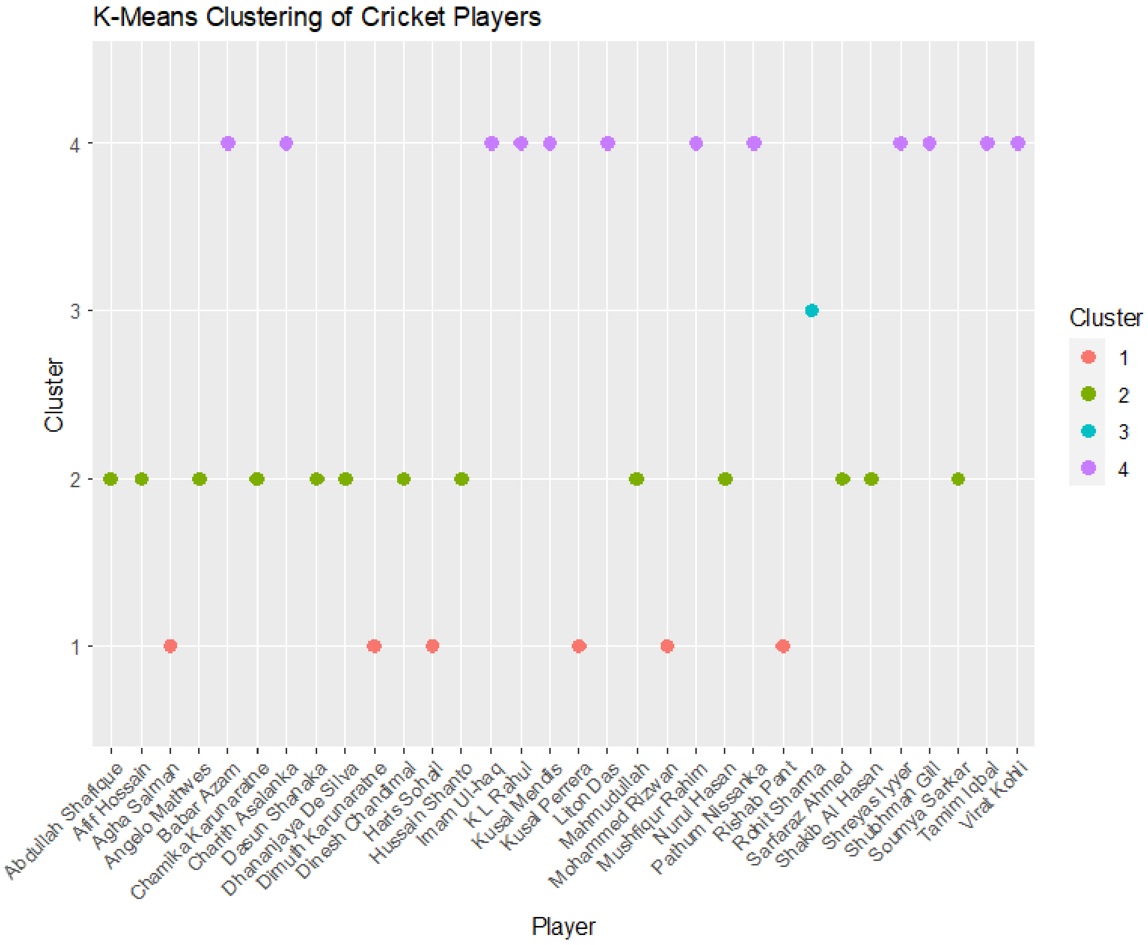

Figure 3.15: **Clustering of men batters using k-means clustering**

Interpretation: The scatter plot shows how players are grouped into clusters based on their cricket statistics. Here Abdullah Shafique, Angelo Mathews, Dinesh Chandimal, Haris Sohail, Nurul Hasan, Sarfaraz Ahmed, and Soumya Sarkar belongs to cluster1 and Afif Hossain, Agha Salman, Chamika Karunaratne,Dimuth Karunaratne and few more belongs to second cluster and so on. Each point on the scatter plot represents a player, and the color indicates the cluster. Players in the same cluster are similar in terms of their cricket statistics.

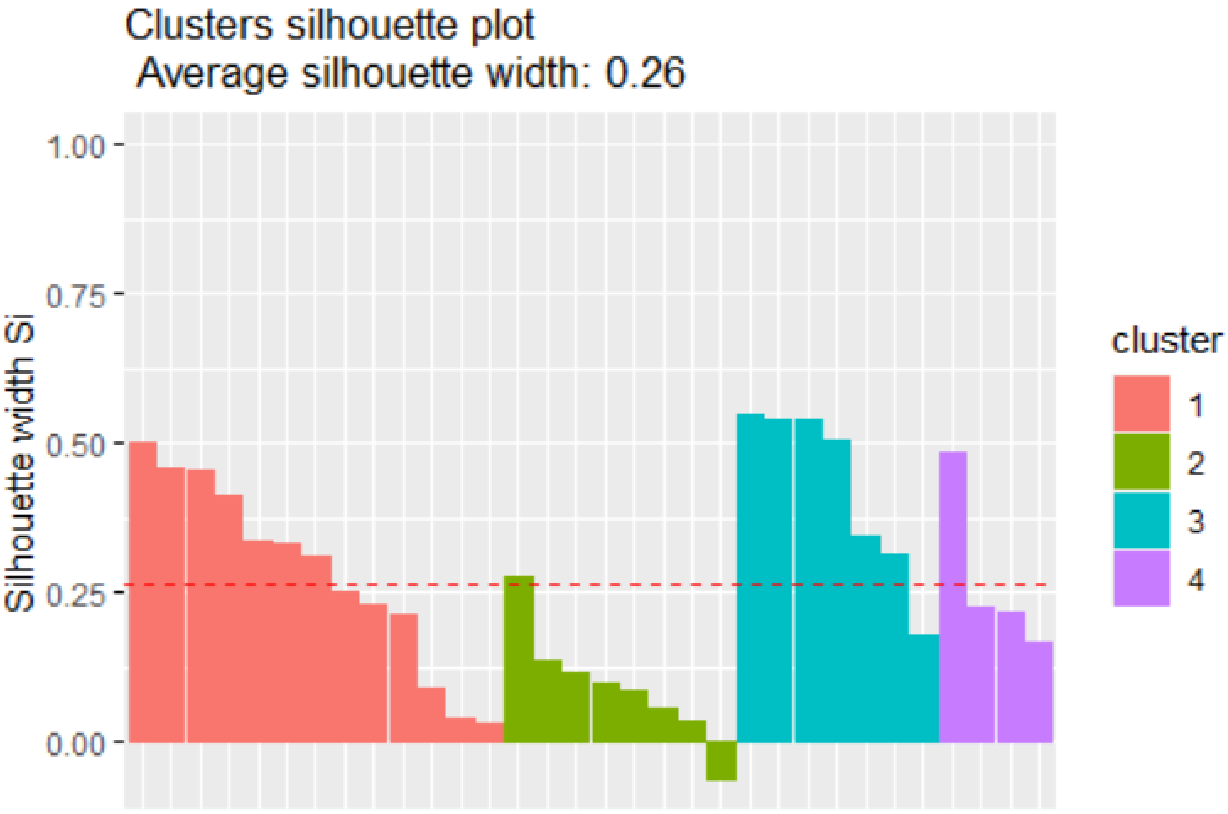

Figure 3.16: **Clustering of men batters using silloutte plot**

Interpretation: Silhouette plot measures the quality of the clustering. Higher silhouette values indicate better separation between clusters. Higher average silhouette width suggests that the clusters are well-separated.

Grouping of men bowlers

The results of K-means cluster analysis pertaining to the grouping of men bowlers are presented below.

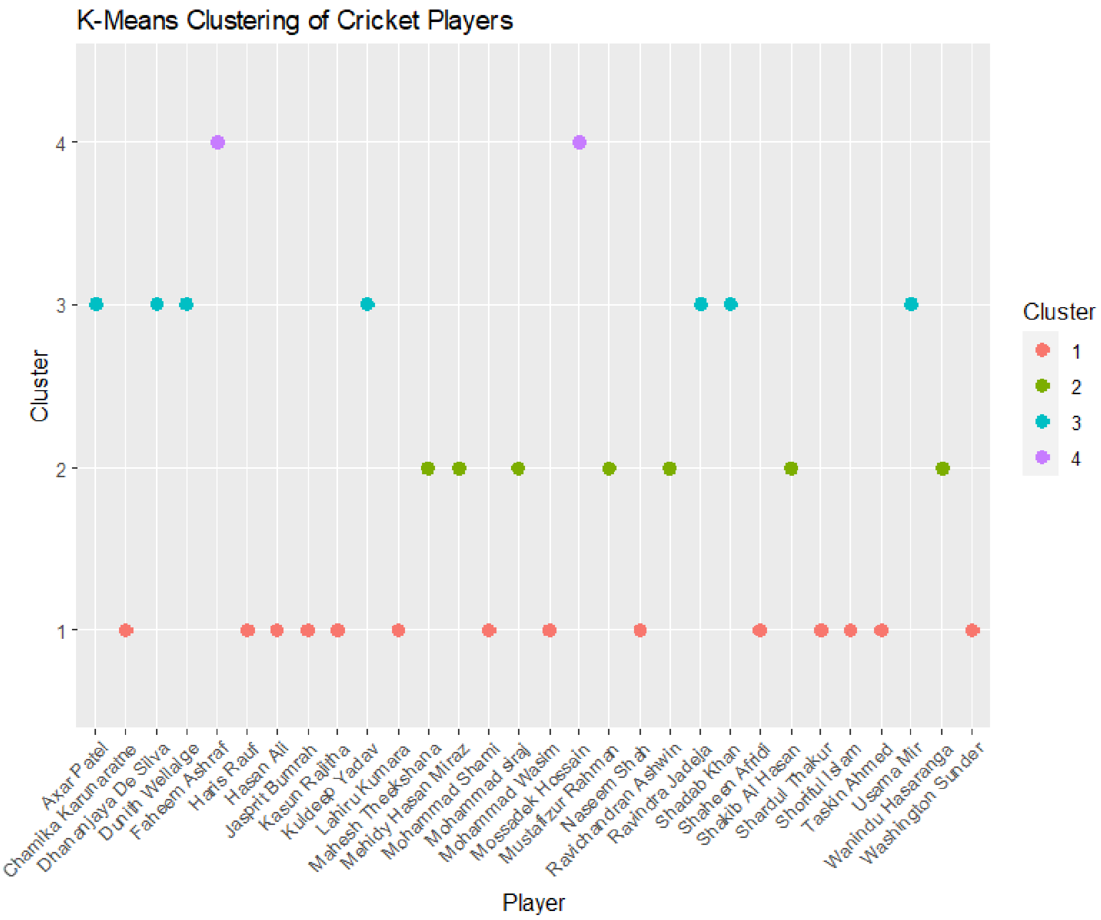

Figure 3.17: **Clustering of men bowlers using k-means clustering**

Interpretation: The scatter plot shows how players are grouped into clusters based on their cricket statistics. Chamika Karunaratne, Dhananjaya De Silva, Dunith Wellalge, Haris Rauf, Hasan Ali, Kuldeep Yadav, Ravindra Jadeja, Shadab Khan, Shardul Thakur, Usama Mir belongs to first cluster and Ravichandran Ashwin alone belongs to cluster 2 Each point on the scatter plot represents a player, and the color indicates the cluster. Players in the same cluster are similar in terms of their cricket statistics.

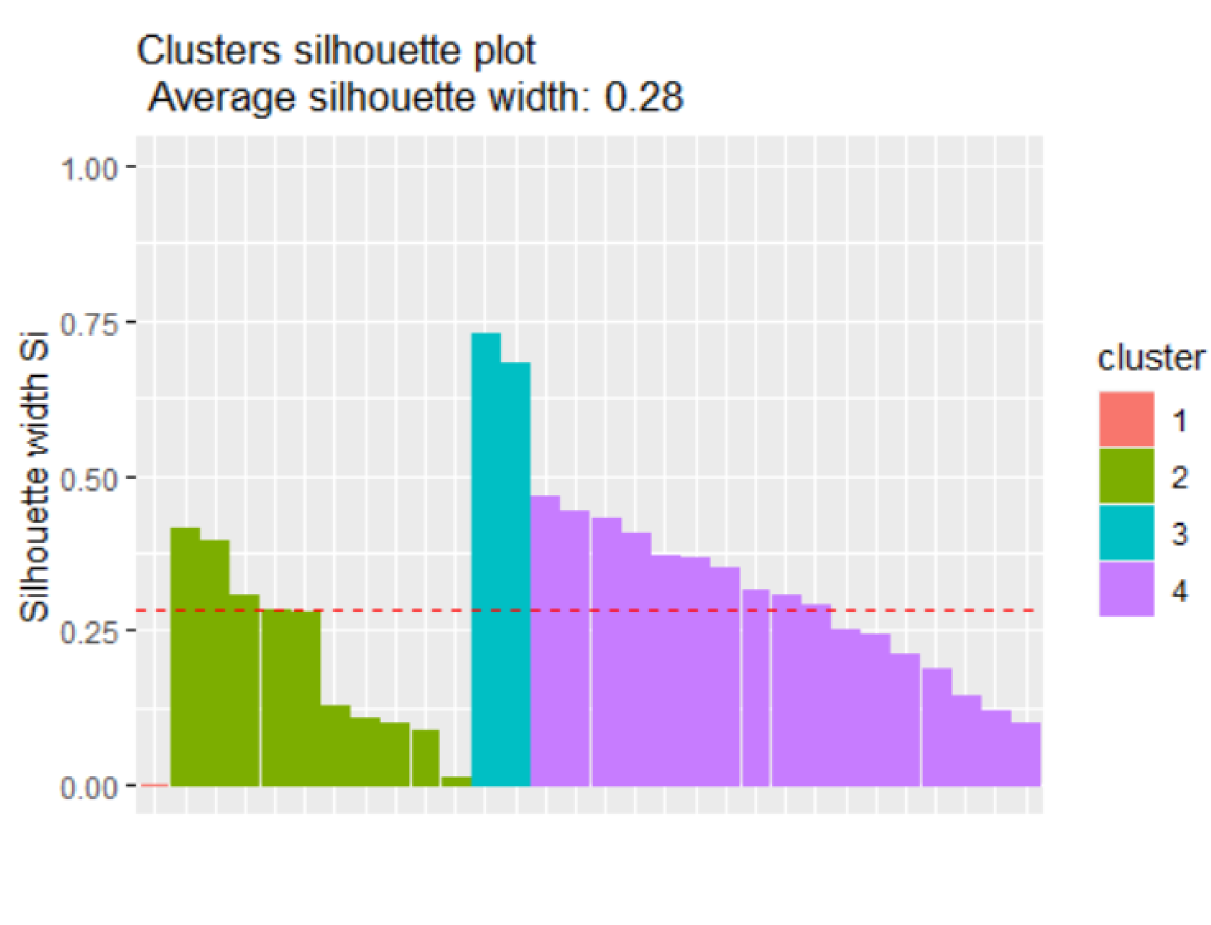

Figure 3.18: **Clustering of men bowlers using silloutte plot**

Interpretation: Silhouette plot measures the quality of the clustering. Higher silhouette values indicate better separation between clusters. Higher average silhouette width suggests that the clusters are well-separated.

3.3.4 Grouping of women cricketers using K-means clustering : all-rounders, batters and bowlers)

Grouping of women all-rounders

The results of K-means cluster analysis pertaining to the grouping of women all-rounders are presented below.

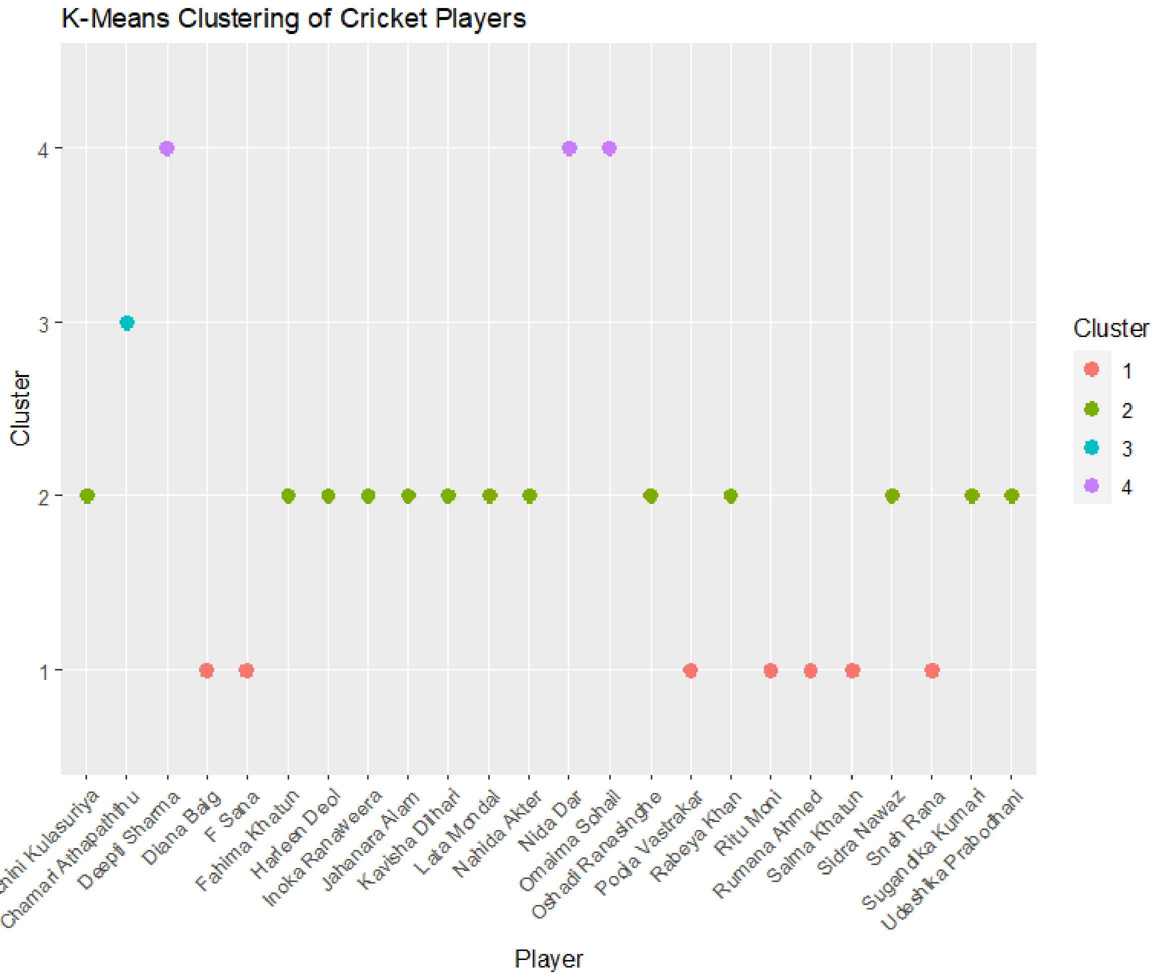

Figure 3.19: **Clustering of women all-rounders using k-means clustering**

Interpretation: The scatter plot shows how players are grouped into clusters based on their cricket statistics. Chamari Athapaththu alone belongs to cluster 1 and Fathima Khatun, Harleen Deol, Jahanara Alam, Kavisha Dilhari, Oshado Ranasinghe, Rubeya Khan, Ritu Moni, Rumana Ahmed, Salma Khatun, and Sneh Rana belongs to cluster 2 and so on. Each point on the scatter plot represents a player, and the color indicates the cluster. Players in the same cluster are similar in terms of their cricket statistics.

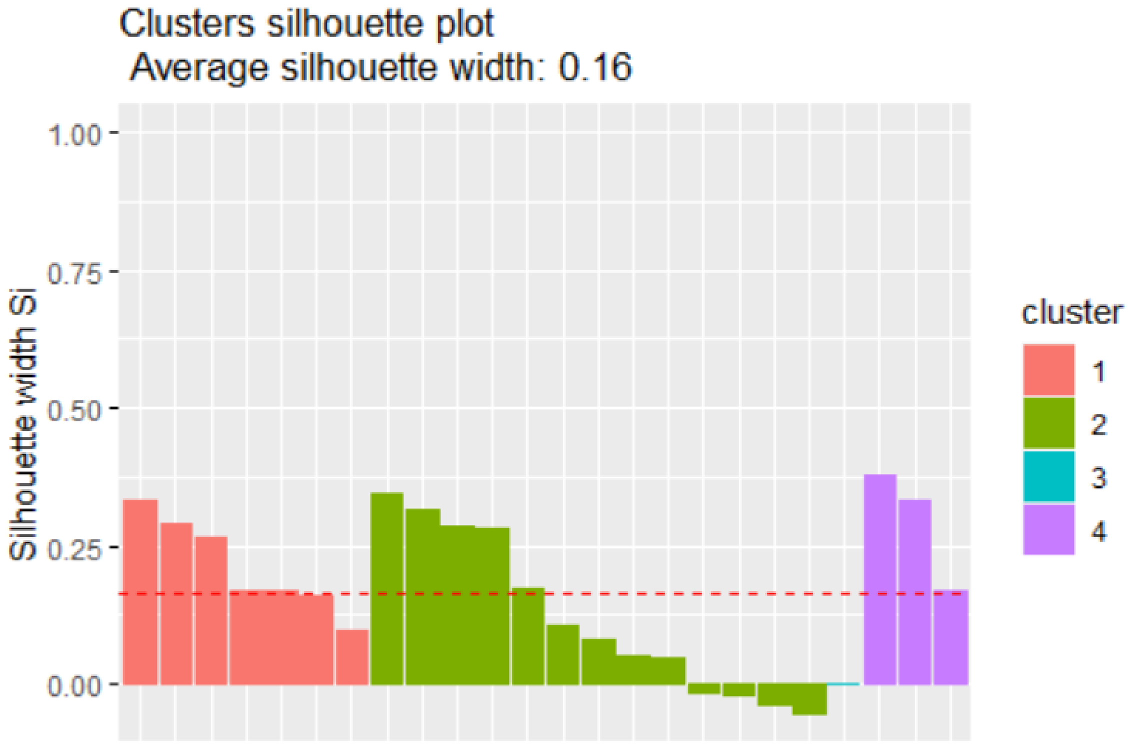

Figure 3.20: **Clustering of women all-rounders using silloutte plot**

Interpretation: Silhouette plot measures the quality of the clustering. Higher silhouette values indicate better separation between clusters. Higher average silhouette width suggests that the clusters are well-separated.

Grouping of women batters

The results of K-means cluster analysis pertaining to the grouping of women batters are presented below.

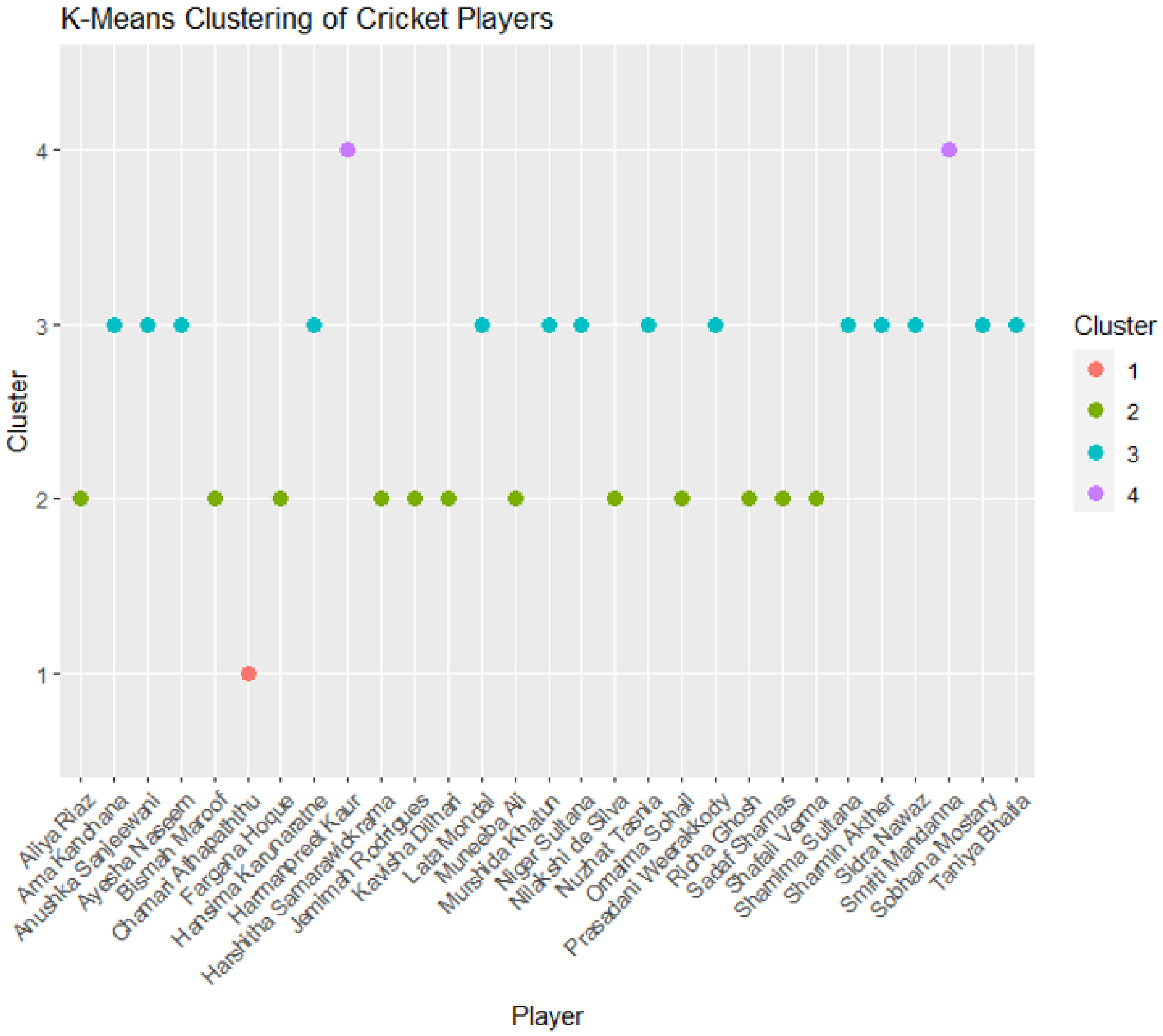

Figure 3.21: **Clustering of women batters using k-means clustering**

Interpretation: The scatter plot shows how players are grouped into clusters based on their cricket statistics. Anushka Sanjeevini, Ayesha Naseem, Harshitha Samarawickrama, Jemmaih Rodrigues, Kavisha Dilhari, Nilakshi De Silva, Richa Ghosh, Sadaf Shamas, Shobana Mostary belongs to cluster 1 and Chamari Atapaththu, Harmanpreet Kaur and Smriti Mandhanna belongs to cluster 2 Each point on the scatter plot represents a player, and the color indicates the cluster. Players in the same cluster are similar in terms of their cricket statistics.

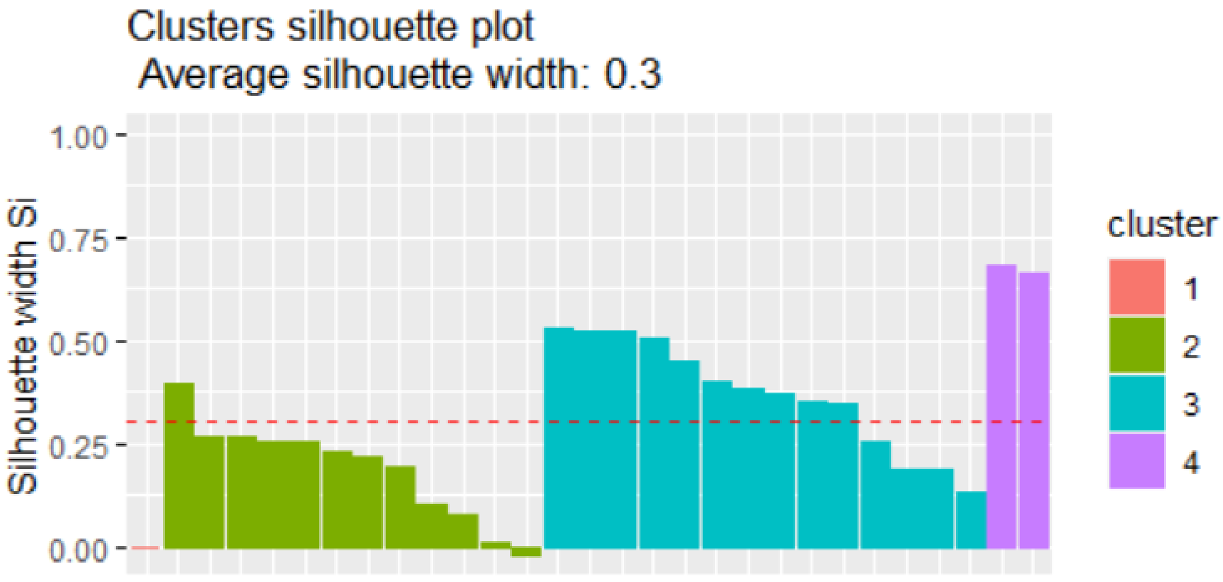

Figure 3.22: **Clustering of women batters using silloutte plot**

Interpretation: Silhouette plot measures the quality of the clustering. Higher silhouette values indicate better separation between clusters. Higher average silhouette width suggests that the clusters are well-separated.

Grouping of women bowlers

The results of K-means cluster analysis pertaining to the grouping of women bowlers are presented below.

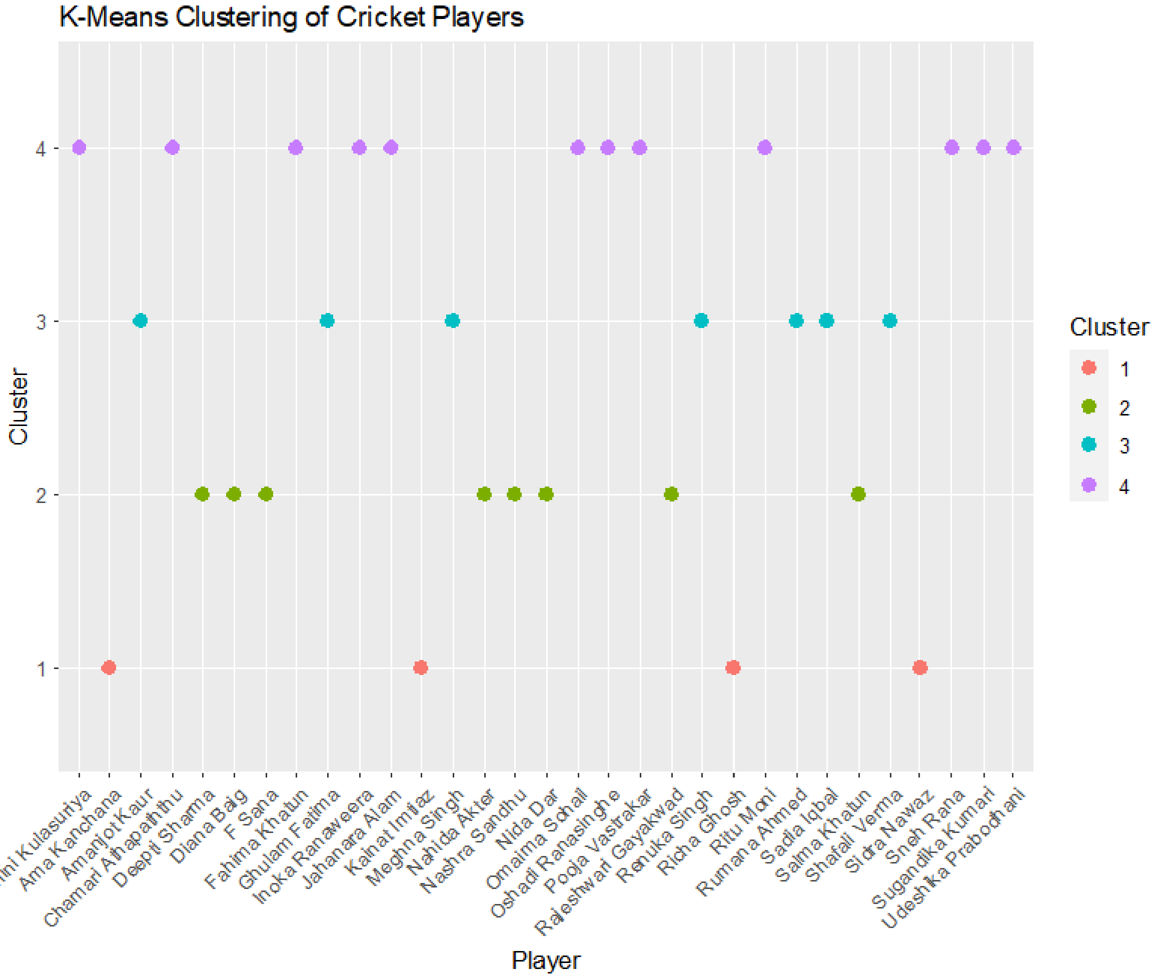

Figure 3.23: **Clustering of women bowlers using k-means clustering**

Interpretation: The scatter plot shows how players are grouped into clusters based on their cricket statistics. Deepti Sharma, Diana Biag, F Sana, Nashra Sandhu, Nida Dar, and Rajeshwari Gayakwad belongs to cluster 1 and Achini Kulasuriya, Amanjot Kaur, Ghulam Fatima, Meghna Singh, Renuka Singh, Rumana Ahmed, Sadia Iqbal, Shefali Varma, Sugandika Kumari, and Udeshika Prabodhani belongs to cluster2 and so on Each point on the scatter plot represents a player, and the color indicates the cluster. Players in the same cluster are similar in terms of their cricket statistics.

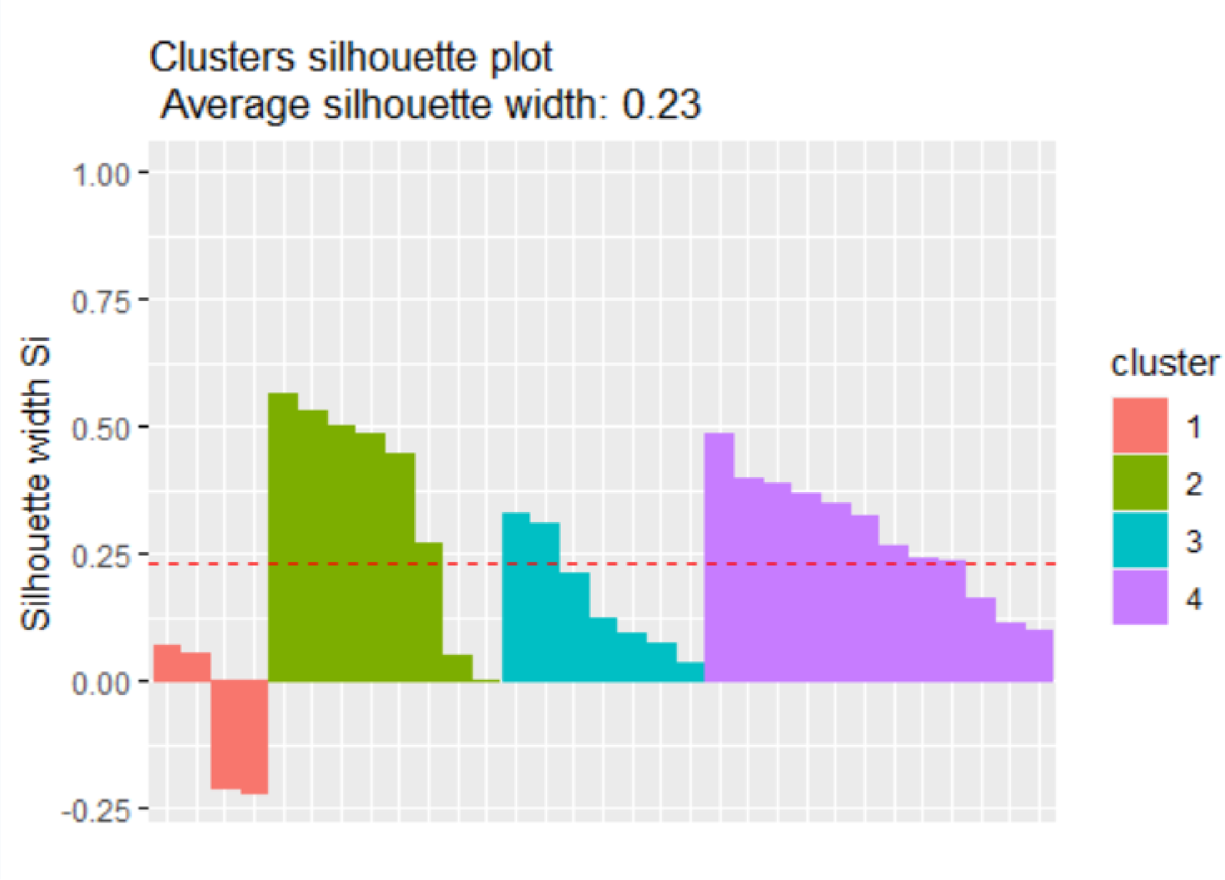

Figure 3.24: **Clustering of women bowlers using silloutte plot**

Interpretation: Silhouette plot measures the quality of the clustering. Higher silhouette values indicate better separation between clusters. Higher average silhouette width suggests that the clusters are well-separated.

3.4 Classification of cricket players using machine learning techniques

The machine learning techniques as we discussed in section **??** used to classify the players:

3.4.1 Classification of men cricketers based on accuracy measures using confusion matrix

The results of machine learning techniques pertaining to the classification of men cricketers are presented below.

Classification Technique	**Accuracy Measure**
SVM	0.90
Neural Networks	0.66
Decision Trees	0.94
Multinomial Regression	0.95
LDA	0.93

Table 3.13: **Classification techniques and its accuracy measures**

Interpretation: From the above table we can see that the multinomial regression model is more accurate than any of those mentioned above.

3.4.2 Classification of women cricketers based on accuracy measures using confusion matrix

The results of machine learning techniques pertaining to the classification of women cricketers are presented below.

Classification Technique	**Accuracy Measure**
SVM	0.87
Neural Networks	0.48
Decision Trees	0.88
Multinomial Regression	0.89
LDA	0.76

Table 3.14: **Classification techniques and its accuracy measures**

Interpretation: From the above table we can see that the multinomial regression model is more accurate than any of those mentioned above.

3.5 Comparision of overall performance of a players across different continents using MANOVA

The statistical technique as we discussed in section 2.7 for comparision of overall performance of a players across different continents.

3.5.1 Comparision of overall performance of men cricketrs using MANOVA

The results of MANOVA pertaining to the comparision of overall performance of a men cricketrs are presented below.

	Df	Pillai value	Approx F	Num DF	Den Df	Pr(≥F)
Continents	3	0.6969	1.5511	24	123	0.06382
Residuals	46					

Table 3.15: **Summary of overall performance of men players using manova across different continent**

Interpretation:In this case, the p-value for the continents is 0.06382, which is greater than the commonly used significance level of 0.05. This means that there is not enough evidence to conclude that there are significant differences between the means of the groups on the dependent variables.

Summary of ANOVA model for men players across different continents

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	655788	218596	0.595	0.622
Residuals	46	16907886	367563		

Table 3.16: **Summary of runs of men players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	1706	568.6	0.9	0.448
Residuals	46	29056	631.7		

Table 3.17: **Summary of wickets of men players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	120	39.98	0.16	0.923
Residuals	46	11482	249.60		

Table 3.18: **Summary of average of men players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	1000	333.5	0.815	0.492
Residuals	46	18820	409.1		

Table 3.19: **Summary of strike rate of men players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	1988	662.7	1.43	0.246
Residuals	46	21316	463.4		

Table 3.20: **Summary of bowling average of men players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	3059	1019.8	1.279	0.293
Residuals	46	36667	797.1		

Table 3.21: **Summary of bowling strike rate of men players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	1457	485.7	0.677	0.57
Residuals	46	32983	717.1		

Table 3.22: **Summary of bowling dot percentage of men players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	371	123.5	0.841	0.478
Residuals	46	6758	146.9		

Table 3.23: **Summary of catches of men players using anova for different continents**

Interpretation: From the above tables we can observe based on the p-values reported in these above summaries, it appears that there is no significant difference in the means of any of the dependent variables (runs, wickets, average, strike rate, bowling average, bowling strike rate, bowling dot percentage, catches).

3.5.2 Comparision of overall performance of women cricketrs across different continents using MANOVA

The results of MANOVA pertaining to the comparision of overall performance of a women cricketrs are presented below.

	Df	Pillai value	Approx F	Num DF	Den Df	Pr(≥F)
Continents	3	0.70946	1.5874	24	123	0.05438
Residuals	46					

Table 3.24: **Summary of overall performance of women players using manova across different continent**

Interpretation: In this case, the p-value is greater than 0.05, so we can say that there is no statistically significant differences between the group means for the dependent variables across the levels of continents.

Summary of ANOVA model for women players across different continents

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	5305093	1768364	4.928	0.00473
Residuals	46	16505180	358808		

Table 3.25: **Summary of runs of women players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	2866	955.2	2.358	0.084
Residuals	46	18638	405.2		

Table 3.26: **Summary of wickets of women players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	371	123.5	0.841	0.478
Residuals	46	21832	474.6		

Table 3.27: **Summary of average of women players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	7593	2530.8	6.141	0.00133
Residuals	46	18956	412.1		

Table 3.28: **Summary of strike rate of women players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	4286	1428	2.174	0.104
Residuals	46	30221	657		

Table 3.29: **Summary of bowling average of women players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	2195	731.5	1.35	0.27
Residuals	46	24930	542.0		

Table 3.30: **Summary of bowling strike rate of women players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	252	83.9	0.134	0.939
Residuals	46	28738	624.7		

Table 3.31: **Summary of bowling dot percentage of women players using anova for different continents**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Continents	3	165	54.85	0.701	0.556
Residuals	46	3598	78.22		

Table 3.32: **Summary of catches of women players using anova for different continents**

Interpretation: From the above tables, we can say that there is a statistically significant difference between the means of the groups for the dependent variables runs and strike rate (p-value of 0.00473 and 0.00133 respectively). For the other dependent variables (wickets, average, bowling average, bowling strike rate, bowling dot percentage and catches), there is no statistically significant difference between the means of the groups(p-value greater than 0.05).

3.6 Comparision of overall performance of a players in asia continent using MANOVA

The statistical technique as we discussed in section for comparision of overall performance of a players in asia continent.

3.6.1 Comparision of overall performance of men cricketers in asia continent using MANOVA

The results of MANOVA pertaining to the comparision of overall performance of a men cricketers in asia continent are presented below.

	Df	Pillai value	Approx F	Num DF	Den Df	Pr(≥F)
Country	1	0.2457	2.0358	8	50	0.0609
Residuals	57					

Table 3.33: **Summary of overall performance using manova of men players in asia continent**

Interpretation: From the above table the p-value is greater than the commonly used significance level of 0.05, indicating that there is not enough evidence to conclude that there is a significant difference in the means of the dependent variables between the different levels of the independent variable COUNTRY.

Summary of ANOVA model for men players in asia continent

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	1	994862	994862	3.5852	0.06338
Residuals	57	15817036	277492		

Table 3.34: **Summary of runs of men players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	1	736.8	736.8	2.6987	0.1059
Residuals	57	15563.3	273.04		

Table 3.35: **Summary of average of men players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	1	0.6	0.6	0.0013	0.9716
Residuals	57	28848	506.12		

Table 3.36: **Summary of strike rate of men players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	1	2202	2202	3.2358	0.07734
Residuals	57	38792	680.57		

Table 3.37: **Summary of wickets of men players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	1	1287	1286.6	0.7897	0.3779
Residuals	57	92862	1629.2		

Table 3.38: **Summary of bowling average of men players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	1	484	484.15	0.6387	0.4275
Residuals	57	38100	668.42		

Table 3.39: **Summary of bowling strike rate of men players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	1	470	469.79	0.7028	0.4053
Residuals	57	38100	668.42		

Table 3.40: **Summary of bowling dot percenta of men players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	1	155.2	155.169	1.6328	0.2065
Residuals	57	5416.9	95.033		

Table 3.41: **Summary of catches of men players using anova for asia continent**

Interpretation: From the above tables, statistical analysis suggests that there is no significant difference in performance metrics such as runs, average, strike rate, wickets, bowling average, bowling strike rate, bowling dot percentage, and catches among players from different countries.

3.6.2 Comparision of overall performance of women cricketers in asia continent using MANOVA

The results of MANOVA pertaining to the comparision of overall performance of a women cricketers in asia continent are presented below.

	Df	Pillai value	Approx F	Num DF	Den Df	Pr(≥F)
Country	3	0.5451	1.4155	24	153	0.1079
Residuals	56					

Table 3.42: **Summary of overall performance using manova of women players**

Interpretation: From the above table p-value associated with the multivariate test. For ‘COUNTRY’, it is 0.1079. A common threshold for significance is 0.05, so in this case, we would not reject the null hypothesis as 0.1079 ¿ 0.05, indicating that the country variable does not have a significant effect on the dependent variables at this level.

Summary of ANOVA model for women players in asia continent

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	3	305819	101940	1.3651	0.2628
Residuals	56	4181844	74676		

Table 3.43: **Summary of runs of women players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	3	66.6	22.215	0.1238	0.9457
Residuals	56	10046.7	179.405		

Table 3.44: **Summary of average of women players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	3	11529	3842.9	1.4324	0.243
Residuals	56	150234	2682.8		

Table 3.45: **Summary of strike rate of women players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	3	646.2	215.39	1.5983	0.2
Residuals	56	7546.8	134.76		

Table 3.46: **Summary of wickets of women players using anova for asia continent**

	Df	Sum sq	Mean sum sq	F-value	Pr(≥F)
Country	3	10970	3656.5	1.6469	0.1889
Residuals	56	124333	2220.2		

Table 3.47: **Summary of bowling average of women players using anova for asia continent**

	Df	**Sum sq**	**Mean sum sq**	**F-value**	**Pr(≥F)**
Country	3	9527	3175.6	1.4225	0.2458
Residuals	56	125013	2232.4		

Table 3.48: **Summary of bowling strike rate of women players using anova for asia continent**

	Df	**Sum sq**	**Mean sum sq**	**F-value**	**Pr(≥F)**
Country	3	3555	1185	1.7332	0.1706
Residuals	56	38287	683.7		

Table 3.49: **Summary of bowling dot percentage of women players using anova for asia continent**

	Df	**Sum sq**	**Mean sum sq**	**F-value**	**Pr(≥F)**
Country	3	345.93	115.311	3.5637	0.01971
Residuals	56	1812	32.357		

Table 3.50: **Summary of catches of women players using anova for asia continent**

Interpretation: From the above tables,it appears that there is no significant difference in the means of most of the response variables between the levels of the predictor variable COUNTRY at a significance level of 0.05. The only exception is for the response variable CATCHES, where there is a significant difference in means between the levels of COUNTRY at a significance level of 0.05.

3.7 Prediction of an match outcome using machine learning techniques

The machine learning techniques as we discussed in section 2.6 used to predict the match outcome:

3.7.1 Prediction of an match outcome using ROC curves

The results of machine learning techniques pertaining to the prediction of an match outcome are presented below.

Predictive Model	Area Under Curve
KNN	0.5
Naive Bayes	0.55
SVM	0.67
Random Forest	0.57
Logistic Regression	0.56
Gradient Boosting	0.82

Table 3.51: **Predictive models and its AUC values**

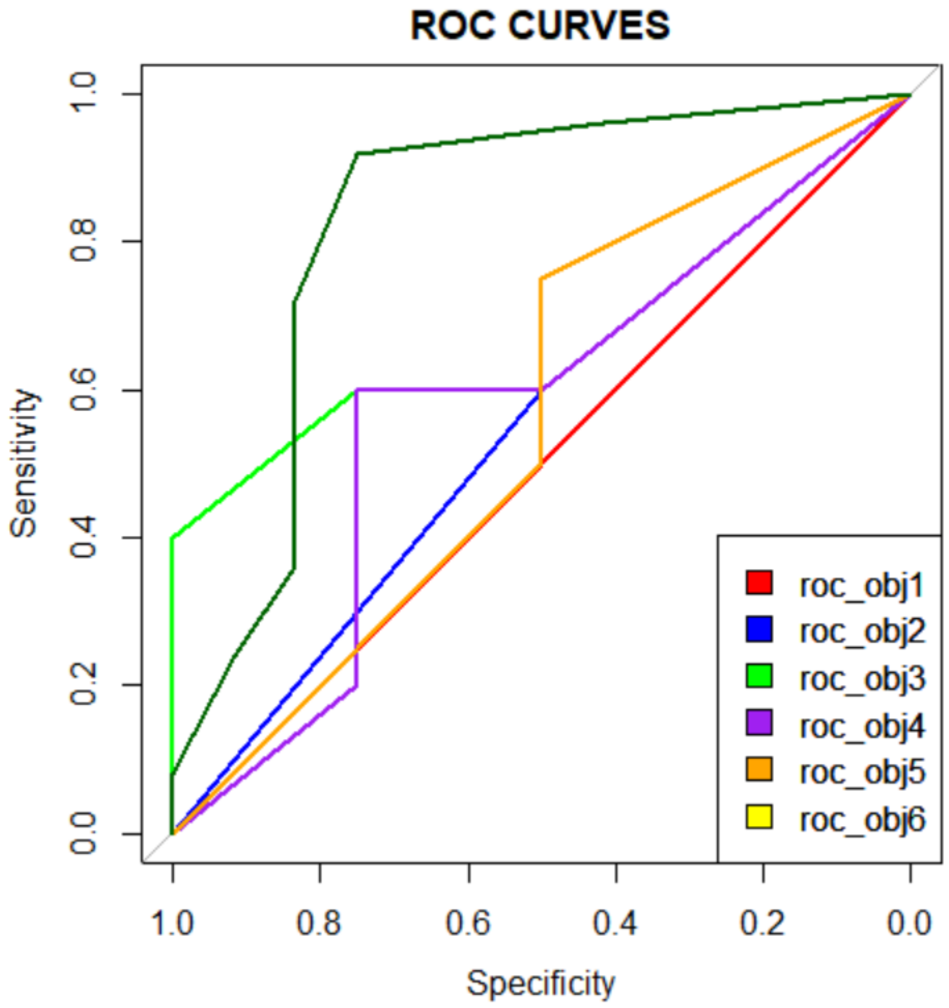

Figure 3.25: **ROC curves for prediction of an match outcome**

Interpretation: From The above table and ROC curves we can observe that gradient boosting is the best model for prediction of a match outcome.

3.8 Odds ratio

The statistical technique as we discussed in section **??** to check the association between match outcome and other variables.

3.8.1 Association between Toss and Match Result

The results of odds ratio pertaining to the association between toss and match result are presented below.

	Match Result	
Toss	Win	Loss
Win	4	12
Loss	7	7

Table 3.52: **Contigency table for checking association between toss and match result**

Interpretation:The obtained odds ratio value is 0.3333333 and it is interpreted as the odds of winning the match are lower for the team that wins the toss.

3.8.2 Association between batting first and Match Result

The results of odds ratio pertaining to the association between batting first and match result are presented below.

	Match Result	
batting first	Win	Loss
Win	6	5
Loss	7	12

Table 3.53: **Contigency table for checking association between batting first and match result**

Interpretation: The obtained odds ratio value is 2.057143 and it is interpreted as the odds of winning the match are higher for the team that bat first.

3.8.3 Association between and Match Result

The results of odds ratio pertaining to the association between home and match result are presented below.

	Match Result	
Home	Win	Loss
Win	9	2
Loss	12	7

Table 3.54: **Contigency table for checking association between home and match result**

Interpretation:The obtained odds ratio value is 2.625 and it is interpreted as the odds of winning the match are higher for the team that played at home.

Chapter 4

Conclusions

The conclusions and discussions about this project are given below:

1. In the given table of 135 players, ”Naseem Shah” emerged as the top-ranked all-rounder, with ”Jayden Seales” and ”Jhye Richardson” following closely in the men’s rankings.

2. In the given table of 80 players, ”Wiaan Mulder ” emerged as the top-ranked batter, with ”Abdullah Shafique” and ”Sarfaraz Ahmed” following closely in the men’s rankings.

3. In the given table of 43 players, ”Wanindu Hasaranga” emerged as the top-ranked bowler, with ”Ravichandran Ashwin” and ”Mohammed Siraj” following closely in the men’s rankings.

4. In the given table of 135 players, ”Inoshi Priyadarshini” emerged as the top-ranked all-rounder, with ”Radha Yadav” and ”Raisibe Ntozakhe” following closely in the women’s rankings.

5. In the given table of 80 players, ”A Khaka ” emerged as the top-ranked batter, with ”Taniya Bhatia” and ” S Molineux” following closely in the women’s rankings.

6. In the given table of 43 players, ”Amanjot Kaur” emerged as the top-ranked bowler, with ”K Garth” and “Freya Kemp” following closely in the women’s rankings.

7. The dendrogram illustrates player groupings based on feature distances, with Ravichandran Ashwin in the first cluster Dasun Shanaka and Mehidy Hasan Miraz in the second cluster, and so on, showcasing the hierarchical relationships among them.

8. The dendrogram illustrates player groupings based on feature distances, with Nurul Hasan, Abdullah Shafique, Sarfaraz Ahmed, Soumya Sarkar, Angelo Mathews, and Dinesh Chandimal in the first cluster and Rishab Pant, Kusal Perera, Haris Sohail, and Agha Salman in the second cluster and so on, showcasing the hierarchical relationships among them.

9. The dendrogram illustrates player groupings based on feature distances, with Nurul Hasan, Abdullah Shafique, Sarfaraz Ahmed, Soumya Sarkar, Angelo Mathews, and Dinesh Chandimal in the first cluster and Rishab Pant, Kusal Perera, Haris Sohail, and Agha Salman in the second cluster and so on, showcasing the hierarchical relationships among them.

10. The dendrogram illustrates player groupings based on feature distances, with Sugandika Kumari, Achini Kulasuriya, Inoka Ranaweera, and so on in the first cluster and Sidra Nawaz in the second cluster and so on, showcasing the hierarchical relationships among them.

11. The dendrogram illustrates player groupings based on feature distances, with Chamari Athapaththu in the first cluster and Smriti Mandanna, and Harmanpreet Kaurin in the second cluster, and so on, showcasing the hierarchical relationships among them.

12. The dendrogram illustrates player groupings based on feature distances, with F Sana, Deepti Sharma, Nida Dar, Rajeshwari Gayakwad, and Nashra Sandhu in the first cluster and Diana Biag, Meghna Singh, Rumana Ahmed, Sadia Iqbal, Amanjot Kaur, Shefali Verma, Renuka Singh, and Ghulam Fatima in the second cluster and so on, showcasing the hierarchical relationships among them.

13. Players in Cluster 1 (Agha Salman, Angelo Mathews, Axar Patel, Chamika Karunaratne, Faheem Ashraf, Mosaddek Hossain, Ravindra Jadeja, Shadab Khan, Shardul Thakur, Sowmya Sarkar, Washington Sundar) exhibit similar cricket statistics, while players in Cluster 2 (Dhananjaya De Silva, Shakib Al Hasan, Wanindu Hasaranga) share comparable cricket performance characteristics.

14. Players in Cluster 1 (Abdullah Shafique, Angelo Mathews, Dinesh Chandimal, Haris Sohail, Nurul Hasan, Sarfaraz Ahmed, and Soumya Sarkar) exhibit similar cricket statistics, while players in Cluster 2 (Afif Hossain, Agha Salman, Chamika Karunaratne, Dimuth Karunaratne and few more) share comparable cricket performance characteristics.

15. Players in Cluster 1(Chamika Karunaratne, Dhananjaya De Silva, Dunith Wellalge, Haris Rauf, Hasan Ali, Kuldeep Yadav, Ravindra Jadeja, Shadab Khan, Shardul Thakur, Usama Mir) exhibit similar cricket statistics, while players in Cluster 2 (Ravichandran Ashwin alone) share comparable cricket performance characteristics.

16. Players in Cluster 1 (Chamari Athapaththu alone) exhibit similar cricket statistics, while players in Cluster 2 (Fathima Khatun, Harleen Deol, Jahanara Alam, Kavisha Dilhari, Oshado Ranasinghe, Rubeya Khan, Ritu Moni, Rumana Ahmed, Salma Khatun, and Sneh Rana) share comparable cricket performance characteristics.

17. Players in Cluster 1 (Anushka Sanjeevini, Ayesha Naseem, Harshitha Samarawickrama, Jemmaih Rodrigues, Kavisha Dilhari, Nilakshi De Silva, Richa Ghosh, Sadaf Shamas, Shobana Mostary) exhibit similar cricket statistics, while players in Cluster 2 (Chamari Atapaththu, Harmanpreet Kaur and Smriti Mandhanna) share comparable cricket performance characteristics.

18. Players in Cluster 1 (Deepti Sharma, Diana Biag, F Sana, Nashra Sandhu, Nida Dar, and Rajeshwari Gayakwad) exhibit similar cricket statistics, while players in Cluster 2 (Achini Kulasuriya, Amanjot Kaur, Ghulam Fatima, Meghna Singh, Renuka Singh, Rumana Ahmed, Sadia Iqbal, Shefali Varma, Sugandika Kumari, and Udeshika Prabodhani) share comparable cricket performance characteristics.

19. The multinomial regression model demonstrates superior accuracy compared to the other models discussed in the table for men classification.

20. The multinomial regression model demonstrates superior accuracy compared to the other models discussed in the table for women classification.

21. Based on the AUC table and ROC curves, it can be concluded that gradient boosting is the most effective model for predicting match outcomes.

22. The team that wins the toss is approximately three times less likely to win the match compared to the team that loses the toss.

23. The odds ratio of 2.057143 suggests that the team batting first has approximately 2.06 times higher odds of winning the match compared to the team batting second.

24. An odds ratio of 2.625 suggests that the odds of winning a match when playing at home are 2.625 times higher than the odds of losing a match when playing away.

25. Based on a p-value of 0.06382, there is insufficient evidence to conclude significant differences between the means of the groups of continents on the dependent variables at the 0.05 significance level.

26. Based on the reported p-values, there is no significant difference in the means of any of the dependent variables across the levels of the independent variable (CONTINENTS).

27. The p-value (¿ 0.05) indicates no statistically significant differences in dependent variables across CONTINENTS.

28. The groups show statistically significant differences in means for runs and strike rate (p ¡ 0.05), but not for wickets, average, bowling average, bowling strike rate, bowling dot percentage, and catches (p ¿ 0.05).

29. The p-value (¿ 0.05) suggests there is insufficient evidence to conclude a significant difference in means of dependent variables among different levels of the independent variable ”COUNTRY.”

30. In summary, there is no significant difference in the means of the mentioned cricket performance metrics across different countries, as indicated by p-values greater than 0.05, except for ’WICKETS,’ which warrants further investigation due to its borderline p-value.

31. In conclusion, the p-value of 0.1079 suggests that the ’COUNTRY’ variable does not have a significant effect on the dependent variables, as it exceeds the common significance threshold of 0.05.

32. There is generally no significant difference in the means of most response variables across different countries, except for ”CATCHES,” which does exhibit a significant difference at a 0.05 significance level.

Chapter 5

Limitations and Future work

5.1 Limitations

There were certain limitations in the study. Some of them are:

1. **Data Availability**: The availability and quality of cricket statistics can be a significant limitation. Historical data for some countries and players may be incomplete or less detailed, which can affect the accuracy of the analysis.

2. **Data Bias**: Data may be biased towards more prominent cricket-playing nations, leading to the underrepresentation of players from smaller or less-known cricketing nations. This could skew the results and not provide a complete picture of ODI performance.

3. **Sample Size**: The time frame of 2019-2023 may not provide a large enough sample size, especially for emerging players or those who have sporadically played during this period. This can affect the statistical significance of the findings.

4. **Comparing Players Across Eras**: Comparing players from different countries and different periods can be challenging due to changes in playing conditions, rules, and equipment. Adjusting for these factors can be complex.

5. **External Factors**: Cricket performance can be influenced by various external factors, such as team dynamics, pitch conditions, weather, and opposition strength. These factors may not always be accounted for in the analysis.

5.2 Future Work

Keeping the above limitations in mind, the following works can be carried out in the future:

1. **Longitudinal Analysis**: Extend the study to cover a more extended time period to provide a more comprehensive view of player performance trends over time.

2. **Predictive Modeling**: Develop predictive models to forecast player performance based on historical data. This could be valuable for team selection and fantasy cricket applications.

3. **Data Enhancement**: Continuously work on improving data quality and accessibility by collaborating with cricket governing bodies and data providers.

4. **Fan Engagement**: Investigate how player performance data can be used to enhance fan engagement, such as creating fantasy cricket games or providing insights for commentators.

5. **Comparative Analysis**: Comparing the ODI performance of players across different formats (T20, Test, etc.) can provide a broader perspective on their skills and adaptability.

Appendix

This includes the R codes used for this project.

R Codes

```
\normalsize
#1.Principal component analysis
#For men
#Men all-rounders
library(tidyverse)
library(dplyr)
library(ggplot2)
library(FactoMineR)
library(factoextra)
men_data=read.csv("C:\\Users\\divak\\Downloads\\MenCluster.csv")
men_data
#Perform PCA on men’s data
men_pca_data =  men_data[,c(’INNINGS’, ’RUNS’, ’BALLSFACED’,
’OUTS’, ’AVG’, ’STRIKERATE’, ’FIFTIES’, ’HUNDREDS’, ’FOURS’,
’SIXES’, ’DOTPERCENTAGE’, ’WICKETS’, ’ECONOMY’, ’BOWLINGAVG’,
’BOWLINGSTRIKERATE’, ’FIFERS’, ’BOWLINGDOTPERCENTAGE’,
’CATCHES’, ’STUMPING’)]
#Standardize the data
men_pca_data_std <- scale(men_pca_data)
#Perform PCA
men_pca_result=prcomp(men_pca_data_std)
men_pca_result
player_rankings=data.frame(player_name=men_data$NAME,
PCA_score=men_pca_result$x[,1])
player_rankings
player_rankings=player_rankings[order
(-player_rankings$PCA_score),]
```

```
player_rankings
print(player_rankings)
#Men batting
menbatting_data=read.csv("C:\\Users\\divak\\Downloads\\Men_Batting.csv")
menbatting_data
#Perform PCA on men’s data
menbatting_pca_data <- menbatting_data[, c(’INNINGS’, ’RUNS’,
’BALLSFACED’, ’OUTS’, ’AVG’,’STRIKERATE’, ’FIFTIES’, ’HUNDREDS’,
’FOURS’, ’SIXES’, ’DOTPERCENTAGE’, ’CATCHES’)]
#Standardize the data
menbatting_pca_data_std <- scale(menbatting_pca_data)
#Perform PCA
menbatting_pca_result=prcomp(menbatting_pca_data_std)
men_pca_result player_rankings3=data.frame
(player_name=menbatting_data
$NAME,PCA_score=menbatting_pca_result$x[,1])
player_rankings3
player_rankings3=player_rankings3[order
(-player_rankings3$PCA_score),]
player_rankings3
print(player_rankings3)

#Men bowling
menbowling_data=read.csv("C:\\Users\\divak\\Downloads\\menbowling.csv")
menbowling_data
#Perform PCA on men’s data
menbowling_pca_data <-menbowling_data[,c(’INNINGS’, ’RUNS’,
’BALLSFACED’, ’OUTS’, ’AVG’, ’STRIKERATE’, ’FIFTIES’, ’HUNDREDS’,
’FOURS’, ’SIXES’,’DOTPERCENTAGE’,
’CATCHES’)]
#Standardize the data
menbowling_pca_data_std <- scale(menbowling_pca_data)
#Perform PCA
menbowling_pca_result = prcomp(menbowling_pca_data_std)
men_pca_result
player_rankings4=data.frame(player_name=menbowling_data
$NAME,PCA_score=menbowling_pca_result$x[,1])
player_rankings4
player_rankings4=player_rankings4[order
(-player_rankings4$PCA_score),]
player_rankings4
```

```
print(player_rankings4)

# PCA for women
#Women all-rounders
library(tidyverse)
library(dplyr)
library(ggplot2)
library(FactoMineR)
library(factoextra)
#Women data women_data=read.csv("C:\\Users\\divak\\
Downloads\\WomenCluster.csv")
#Perform PCA on women's data
women_pca_data <- women_data[, c('INNINGS', 'RUNS', 'BALLSFACED',
'OUTS', 'AVG', 'STRIKERATE', 'FIFTIES', 'HUNDREDS', 'FOURS', 'SIXES',
'DOTPERCENTAGE', 'WICKETS', 'ECONOMY', 'BOWLINGAVG',
'BOWLINGSTRIKERATE', 'FIFERS', 'CATCHES',
'STUMPING')]
#Standardize the data
women_pca_data_std <- scale(women_pca_data)

#Perform PCA
women_pca_result=prcomp(women_pca_data)
women_pca_result

player_rankings1=data.frame(player_name=women_data$
NAME,PCA_score=women_pca_result$x[,1])
player_rankings1
player_rankings1=player_rankings1[order
(-player_rankings1$PCA_score),]
player_rankings1
print(player_rankings1)

#Women batting
womenbatting_data=read.csv("C:\\Users\\divak\\Downloads\\womenbatting.csv")
womenbatting_data
#Perform PCA on women's data
womenbatting_pca_data  <- womenbatting_data[,c('INNINGS',
'RUNS', 'BALLSFACED', 'OUTS', 'STRIKERATE', 'AVG', 'FIFTIES', 'HUNDREDS',
'FOURS', 'SIXES', 'CATCHES')]
#Standardize the data
womenbatting_pca_data_std <- scale(womenbatting_pca_data)
```

```
#Perform PCA
womenbatting_pca_result=prcomp(womenbatting_pca_data)
womenbatting_pca_result
player_rankings2=data.frame(player_name=womenbatting_data
$NAME,PCA_score=womenbatting_pca_result$x[,1])
player_rankings2
player_rankings2=player_rankings2[order
(-player_rankings2$PCA_score),]
player_rankings2
print(player_rankings2)

#Women bowling
womenbowling_data=read.csv("C:\\Users\\divak\\Downloads\\womenbowling.csv")
womenbowling_data
#Perform PCA on
women’s data womenbowling_pca_data=womenbowling_data[,
c(’OVERS’, ’WICKETS’, ’ECONOMY’, ’BOWLINGAVG’,
’BOWLINGSTRIKERATE’, ’BOWLINGDOTPERCENT’, ’CATCHES’)]
#Standardize the data
womenbowling_pca_data_std <- scale(womenbowling_pca_data)
#Perform PCA
womenbowling_pca_result = prcomp(womenbowling_pca_data)
womenbowling_pca_result
player_rankings5=data.frame(player_name=womenbowling_data
$NAME,PCA_score=womenbowling_pca_result$x[,1])
player_rankings5
player_rankings5=player_rankings5[order
(-player_rankings5$PCA_score),]
player_rankings5
print(player_rankings5)

#2.Cluster analysis
#Heirarchichal clustering for men
# Load required libraries
library(cluster)
library(factoextra)
library(dplyr)
library(dendextend)
#Men all-rounders
# Load the dataset (replace ’your_data.csv’ with the actual file path)
data <- read.csv("C:\\Users\\divak\\Downloads\\MenallroundersAsia.csv")
```

```
# Select relevant features for clustering
selected_data <- data[, c("NAME", "RUNS", "STRIKERATE",
"AVG", "INNINGS", "FOURS", "SIXES", "DOTPERCENTAGE",
"HUNDREDS", "FIFTIES", "WICKETS", "BOWLINGSTRIKERATE",
"BOWLINGAVG", "BOWLINGINNINGS", "ECONOMY",
"BOWLINGDOTPERCENTAGE")]
# Standardize the data
scaled_data <- scale(selected_data[, -1]) # Exclude the PlayerName column

# Perform hierarchical clustering using Ward’s method
dist_matrix <- dist(scaled_data, method = "euclidean")
hc <- hclust(dist_matrix, method = "ward.D2")
# Plot the dendrogram with rectangle clusters
plot(hc, hang = -1, labels = data$NAME, main =
"Hierarchical Clustering Dendrogram")
%rect.hclust(hc, k = 5, border = "red") # Adjust k as needed

#For Men Batters
# Load the dataset (replace ’your_data.csv’ with the actual file path)
data <- read.csv("C:\\Users\\divak\\Downloads\\MenbatsmansAsia.csv")
selected_data <- data[, c("NAME", "INNINGS", "RUNS",
"BALLSFACED", "OUTS", "AVG", "STRIKERATE", "HUNDREDS",
"FIFTIES", "DOTPERCENTAGE", "FOURS", "SIXES")]

# Standardize the data
scaled_data <- scale(selected_data[, -1])
# Perform hierarchical clustering using Ward’s method
dist_matrix <- dist(scaled_data, method = "euclidean")
hc <- hclust(dist_matrix, method = "ward.D2")
# Plot the dendrogram with rectangle clusters
plot(hc, hang = -1, labels = data$NAME, main =
"Hierarchical Clustering Dendrogram")
rect.hclust(hc, k = 5, border = "red")

#For Men Bowlers
# Load the dataset (replace ’your_data.csv’ with the
actual file path)
data <- read.csv("C:\\Users\\divak\\Downloads\\MenbowlerAsia.csv")
# Select relevant features for clustering
selected_data <- data[, c("NAME", "WICKETS",
"BOWLINGSTRIKERATE","BOWLINGAVG","BOWLINGINNINGS",
"ECONOMY", "BOWLINGDOTPERCENTAGE")]
# Standardize the data
scaled_data <- scale(selected_data[, -1
# Perform hierarchical clustering using Ward’s method
dist_matrix <- dist(scaled_data, method = "euclidean")
hc <- hclust(dist_matrix, method = "ward.D2")
# Plot the dendrogram with rectangle clusters
plot(hc, hang = -1, labels = data$NAME, main =
"Hierarchical Clustering Dendrogram")
rect.hclust(hc, k = 5, border = "red") # Adjust k as needed

#Heirarchichal Clustering for women
# Load required libraries
library(cluster)
library(factoextra)
library(dplyr)
library(dendextend)

#For Women all-rounders
data<-read.csv("C:\\Users\\divak\\Downloads\\
WomenAsiaAllRounders.csv")
# Select relevant features for clustering
selected_data <- data[, c("NAME", "RUNS", "STRIKERATE", "AVG",
"INNINGS", "FOURS", "SIXES", "DOTPERCENTAGE", "HUNDREDS",
"FIFTIES", "WICKETS", "BOWLINGSTRIKERATE", "BOWLINGAVG",
"BOWLINGINNINGS", "ECONOMY", "BOWLINGDOTPERCENTAGE")]
# Standardize the data
scaled_data <- scale(selected_data[, -1])
# Perform hierarchical clustering using Ward’s method
dist_matrix <- dist(scaled_data, method = "euclidean")
hc <- hclust(dist_matrix, method = "ward.D2")
# Plot the dendrogram with rectangle clusters
plot(hc, hang = -1, labels = data$NAME, main =
"Hierarchical Clustering Dendrogram",asp=0.5)
rect.hclust(hc, k = 5, border = "red") # Adjust k as needed

#For women batters
# Load the dataset (replace ’your_data.csv’ with the actual file path)
data<-read.csv("C:\\Users\\divak\\Downloads
\\WomenAsiaBatsmans.csv")
selected_data <- data[, c("NAME", "RUNS","STRIKERATE",
"AVG","INNINGS","FOURS","SIXES","DOTPERCENTAGE",
"HUNDREDS", "FIFTIES")]
# Standardize the data
scaled_data <- scale(selected_data[, -1])
# Perform hierarchical clustering using Ward's method
dist_matrix <- dist(scaled_data, method = "euclidean")
hc <- hclust(dist_matrix, method = "ward.D2")
# Plot the dendrogram with rectangle clusters
plot(hc, hang = -1, labels = data$NAME, main =
"Hierarchical Clustering Dendrogram",asp=0.5)
rect.hclust(hc, k = 5, border = "red")

#For Women bowlers
data <- read.csv("C:\\Users\\divak\\Downloads
\\WomenAsiaBowlers.csv")
# Select relevant features for clustering
selected_data <- data[, c("NAME","WICKETS",
"BOWLINGSTRIKERATE","BOWLINGAVG",
"BOWLINGINNINGS", "ECONOMY",
"BOWLINGDOTPERCENTAGE")]
# Standardize the data
scaled_data <- scale(selected_data[, -1])
# Perform hierarchical clustering using Ward's method
dist_matrix <- dist(scaled_data, method = "euclidean")
hc <- hclust(dist_matrix, method = "ward.D2")
# Plot the dendrogram with rectangle clusters
plot(hc, hang = -1, labels = data$NAME, main =
"Hierarchical Clustering Dendrogram",asp=0.5)
rect.hclust(hc, k = 5, border = "red") # Adjust k as needed

#K Means Clustering For Men
# Load required libraries
library(dplyr)
library(cluster)
library(factoextra)
library(ggplot2)
#For Men All rounders
# Read the CSV file
cricket_data <- read.csv("C:\\Users\\divak\\Downloads\\
MenallroundersAsia.csv", header = TRUE)
# Select the columns for clustering
```

```
features <- cricket_data[, c("RUNS","STRIKERATE", \AVG", "INNINGS",
"FOURS", "SIXES", "DOTPERCENTAGE", "HUNDREDS", "FIFTIES",
"WICKETS", "ECONOMY", "BOWLINGAVG", "BOWLINGINNINGS",
"BOWLINGSTRIKERATE",  "BOWLINGDOTPERCENTAGE")]
# Perform scaling on the features
scaled_features <- scale(features)
# Choose the optimal number of clusters
optimal_k <- 4
# Perform K-Means clustering
kmeans_result <- kmeans(scaled_features, centers = optimal_k)
# Add cluster assignments to the original data
cricket_data$Cluster <- as.factor(kmeans_result$cluster)

# Scatter plot
plot_data <- as.data.frame(cbind(Player = cricket_data$NAME,
Cluster = cricket_data$Cluster))
plot_data$Cluster <- as.factor(plot_data$Cluster)
plot_data$Player <- as.character(plot_data$Player)

# Scatter plot
plot_data %>%
ggplot(aes(x = Player, y = Cluster, color = Cluster)) +
geom_point(size = 3) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = "Player", y = "Cluster") +
ggtitle("K-Means Clustering of Cricket Players") +
scale_color_discrete(name = "Cluster")+
theme(aspect.ratio = 1/2)

# Silhouette plot
silhouette <- silhouette(kmeans_result$cluster, dist(scaled_features))
fviz_silhouette(silhouette)+
theme(aspect.ratio = 3/4)

#For Women Batters
# Read the CSV file
cricket_data <- read.csv("C:\\Users\\divak\\Downloads\\
MenbatsmansAsia.csv", header = TRUE)

# Select the columns for clustering
features <- cricket_data[, c("RUNS", "STRIKERATE",
"AVG", "INNINGS", "FOURS", "SIXES", "DOTPERCENTAGE",
"HUNDREDS", "FIFTIES")]
# Perform scaling on the features
scaled_features <- scale(features)
# Choose the optimal number of clusters
optimal_k <- 4
# Perform K-Means clustering
kmeans_result <- kmeans(scaled_features, centers = optimal_k)
# Add cluster assignments to the original data
cricket_data$Cluster <- as.factor(kmeans_result$cluster)

# Scatter plot
plot_data <- as.data.frame(cbind(Player = cricket_data$NAME,
Cluster = cricket_data$Cluster))
plot_data$Cluster <- as.factor(plot_data$Cluster)
plot_data$Player <- as.character(plot_data$Player)

# Scatter plot
plot_data %>%
ggplot(aes(x = Player, y = Cluster, color = Cluster)) +
geom_point(size = 3) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = "Player", y = "Cluster") +
ggtitle("K-Means Clustering of Cricket Players") +
scale_color_discrete(name = "Cluster") +
theme(aspect.ratio = 1/2)

# Silhouette plot
silhouette <- silhouette(kmeans_result$cluster, dist(scaled_features))
fviz_silhouette(silhouette) +
theme(aspect.ratio = 1/2)

#For Women Bowlers
# Read the CSV file
cricket_data <- read.csv("C:\\Users\\divak\\Downloads\\
MenbowlerAsia.csv", header = TRUE)
# Select the columns for clustering
features <- cricket_data[, c("WICKETS", "ECONOMY",
"BOWLINGAVG", "BOWLINGINNINGS",
"BOWLINGSTRIKERATE", "BOWLINGDOTPERCENTAGE")]
# Perform scaling on the features
```

```
scaled_features <- scale(features)
# Choose the optimal number of
optimal_k <- 4
# Perform K-Means clustering
kmeans_result <- kmeans(scaled_features, centers = optimal_k)
# Add cluster assignments to the original data
cricket_data$Cluster <- as.factor(kmeans_result$cluster)

# Scatter plot
plot_data <- as.data.frame(cbind(Player = cricket_data$NAME,
Cluster = cricket_data$Cluster))
plot_data$Cluster <- as.factor(plot_data$Cluster)
plot_data$Player <- as.character(plot_data$Player)

# Scatter plot
plot_data %>%
ggplot(aes(x = Player, y = Cluster, color = Cluster)) +
geom_point(size = 3) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = "Player", y = "Cluster") +
ggtitle("K-Means Clustering of Cricket Players") +
scale_color_discrete(name = "Cluster") +
theme(aspect.ratio = 1/2)

# Silhouette plot
silhouette <- silhouette(kmeans_result$cluster,
dist(scaled_features))
fviz_silhouette(silhouette) +
theme(aspect.ratio = 3/4)

#For women all-rounders
# Read the CSV file
cricket_data <- read.csv("C:\\Users\\lohithlikith\\
Documents\\Lohith B N\\Project work\\
WomenAsiaAllrounders.csv", header = TRUE)

# Select the columns for clustering
features <- cricket_data[, c("RUNS", "STRIKERATE", "AVG",
"INNINGS", "FOURS","SIXES", "DOTPERCENTAGE", "HUNDREDS",
"FIFTIES", "WICKETS", "ECONOMY", "BOWLINGAVG",
"BOWLINGINNINGS", "BOWLINGSTRIKERATE",
"BOWLINGDOTPERCENTAGE")]

# Perform scaling on the features
scaled_features <- scale(features)
# Choose the optimal number of clusters
optimal_k <- 4
# Perform K-Means clustering
kmeans_result <- kmeans(scaled_features, centers = optimal_k)
# Add cluster assignments to the original data
cricket_data$Cluster <- as.factor(kmeans_result$cluster)

# Scatter plot
plot_data <- as.data.frame(cbind(Player = cricket_data$NAME,
Cluster = cricket_data$Cluster))
plot_data$Cluster <- as.factor(plot_data$Cluster)
plot_data$Player <- as.character(plot_data$Player)

# Scatter plot
plot_data %>% ggplot(aes(x = Player, y = Cluster, color = Cluster)) +
geom_point(size = 3) +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(x = "Player", y = "Cluster") +
ggtitle("K-Means Clustering of Cricket Players") +
scale_color_discrete(name = "Cluster") +
theme(aspect.ratio = 1/2)

# Silhouette plot
silhouette <- silhouette(kmeans_result$cluster, dist(scaled_features))
fviz_silhouette(silhouette)+
theme(aspect.ratio = 1/2)

#Classification of players using machine learning tools for Men

#1)Random Forest
library(randomForest)
# Load the dataset
cricket_data <- read.csv("C:\\Users\\lohithlikith\\
Documents\\Lohith B N\\Project Work\\MENCRICKETDATA.csv")
# Explore the dataset (optional)
head(cricket_data)
# Splitting the dataset into training and testing sets
```

```
set.seed(123)  # For reproducibility
train_indices <- sample(nrow(cricket_data), 0.7 * nrow(cricket_data))
train_data <- cricket_data[train_indices, ]
test_data <- cricket_data[-train_indices, ]
test_data

# Define the Random Forest model
rf_model <- randomForest(
PLAYERTYPE ~ INNINGS + RUNS + BALLSFACED + STRIKERATE + AVG +
FIFTIES + HIGHEST.SCORE + WICKETS + ECONOMY + BOWLINGAVG
+ BOWLINGSTRIKERATE + BOWLINGDOTPERCENTAGE + CATCHES ,
data = train_data,
ntree = 500,     # Number of trees in the forest
mtry = 3,
importance = TRUE)     # Calculate variable importance
# View the summary of the Random Forest model
print(rf_model)
# Make predictions on the test set
predictions <- predict(rf_model, newdata = test_data)
predictions

# Evaluate the model's performance
confusion_matrix <- table(predictions, test_data$PLAYERTYPE)
print(confusion_matrix)

# Calculate accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy:", accuracy))

#2)Multinomial logistic regression
# Load required libraries
library(dplyr)
library(nnet)
library(ROCR)
library(caret)
# Define predictor variables (features) and target variable (label)
predictors <- cricket_data[, c("RUNS","AVG","STRIKERATE",
"WICKETS","BOWLINGAVG","BOWLINGSTRIKERATE",
"BOWLINGDOTPERCENTAGE","CATCHES")]
target <- cricket_data$PLAYERTYPE
# Create training and testing datasets
```

```
set.seed(123)  # For reproducibility
train_indices <- createDataPartition(target, p = 0.7, list = FALSE)
train_data <- predictors[train_indices, ]
test_data <- predictors[-train_indices, ]
train_labels <- target[train_indices]
test_labels <- target[-train_indices]

# Train the multinomial logistic regression model
model <- multinom(label ˜ ., data = data.frame(label =
train_labels, train_data))

# Predict labels for the test set
predicted_labels1 <- predict(model, newdata = test_data)

# Calculate accuracy
accuracy1 <- mean(predicted_labels1 == test_labels)
accuracy1
cat("Accuracy1:", accuracy1, "\n")

# Create confusion matrix
conf_matrix1 <- table(predicted_labels1, test_labels)
cat("Confusion Matrix1:\n", conf_matrix1, "\n")

#3)Neural Networks
# Load required libraries
library(neuralnet)
library(dplyr)

# Preprocessing: normalize the features
data <- cricket_data

# Define the target categories as numeric labels
# 1: Batsman, 2: Bowler, 3: All-rounder
data$player_type <- as.numeric(factor(data$PLAYERTYPE))

# Split the data into training and testing sets
set.seed(123)
split_index <- sample(1:nrow(data), nrow(data) * 0.8)
train_data <- data[split_index, ]
test_data <- data[-split_index, ]
```

```
# Define the neural network model
nn_model <- neuralnet(
PLAYERTYPE ~ RUNS + AVG + STRIKERATE + WICKETS +
ECONOMY + BOWLINGAVG + BOWLINGSTRIKERATE +
BOWLINGDOTPERCENTAGE + CATCHES,
data = train_data,
hidden = c(5, 3),  # You can adjust the hidden layer sizes
linear.output = FALSE
)

# Make predictions on the test data
predictions <- predict(nn_model, test_data)
predicted_labels2 <- apply(predictions, 1, which.max)

# Evaluate the accuracy
true_labels <- test_data$player_type
accuracy2 <- sum(predicted_labels2 == true_labels)
/ length(true_labels)
cat("Accuracy2:", accuracy2, "\n")

# Load the caret library
library(caret)

# Generate the confusion matrix
confusion_matrix2 <- confusionMatrix(as.factor(predicted_labels2),
as.factor(true_labels))

# Print the confusion matrix
print(confusion_matrix2$table)

#4) SVM
# Load the necessary library
library(e1071)
# Split the dataset into features and labels
features <- cricket_data[, c("RUNS",  "AVG", "STRIKERATE", "WICKETS",
"BOWLINGAVG", "CATCHES")]
labels <- cricket_data$PLAYERTYPE
# Convert labels to factor
labels <- as.factor(labels)
# Split the dataset into training and testing sets
set.seed(123)  # For reproducibility
```

```
train_indices <- sample(1:nrow(cricket_data), 0.7 * nrow(cricket_data))
train_features <- features[train_indices, ]
train_labels <- labels[train_indices]
test_features <- features[-train_indices, ]
test_labels <- labels[-train_indices]

# Train the SVM model
svm_model <- svm(train_labels ~ ., data = train_features, kernel = "linear")
# Predict on the test set
predictions3 <- predict(svm_model, test_features)
# Evaluate the accuracy
accuracy3 <- sum(predictions3 == test_labels) / length(test_labels)
cat("Accuracy3:", accuracy3)
# Create the confusion matrix
confusion_matrix3 <- table(predictions3, test_labels)

# Display the confusion matrix
confusion_matrix3

#5) Decision Trees
library(rpart)
# Splitting the data into features (X) and target (y)
X <- cricket_data[, c("RUNS",  "AVG",  "STRIKERATE",  "WICKETS",
"BOWLINGAVG",  "CATCHES")]
y <- cricket_data$PLAYERTYPE

# Training the Decision Tree model
tree_model <- rpart(y ~ ., data = X, method = "class")
# Predicting player types using the trained model
predictions4 <- predict(tree_model, X, type = "class")
# Evaluating the model (You might use other evaluation metrics)
accuracy4 <- sum(predictions4 == y) / length(y)
cat("Accuracy4:", accuracy4, "\n")
# Generating the confusion matrix
confusion_matrix4 <- table(predictions4, y)
confusion_matrix4

#6)LDA
# Load required libraries
library(MASS)  # For the lda() function
cricket_data <- read.csv("C:\\Users\\lohithlikith\\
Documents\\Lohith B N\\Project work\\men12.csv")
labels <- cricket_data[, 1]
features <- cricket_data[, -1]

# Perform Linear Discriminant Analysis (LDA)
lda_model <- lda(PLAYERTYPE ~ ., data = cricket_data)
# Print summary of the LDA model
print(lda_model)
# Predict class labels using the LDA model
predicted_labels5<- predict(lda_model, newdata = features)$class
# Compare predicted labels with the actual labels
confusion_matrix5 <- table(Actual = labels, Predicted = predicted_labels5)
print(confusion_matrix5)
# Calculate accuracy
accuracy5 <- sum(diag(confusion_matrix5)) / sum(confusion_matrix5)
print(paste("Accuracy5:", accuracy5))
#Classification of players using machine learning tools for Women

#1)Random Forest
library(randomForest)
# Load the dataset
cricket_data <- read.csv("C:\\Users\\lohithlikith\\
Documents\\Lohith B N\\Project Work\\Womendata.csv")
# Explore the dataset (optional)
head(cricket_data)
# Splitting the dataset into training and testing sets
set.seed(123)  # For reproducibility
train_indices <- sample(nrow(cricket_data), 0.7 * nrow(cricket_data))
train_data <- cricket_data[train_indices, ]
test_data <- cricket_data[-train_indices, ]
test_data
# Define the Random Forest model
rf_model <- randomForest(
PLAYERTYPE ~  RUNS + AVG + STRIKERATE + WICKETS + BOWLINGAVG
+  BOWLINGSTRIKERATE + CATCHES,
data = train_data,
ntree = 500,         # Number of trees in the forest
mtry = 3,
importance = TRUE)
print(rf_model)
# Make predictions on the test set
```

```
predictions <- predict(rf_model, newdata = test_data)
predictions
# Evaluate the model’s performance
confusion_matrix <- table(predictions, test_data$PLAYERTYPE)
print(confusion_matrix)
# Calculate accuracy
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Accuracy:", accuracy))

#2)Multinomial logistic regression
# Load required libraries
library(dplyr)
library(nnet)
install.packages("ROCR")
library(ROCR)
library(caret)
# Define predictor variables (features) and target variable (label)
predictors <- cricket_data[, c("RUNS", "AVG", "STRIKERATE",
"WICKETS", "BOWLINGAVG", "CATCHES")]
target <- cricket_data$PLAYERTYPE
# Create training and testing datasets
set.seed(123)  # For reproducibility
train_indices <- createDataPartition(target, p = 0.7, list = FALSE)
train_data <- predictors[train_indices, ]
test_data <- predictors[-train_indices, ]
train_labels <- target[train_indices]
test_labels <- target[-train_indices]
# Train the multinomial logistic regression model
model <- multinom(label ˜ ., data = data.frame(label =
train_labels, train_data))
# Predict labels for the test set
predicted_labels1 <- predict(model, newdata = test_data)
# Calculate accuracy
accuracy1 <- mean(predicted_labels1 == test_labels)
accuracy1
cat("Accuracy1:", accuracy1, "\n")
# Create confusion matrix
conf_matrix1 <- table(predicted_labels, test_labels)
cat("Confusion Matrix:\n", conf_matrix1, "\n")

#3)Neural Networks
```

```
# Load required libraries
library(neuralnet)
library(dplyr)
# Preprocessing: normalize the features
data <- cricket_data
# Define the target categories as numeric labels
# 1: Batsman, 2: Bowler, 3: All-rounder
data$player_type <- as.numeric(factor(data$PLAYERTYPE))
# Split the data into training and testing sets
set.seed(123)
split_index <- sample(1:nrow(data), nrow(data) * 0.8)
train_data <- data[split_index, ]
test_data <- data[-split_index, ]
# Define the neural network model
nn_model <- neuralnet(
PLAYERTYPE ~ RUNS + AVG + STRIKERATE + WICKETS
+ BOWLINGAVG + CATCHES,
data = train_data,
hidden = c(5, 3),  # You can adjust the hidden layer sizes
linear.output = FALSE)
# Make predictions on the test data
predictions <- predict(nn_model, test_data)
predicted_labels2 <- apply(predictions, 1, which.max)
# Evaluate the accuracy
true_labels <- test_data$player_type
accuracy2 <- sum(predicted_labels2 ==
true_labels) / length(true_labels)
cat("Accuracy2:", accuracy2, "\n")

#4) SVM
# Load the necessary library
library(e1071)
# Split the dataset into features and labels
features <- cricket_data[, c("RUNS",  "AVG",  "STRIKERATE",
"WICKETS",  "BOWLINGAVG", "CATCHES")]
labels <- cricket_data$PLAYERTYPE
# Convert labels to factor
labels <- as.factor(labels)
# Split the dataset into training and testing sets
set.seed(123)  # For reproducibility
train_indices <- sample(1:nrow(cricket_data), 0.7 * nrow(cricket_data))
```

```
train_features <- features[train_indices, ]
train_labels <- labels[train_indices]
test_features <- features[-train_indices, ]
test_labels <- labels[-train_indices]
# Train the SVM model
svm_model <- svm(train_labels ˜ ., data = train_features, kernel = "linear")
# Predict on the test set
predictions3 <- predict(svm_model, test_features)

# Evaluate the accuracy
accuracy3 <- sum(predictions3 == test_labels) / length(test_labels)
cat("Accuracy3:", accuracy3)

#5) Decision Trees
library(rpart)
# Splitting the data into features (X) and target (y)
X <- cricket_data[, c("RUNS", "AVG", "STRIKERATE", "WICKETS",
"BOWLINGAVG", "CATCHES")]
y <- cricket_data$PLAYERTYPE
# Training the Decision Tree model
tree_model <- rpart(y ˜ ., data = X, method = "class")
# Predicting player types using the trained model
predictions4 <- predict(tree_model, X, type = "class")
# Evaluating the model (You might use other evaluation metrics)
accuracy4 <- sum(predictions4 == y) / length(y)
cat("Accuracy4:", accuracy4, "\n")

#6)LDA
# Load required library
library(MASS)
data <- read.csv("C:\\Users\\lohithlikith\\Documents\\
Lohith B N\\Project Work\\Womendata.csv")
# Define the features and target variable
features <- data[, c("RUNS", "AVG", "STRIKERATE", "WICKETS",
"BOWLINGAVG", "CATCHES")]
target <- data$PLAYERTYPE
# Perform Linear Discriminant Analysis (LDA)
lda_model <- lda(features, target)
# Make predictions using the LDA model
predictions5 <- predict(lda_model, features)
confusion_matrix5 <- table(predictions5$class, target)
```

```
print(confusion_matrix5)
accuracy5 <- sum(diag(confusion_matrix5)) / sum(confusion_matrix5)
print(paste("Accuracy5:", accuracy5))
# MANOVA of Continents for Men
# Load the required libraries
library(tidyverse)
library(car)
# Read in the data
data <- read.csv("C:\\Users\\lohithlikith\\Documents\\Lohith B N
\\Project work\\Continents.csv")
# Fit a MANOVA model
manova_fit <- manova(cbind(RUNS, WICKETS, AVG,
STRIKERATE, BOWLINGAVG, BOWLINGSTRIKERATE,
BOWLINGDOTPERCENTAGE,
CATCHES) ~ CONTINENTS, data = data)
# Print the summary of the MANOVA model
summary(manova_fit)
summary
# Perform ANOVA on each response variable
anova_runs <- aov(RUNS ~ CONTINENTS, data = data)
anova_wickets <- aov(WICKETS ~ CONTINENTS, data = data)
anova_average <- aov(AVG ~ CONTINENTS, data = data)
anova_strike_rate <- aov(STRIKERATE ~
CONTINENTS, data = data)
anova_bowling_average <- aov(BOWLINGAVG ~
CONTINENTS, data = data)
anova_bowling_strike_rate <- aov(BOWLINGSTRIKERATE ~
CONTINENTS, data = data)
anova_bowling_dot_percentage <- aov(BOWLINGDOTPERCENTAGE
~ CONTINENTS, data = data)
anova_catches <- aov(CATCHES ~ CONTINENTS, data = data)
# Print the summary of each ANOVA model
summary(anova_runs)
summary(anova_wickets)
summary(anova_average)
summary(anova_strike_rate)
summary(anova_bowling_average)
summary(anova_bowling_strike_rate)
summary(anova_bowling_dot_percentage)
summary(anova_catches)
```

```
# MANOVA of Continents for Women
# Load the required libraries
library(tidyverse)
library(car)
# Read in the data
data <- read_csv("C:\\Users\\lohithlikith\\
Documents\\Lohith B N\\Project work\\WomenContinents.csv")
# Fit a MANOVA model
manova_fit <- manova(cbind(RUNS,  WICKETS,  AVG,
STRIKERATE, BOWLINGAVG, BOWLINGSTRIKERATE,
BOWLINGDOTPERCENTAGE, CATCHES) ~ CONTINENTS, data = data)
# Print the summary of the MANOVA model
summary(manova_fit)
# Perform ANOVA on each response variable
anova_runs <- aov(RUNS ~ CONTINENTS, data = data)
anova_wickets <- aov(WICKETS ~ CONTINENTS, data = data)
anova_average <- aov(AVG ~ CONTINENTS, data = data)
anova_strike_rate <- aov(STRIKERATE ~
CONTINENTS, data = data)
anova_bowling_average <- aov(BOWLINGAVG ~
CONTINENTS, data = data)
anova_bowling_strike_rate <- aov(BOWLINGSTRIKERATE ~
CONTINENTS, data = data)
anova_bowling_dot_percentage <- aov
(BOWLINGDOTPERCENTAGE ~ CONTINENTS, data = data)
anova_catches <- aov(CATCHES ~ CONTINENTS, data = data)

# Print the summary of each ANOVA model
summary(anova_runs)
summary(anova_wickets)
summary(anova_average)
summary(anova_strike_rate)
summary(anova_bowling_average)
summary(anova_bowling_strike_rate)
summary(anova_bowling_dot_percentage)
summary(anova_catches)

# MANOVA of Asia For Men
data <- read.csv("C:\\Users\\lohithlikith\\Documents\\
Lohith B N\\Project Work\\Asia.csv")
# Load the required libraries
```

```
library(tidyverse)
library(car)
# Fit the MANOVA model
fit <- manova(cbind(RUNS, AVG, STRIKERATE, WICKETS,
BOWLINGAVG, BOWLINGSTRIKERATE, BOWLINGDOTPERCENTAGE,
CATCHES) ~ COUNTRY, data = data)
# Print the summary of the MANOVA model
summary(fit)
# Perform ANOVA on each dependent variable separately
summary.aov(fit)

# MANOVA of Asia For Women
data <- read.csv("C:\\Users\\lohithlikith\\Documents\\
Lohith B N\\Project Work\\WomenAsia.csv")
# Load the required libraries
library(tidyverse)
library(car)
# Fit the MANOVA model
fit <- manova(cbind(RUNS,  AVG,  STRIKERATE,  WICKETS,  BOWLINGAVG,
1BOWLINGSTRIKERATE, BOWLINGDOTPERCENTAGE, CATCHES) ~
COUNTRY, data = data)
# Print the summary of the MANOVA model
summary(fit)


# Perform ANOVA on each dependent variable separately
summary.aov(fit)
# Prediction of an outcome of a ODI match

#1) KNN Prediction
# Load required libraries
library(class)
library(pROC)
# Read in the data
data = read.csv("C:\\Users\\lohithlikith\\Documents\\
Lohith B N\\Project work\\Mentoss.csv")
# Set the seed for reproducibility
set.seed(123)
# Split the data into training and test sets
train_index <- sample(1:nrow(data), 0.7*nrow(data))
train_data <- data[train_index,]
```

```
test_data <- data[-train_index,]
# Define the predictor variables and the outcome variable
predictors <- c("TOSS", "BATTINGFIRST", "HOME")
outcome <- "MATCHRESULT"
# Scale the predictor variables
train_data[predictors] <- scale(train_data[predictors])
test_data[predictors] <- scale(test_data[predictors])
# Use KNN to make predictions on the test set
knn_predictions <- knn(train_data[predictors],
test_data[predictors], train_data[[outcome]], k=3)
# Convert predictions to a numeric variable
knn_predictions_numeric <- as.numeric(knn_predictions)
# Create an ROC plot
roc_obj1 <- roc(test_data[[outcome]], knn_predictions_numeric)
plot(roc_obj1)

#2) Naive Bayes
# Load necessary libraries
library(e1071)
library(pROC)
# Split the data into training and test sets
set.seed(123)
train_index <- sample(1:nrow(data), 0.7*nrow(data))
train_data <- data[train_index,]
test_data <- data[-train_index,]
# Train a Naive Bayes model on the training data
model <- naiveBayes(MATCHRESULT ~ TOSS + BATTINGFIRST +
HOME, data = train_data)
# Make predictions on the test data
predictions1 <- predict(model, test_data)
# Convert predictions to a numeric variable
predictions_numeric1 <- as.numeric(predictions1)
# Generate a ROC plot to evaluate the performance of the model
roc_obj2 <- roc(test_data$MATCHRESULT,  predictions_numeric1)
plot(roc_obj2)

#3) SVM
# Load necessary libraries
library(e1071)   # For SVM
library(caret)   # For data preprocessing
library(e1071)
library(pROC)
# Split the data into training and test sets
set.seed(123)
train_index <- sample(1:nrow(data), 0.7 * nrow(data))
train_data <- data[train_index, ]
test_data <- data[-train_index, ]

# Train the SVM model
svm_model <- svm(MATCHRESULT ~ TOSS + BATTINGFIRST + HOME,
data = train_data, kernel = "radial")
# Make predictions on the test set
predictions2 <- predict(svm_model, test_data)
predictions2
# Convert predictions to a numeric variable
predictions_numeric2 <- as.numeric(predictions2)
# Generate a ROC plot using the numeric predictions
roc_obj3 <- roc(test_data$MATCHRESULT, predictions_numeric2)
plot(roc_obj3)

#4) Neural Networks
# Load necessary libraries
library(neuralnet)
library(ROCR)
# Split data into training and test sets
set.seed(123)
train_index <- sample(1:nrow(data), 0.7*nrow(data))
train_data <- data[train_index,]
test_data <- data[-train_index,]
# Normalize the data
maxs <- apply(train_data, 2, max)
mins <- apply(train_data, 2, min)
scaled_train_data <- as.data.frame(scale(train_data,
center = mins, scale = maxs - mins))
scaled_test_data <- as.data.frame(scale(test_data,
center = mins, scale = maxs - mins))
# Define the formula for the neural network
formula <- MATCHRESULT ~ TOSS + BATTINGFIRST + HOME
# Train the neural network
nn <- neuralnet(formula, data = scaled_train_data, hidden = c(5, 3),
linear.output = FALSE)
```

```
# Make predictions on the test set
predictions <- compute(nn, scaled_test_data[,
c("TOSS", "BATTINGFIRST", "HOME")])
predictions <- predictions$net.result
# Convert predictions to binary outcome
predictions[predictions > 0.5] <- 1
predictions[predictions <= 0.5] <- 0
# Plot ROC curve
pred <- prediction(predictions, scaled_test_data$MATCHRESULT)
perf <- performance(pred,"tpr","fpr")
plot(perf)
abline(0,1)

#5) Random Forest
# Load necessary libraries
library(randomForest)
library(pROC)
# Split the data into training and test sets
set.seed(123)
train_index <- sample(nrow(data), 0.7*nrow(data))
train_data <- data[train_index,]
test_data <- data[-train_index,]
# Train the Random Forest model
rf_model <- randomForest(MATCHRESULT ~ TOSS + BATTINGFIRST
+ HOME, data=train_data, ntree=500)
# Make predictions on the test set
predictions3 <- predict(rf_model, test_data)
# Generate the ROC plot
roc_obj4 <- roc(test_data$MATCHRESULT, predictions3)
plot(roc_obj4)

#6) Logistic Regression
# Load necessary libraries
library(tidyverse)
library(caret)
library(pROC)
# Split data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(data$MATCHRESULT, p = 0.8, list = FALSE)
train <- data[trainIndex, ]
test <- data[-trainIndex, ]
```

```
# Fit logistic regression model
fit <- glm(MATCHRESULT ~ TOSS + BATTINGFIRST + HOME,
data = train, family = binomial())
# Make predictions on test set
pred <- predict(fit, newdata = test, type = "response")
# Generate ROC curve
roc_obj5 <- roc(test$MATCHRESULT, pred)
plot(roc_obj5)
# Calculate AUC
auc(roc_obj5)

#7)Gradient Boosting
# Load data
data = read.csv("C:\\Users\\lohithlikith\\Documents\\
Lohith B N\\Project work\\Mentoss1.csv")
library(gbm)
library(pROC)
# Split data into training and test sets
set.seed(123)
train_index <- sample(nrow(data), 0.7 * nrow(data))
train_data <- data[train_index, ]
test_data <- data[-train_index, ]
# Fit gradient boosting model
gbm_fit <- gbm(MATCHRESULT ~ TOSS + BATTINGFIRST + HOME,
data = train_data, distribution = "bernoulli", n.trees = 1000,
interaction.depth = 4, shrinkage = 0.01)
# Make predictions on test set
test_pred6 <- predict(gbm_fit, newdata = test_data,
n.trees = 1000, type = "response")
test_pred6
# Generate ROC plot
roc_obj6 <- roc(test_data$MATCHRESULT, test_pred6)
plot(roc_obj6)
plot(roc_obj1)
plot(roc_obj2)
plot(roc_obj3)
plot(perf)
plot(roc_obj4)
plot(roc_obj5)
plot(roc_obj6)
plot(roc_obj1,col="red",main="ROC CURVES")
plot(roc_obj2,add=TRUE,col="blue")
plot(roc_obj3,add=TRUE,col="green")
plot(roc_obj4,add=TRUE,col="purple")
plot(roc_obj5,add=TRUE,col="orange")
plot(roc_obj6,add=TRUE,col="yellow")
legend("bottomright",  legend = c("roc_obj1",  "roc_obj2",
"roc_obj3",  "roc_obj4",  "roc_obj5", "roc_obj6"), fill = c("red",
"blue",  "green",  "purple", "orange", "yellow"))
auc(roc_obj1)
auc(roc_obj2)
auc(roc_obj3)
auc(roc_obj4)
auc(roc_obj5)
auc(roc_obj6)

#Odds Ratio
# Load the data
data <- read.csv("C:\\Users\\divak\\Downloads\\Mentoss.csv")
# Create a contingency table
table1 <- table(data$TOSS, data$MATCHRESULT)
# Calculate the odds ratio
odds_ratio1 <- (table1[1,1] * table1[2,2]) / (table1[1,2] * table1[2,1])
odds_ratio1
# Interpret the odds ratio
if (odds_ratio1 > 1)
{cat("The odds of winning the match are higher for the
team that wins the toss.")}
else if (odds_ratio1 < 1)
{cat("The odds of winning the match are lower for the
team that wins the toss.")}
else
{cat("There is no association between winning the
toss and winning the match.")}
# Create a contingency table
table2 <- table(data$MATCHRESULT, data$BATTINGFIRST)
# Calculate the Odds Ratio
odds_ratio2 <- (table2[1,1] * table2[2,2]) / (table2[1,2] * table2[2,1])
odds_ratio2
# Interpret the Odds Ratio
if (odds_ratio2 > 1)
{cat("The odds of winning the match are higher for the
team that bats first.")}
else if (odds_ratio2 < 1)
{cat("The odds of winning the match are higher for the
team that bats second.")}
else
{cat("There is no association between batting first or
second and winning the match.")}
# Create a contingency table
table3 <- table(data$MATCHRESULT, data$HOME)
# Calculate the odds ratio
odds_ratio3 <- (table3[1,1] * table3[2,2]) / (table3[1,2] * table3[2,1])
odds_ratio3
# Print the result
cat("Odds Ratio3:", odds_ratio3)
```

References

[1] Anderson, T. W. (2004), *An Introduction to Multivariate Statistical analysis, 3/e*, John Wiley, New York.

[2] Haykin, S (2009), *Neural Networks and Learning Machines*, Pearson, New Delhi.

[3] Ananda, B. Ferry, B. (2007). *Statistical Analysis of One Day International Cricket, Department of Mathematics*, Sam Houston State University, Huntsville.

[4] Hastie,T. Tibshirani,R. and Friedman,J.(2009), *The Elements of Statistical Learning, 2/e*, Springer, New York, USA.

[5] Tattar, P. N. (2013), *R Statistical Application Development by Example Beginner's Guide*, Packt Publishing Ltd.

Websites:

1. `www.espncricinfo.com`
2. `www.cricmetric.com`
3. `www.researchgate.net`
4. `https://www.datanovia.com/`
5. `www.analyticsvidhya.com`