Digital Cricket Scouting System
A machine learning model to scout best players

Project Report

Submitted by
Thirumal Janakiraman

Table of Contents

List of Figures

List of Tables

Abstract

Franchise cricket teams tend to scout players for an extended period, manually analysing each player's techniques, skills, and the statistics data to identify the best players. Players are being targeted based on the scouting team's expertise or manual analysis of either current or past statistics, which can be immensely vast, time-consuming, tedious, and might be deceiving. Due to past performances, players with a good reputation get picked over emerging players with consistent performance because of their popularity.

The project aims to create a web application based on Machine Learning to recognize the best players to target in the auction by analysing the past performance pattern and quantifying the players' performance. Furthermore, the project shall rate the performers of different players' expertise and help to shortlist the players for owners to target in the auction. Three algorithms, such as Naive Bayes, Decision Tree, and Random Forest, were used for machine learning. Random Forest yielded the best results, and it was utilized for the web application. As a result, the web application gives a better guide for the cricket experts and franchise teams for better accurate scouting.

Abbreviations & Definitions

Terms	**Definitions/Abbreviation**
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
API	Application Programming Interface
BCCI	Board of Control for Cricket in India
C.I	Consistency Index
C.R	Consistency Ratio
DOM	Document Object Model
ICC	International Cricket Council
IPL	Indian Premier League (Domestic cricket league of India)
ML	Machine Learning
NPM	Node Package Manager
ODI	One Day International
R.I	Random Index
T20	Twenty-Twenty
URL	Uniform Resource Locator

1. Introduction and Motivation

The amount of data present in today's world is absolutely tremendous and the importance of it is only realised to humans in recent years. Data, if combined with other technologies like AI or ML, can do wonders for humans. It can make decision taking ability quite simple, precise and accurate. The one industry which has recently realised the power of analytics and AI is sports and if implemented properly can significantly make a huge difference in sports.

Cricket is an international sport that is governed by the authority of the International Cricket Council (ICC). The sport is played between two teams consisting of eleven players each. The game is divided into two parts named bowling and batting. The players can be classified into different categories like Batsmen, Bowlers, All-rounders, Wicketkeepers. The responsibility of the batsman is to strike the ball with the bat across the boundary or run between the wickets to get runs and score maximum runs for the team and set the target or chase the target. The bowler's responsibility is to defend the runs scored by the batsman and take maximum wickets by giving less runs. The wicketkeepers are one among the fielders but they stand behind the wickets and they are considered as batsmen. Allrounders are the ones who can bat as well as bowl, they are responsible for scoring runs while batting and defending the runs while bowling. The matches are played in three different formats such as test match, one day international (ODI), and t20. Test matches are played for 5 days with unlimited overs and two innings for each team, the ODI's are played for 50 overs and T20s are played for 20 overs for each team. In recent times, T20 cricket has gained a lot of popularity worldwide among other formats. There are various T20 cricket leagues played worldwide by professional players. Among them IPL is regarded as one of the top leagues [1].

Indian Premier League, commonly known as IPL, is a popular and richest domestic cricket league played in India that is governed by the authority of the Board of Control for Cricket in India (BCCI), and it is followed by millions of people worldwide. The league consists of eight teams competing with each other. All the teams are built by Franchise owners and other sponsors. Every year the Franchise team owners spend a huge amount of budget to buy players through auction procedures. So, for every year the team needs to rebuild the squad and for rebuilding, the owners of the team have to buy the best players in auctions which are held on a particular day [2].

In modern sports, recognising the next big talent is as much science as it is art. As everyone knows how competitive sports have become in the past years, the players and the coaches constantly have to come up with some unique tactics to keep themselves in the game. Another important factor is to develop new playing strategies and to scout the best and suitable players who fit those strategies. All of these works are very tedious, time and effort consuming tasks for the team managers.

The reason why this topic has been chosen is because cricket is a popular sport and the interest in it has been highly increasing recently [3]. Other than this based on the market research it results in lack of advanced science and technology for predicting player’s performance and there is no standard criteria for evaluating player performance which results in poor prediction. Overall, the existing manual process is complicated and it leads to high risk of investment on buying players in auction. Another reason that led us to implement this project is the overshadowing that emerging players with good statistics are gotten by the famous players.

1.1 Problem Statement

The problem in the current system is that everything related to scouting a player is done manually, from checking past performance, checking techniques, deriving conclusions, which makes it very difficult for the management of the team to pick the right players for the right format. Another problem that can be considered here is that the teams have to spend a large amount of money on their manual scouting network which crosses many countries and sometimes even continents, which is also very important but it does not yield the result in comparison of the time, effort and money expended on it. Always keeping an eye on the players all over the work is very tedious work to do. Another significant problem is that the results concluded by the manual Scouters are also not accurate. The standard procedure of selecting a player is through an action and the problem that can be related here is that usually the owners tend to buy players who are popular even though they might not have performed well for the past few years. Consequently, the young or emerging players could be overshadowed by the famous cricketers. This leads to enormous risky investment decisions for the Franchise owners since it affects the overall team performance.

1.2 Proposed System

The proposed system has been developed for the sport of cricket and it shall quantify the performance of the players for T20 matches in terms of ratings by analysing the past performances of batsmen, bowlers and all-rounders separately. It will also analyze and predict the player's consistency using their long-term data and understand the pattern. The model rates the performers of different expertise such as batsman, bowler and all-rounder and helps to shortlist the players to target in the auction for the team owners. The model's accuracy can be cross verified with the existing past statistics of the players after performance analysis through the model. The proposed model is a Machine Learning based application and it will be provided with a graphical user interface for the end- user, mentioning here the Franchise owners and the team coaches. Through this system all the players will be equally evaluated based on their performance. This leads to a fair competition for the emerging players too.

1.3 Scope

The scope of this project is to develop a web application powered by Machine Learning that has a user interface, which provides many views such as giving the list of top players for each respective position including: batsman, bowlers and all-rounders. Evaluating new players by entering their performance related parameters. Further, visualising the number of players within a certain budget range and other features. These topmost players are fed into a database, based on the performance score and its accuracy, which was trained and tested by the Machine Learning algorithm. The trained data is a collection of historical statistical data of the players participating in IPL. The target variable was identified and the accuracy was calculated on the obtained trained and test data.

1.4 Intended Audience

It is a user-interactive platform for the cricket administrators and franchise owners to select the topmost batsmen, bowlers and all-rounders based on their performance and scout the team for the upcoming cricket tournaments. This application can be used in any cricket franchise for better accuracy in scouting the players.

1.5 Project Objectives and Goals

Following are the objectives of the project:

1. Analysing and collecting the right dataset with necessary attributes for training the model for various player roles such as batsman, bowler, and all-rounders.
2. Identifying the right approach to create a target class variable.
3. Identify the suitable algorithms and perform data training to analyse the past performance pattern and quantify the player's performance. Based on the results, utilize the best algorithm for the web application to be created.
4. Develop a web application to help owners of the franchise cricket teams to scout or identify the best players to target in the auction.

The project can be considered completed after creating the completely functional trained and tested model with a User Interface, which provides functionalities like searching the best players for different roles, evaluating emerging players and visualising the budget of the players.

2. Literature Survey

According to the Market research analysis, there are machine learning approaches for predicting the player's performance.

A neural network algorithm approach is used to predict the Indian's cricket bowler performance with the data containing 7 International matches played, where the resultant bowlers are the prediction of the performance based on the other model comparison [4].

In 2020, [5] according to the International Research Journal of Engineering and Technology, the prediction algorithm is used based on the venue, a number of innings played against the opponents, current form and past records as the major selection formula. Every batsman and bowlers are predicted using multiple algorithms with the limited matches considered. And the model is applicable only for T20 matches.

As per S. R. Iyer and R. Sharda, [6] A heuristic rating approach is employed to rate the player's performance in athletes, later the same strategy is used to predict the player's performance in cricket. The committee of cricket analyses the ratings of batsmen and bowlers received through this model and scout's the players for the matches.

Lemmer, H. H, [7] T20 cricket world cup series was analysed to get the player's performance result. The match was played in South Africa, where batsmen and bowlers are given a ranking system through this world cup series.

Lewis, [8] A. J: The Duckworth/Lewis model used to analyse players' performance in one-day international cricket, which is a well-established method for fair player analysis.

2.1 Existing Solutions

In order to analyse the existing system, available applications in the market were researched. There were not many applications related to AI or analytics for cricket, below are the tabulated applications figured out to look, how AI is being used in other sports and what are the features they are proving to the end-users. Since it was quite difficult to get the details about the exact algorithm or the technology stack for the existing applications.

Table 1. Existing systems comparison

Organization & Product	**Functionalities**	**Algorithm/ Tech**	**Targeted Sport**	**Targeted Region**	**Data Source**
Sport-Logiq	Making smarter decisions by	Computer vision and	Ice hockey, Football	America and Canada - Ice	Games stats

	utilizing analytics products.	ML (Algo name not mentioned)	and American Football	hockey, football and American football	
Spark Cognition	Evaluate student-athletes	Not mentioned	Baseball	States - University sports	Camera using computer vision to track players
IBM - JAAI SCOUT	Provide a deeper perspective on players.	IBM Watson Knowledge Studio and Watson Natural Language	Football	Germany-Bundesliga	Compiled scouting report collected from Bundesliga and 2. Bundesliga
Ludimos	Tracks players performance, Video analysis	Computer Vision	Cricket	Worldwide	Players Performance Videos
AiSCOUT	Insights into the player's technical, athletic, cognitive, and psychometric ability via video recognition technology.	Video recognition technology along with AI Algorithm (Name not mentioned).	Football	England-Premier League	From videos via video recognition technology.
HomeCourt Drills	Live shooting analytics that can't be seen with the naked eye.	Computer vision along with Deep Learning	Basketball	United States	Recorded Video source
Digital Cricket Scouting System	Helps Identifying the best cricket players by predicting their performance	Data Analysis and Machine Learning Algorithms	Cricket	India (Indian Premier League)	Stats of the players from various leagues from different countries

Furthermore, there are a couple of leading companies in sports analytics that collect large data related to many sports and use them for their analysis and metrics. Examples of such companies are Opta Sports, based in London, which is specialised in providing sports metrics and parameters stats. Opta Sports covers more than 30 sports from over 70 countries [9].

Another such company is Wyscout, based in Italy. This company specialises in football analytics and is one of the most used technologies used for scouting and potential transfers in European football[10].

3. Requirement Analysis

The requirement analysis section describes and highlights all the functions of the applications that are developed in order to give a better understanding of what the end-user expects from the application. It gives details about user requirements, functional requirements and non-functional requirements of the system.

3.1 User Requirements

Following are the end user requirements for this application:

- End-users should be provided with a search option to identify the players by choosing different options they are interested in.
- They should be able to access specific data regarding a specific player.
- End users should be able to evaluate new or emerging players by manual statistics entry.
- The players should be grouped by their price tags, so that the end users can see the range of players for a particular price tag

3.2 Functional Requirements

Functional requirements are the functions performed by every page in the application.

- The Digital Cricket Scouting system should allow end-users to search for players to target in the auction.
- The system should provide the best result based on the user's requirements.
- The system should display predicted performance details of the players as graphs.
- The administrator should be able to manage the data through the database.

3.3 Use Case Diagrams

3.3.1 End-User Use Case

The franchise owner or coaches in Digital Cricket Scouting System are given the following functionalities:

- Search for players list to target in the auction: This functionality is used to obtain the performance of the players present in the recently ongoing cricket league.
- View predictions of batsman’s performance: This functionality is used to manually input different features of a batsman and obtain prediction if the player is good or bad.
- View predictions of bowler’s performance: This functionality is used to manually input different features of a bowler and obtain prediction if the player is good or bad.

- View predictions of all-rounder’s performance: This functionality is used to manually input different features of an all-rounder and obtain predictions if the player is good or bad.
- Access several visualizations provided as graphs regarding players statistics: This functionality is used to view the graphs like count of players graph and other analysis graphs.

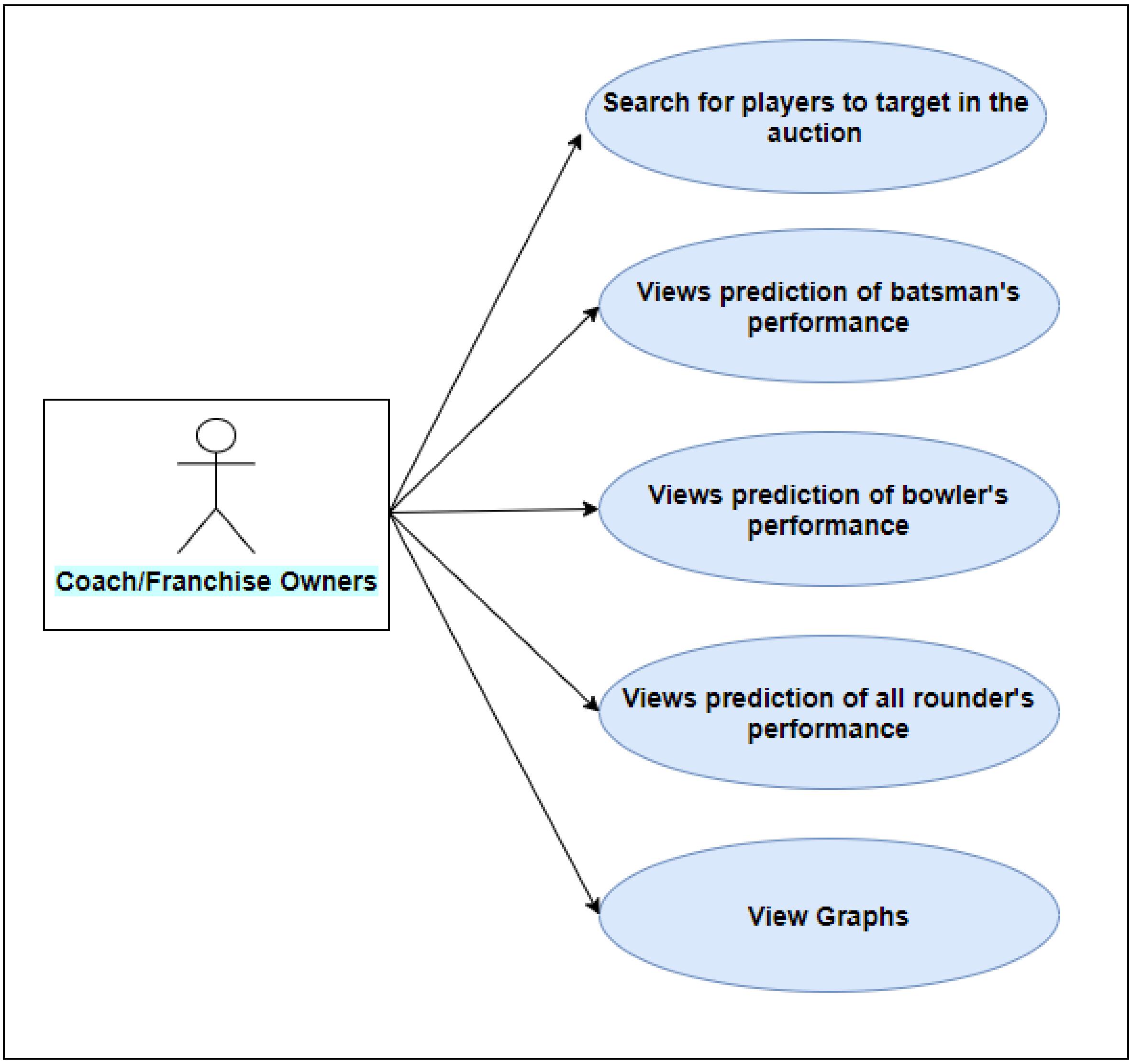

Search for players to target in the auction
Views prediction of batsman's performance
Views prediction of bowler's performance
Views prediction of all rounder's performance
View Graphs
Coach/Franchise Owners

Figure 1. End-user use case diagram

3.3.2 Admin Use Case

The administrator of the Digital Cricket Scouting System can perform the following functionalities:

- Add player’s statistics: This functionality is used to add new player and statistics into the application.
- Delete player’s statistics: This functionality is used to delete players and statistics from the application.

- Update player’s statistics: This functionality is used to update already existing player statistics.
- Manage the overall auction players list: This functionality is used to upload and update the list of auction players.

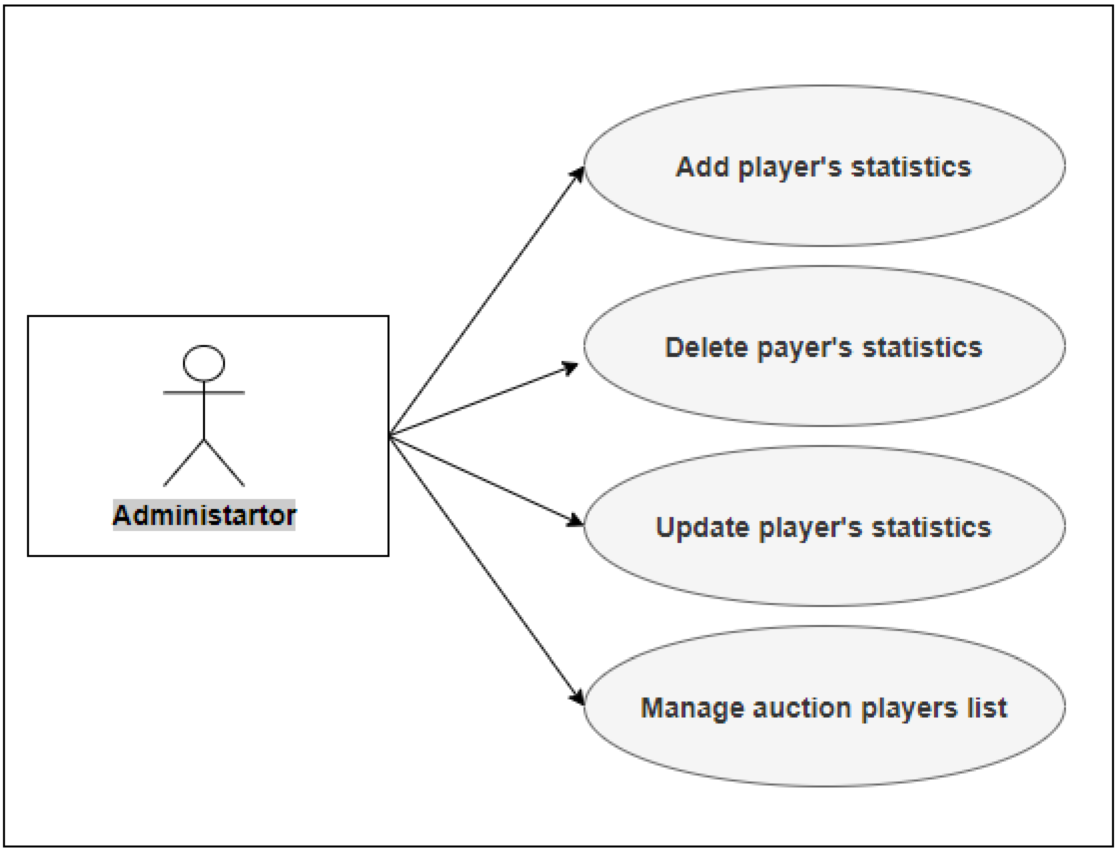

Administartor
Add player's statistics
Delete payer's statistics
Update player's statistics
Manage auction players list

Figure 2. Admin Use Case Diagram

4. System Architecture

The architecture of this project is composed of four components:

- **User Interface - End-user's dashboard**: The user interface or dashboard provides functionalities based on user requirements. The UI is developed using React JS.
- **Backend API**: While accessing the dashboard, the application sends requests to the system in order to get data of what the user is searching, for example, performance of cricket players according to roles or positions. Python flask APIs are developed to transmit the data between dashboard and database. Json format is used as input and output for the API. In this way, the software offers the users a real-time service they are demanding by generating the data automatically.
- **Database**: the data used by the application is stored in PostgreSQL. It consists of tables and views for the implementation of different functionalities in the application.
- **Machine Learning Model**: Once the data gets pre-processed, cleaned and organized, the data which is used for training purposes gets incorporated with the Machine Learning Algorithm. Both these components accomplish the prediction or decision-making process of the system. The model is trained and saved in .pkl file format to be used whenever necessary by the application.

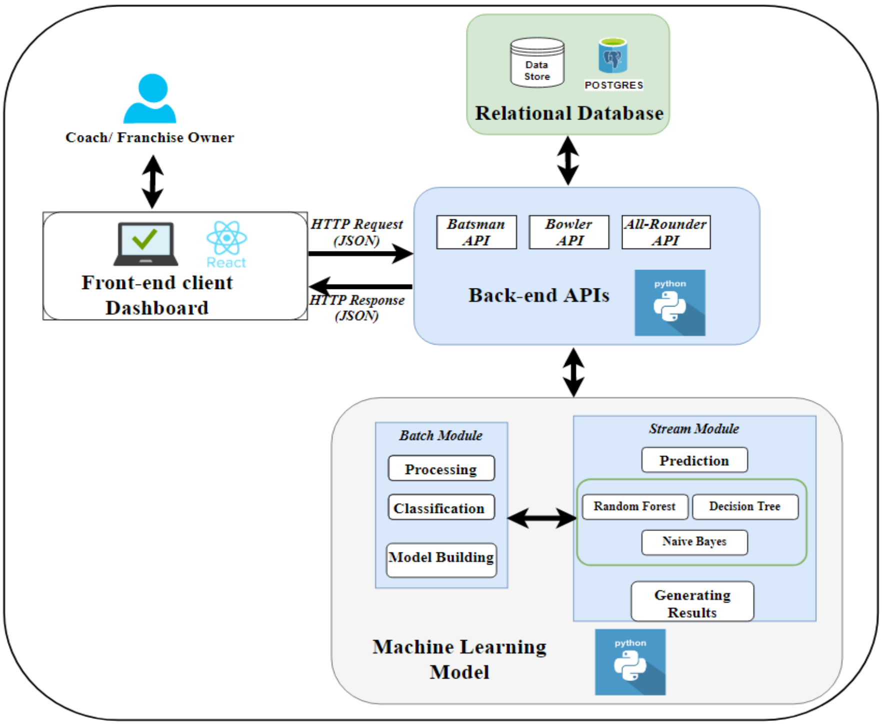
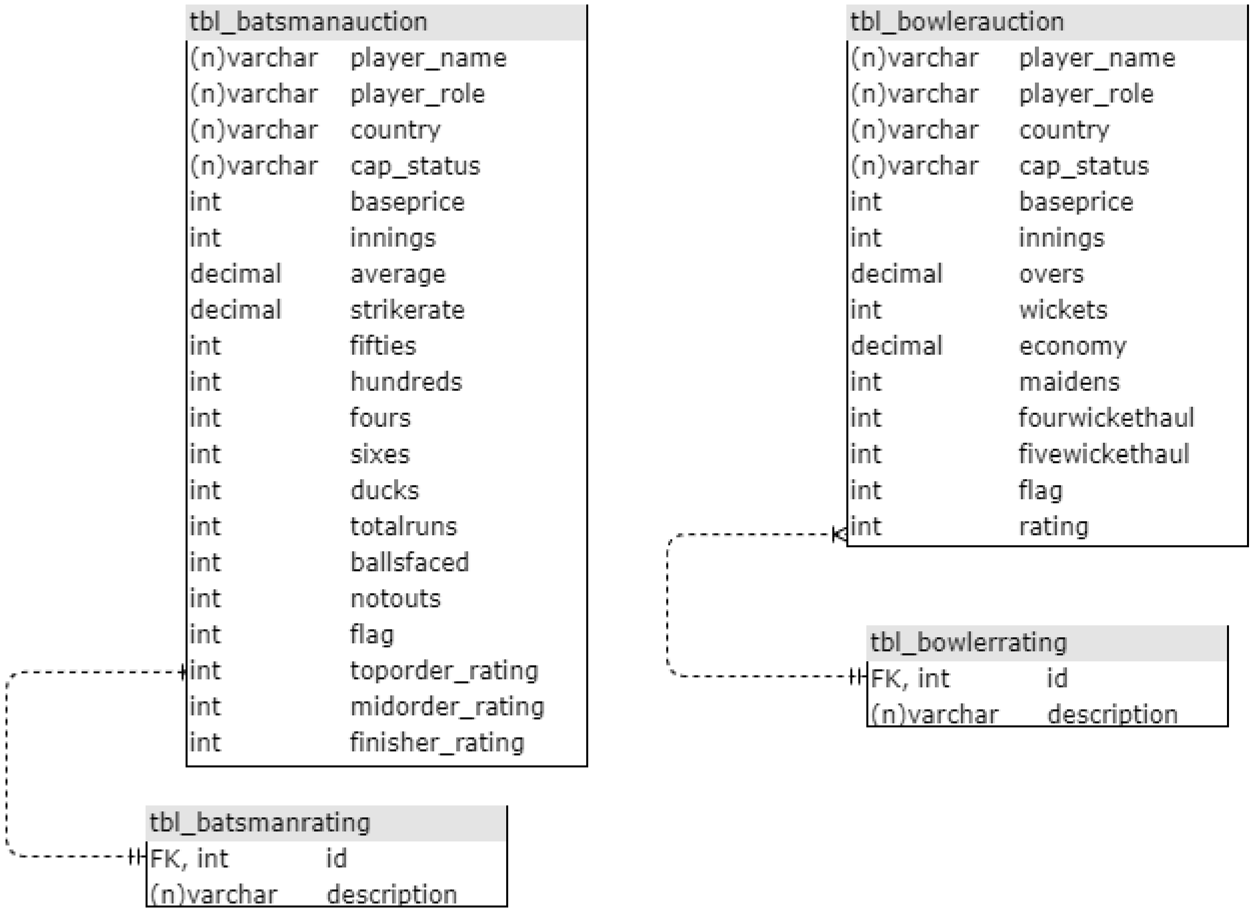


Figure 3. System Architecture

5. Database Architecture

The database used by the application is PostgreSQL database. the database architecture consists of 4 major tables:

- tbl_batsmanauction: is a database table used to store the list of batsmen participating in the auction of ongoing cricket league.
- tbl_batsmanrating: is a database table used to store the description of the predicted ratings from 1-5.
- tbl_bowlerauction: is a database table used to store the list of bowlers participating in the auction of ongoing cricket league.
- tbl_bowlerrating: is a database table used to store the description of the predicted ratings from 1-8.



tbl_batsmanauction	
(n)varchar	player_name
(n)varchar	player_role
(n)varchar	country
(n)varchar	cap_status
int	baseprice
int	innings
decimal	average
decimal	strikerate
int	fifties
int	hundreds
int	fours
int	sixes
int	ducks
int	totalruns
int	ballsfaced
int	notouts
int	flag
int	toporder_rating
int	midorder_rating
int	finisher_rating

tbl_batsmanrating	
FK, int	id
(n)varchar	description

tbl_bowlerauction	
(n)varchar	player_name
(n)varchar	player_role
(n)varchar	country
(n)varchar	cap_status
int	baseprice
int	innings
decimal	overs
int	wickets
decimal	economy
int	maidens
int	fourwickethaul
int	fivewickethaul
int	flag
int	rating

tbl_bowlerrating	
FK, int	id
(n)varchar	description

Figure 4. Database Architecture

6. Data flow Diagram

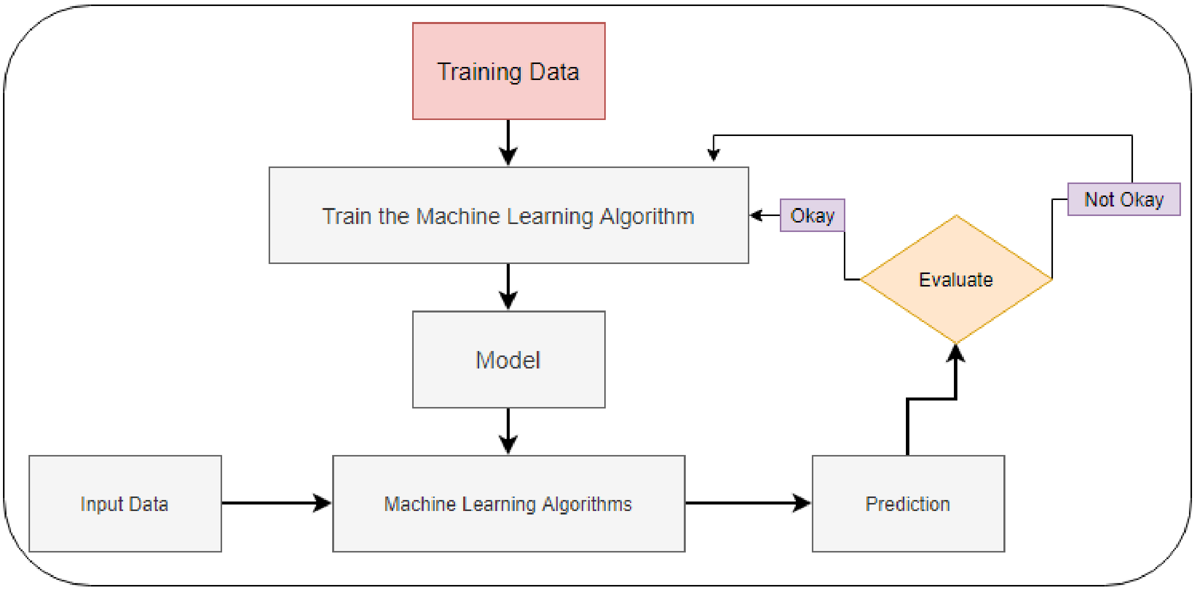

Training Data
Train the Machine Learning Algorithm
Okay
Not Okay
Evaluate
Model
Input Data
Machine Learning Algorithms
Prediction

Figure 5. Machine Learning Training Process

In both figures 5 and 6 are illustrated flow diagrams with the main focus on data on a machine learning system.

Input data is collected through various sources. Before training the dataset, input data is pre-processed to remove the erroneous data and optimise the missing information. This helps to improve the efficiency of the model which in turn increases the accuracy for training.

Once the dataset has been cleaned and organized according to system requirements, data is split into testing and training purposes. This split data is required for training the machine learning algorithm. Then the algorithm builds the model based on training data, which is further processed to predict the model accuracy score.

After a successful prediction, the model is evaluated for satisfactory performance. If the performance is as expected, further deployment through back-end API calls is carried out. Otherwise, the model is looped starting at data preparation until achieving the required performance.

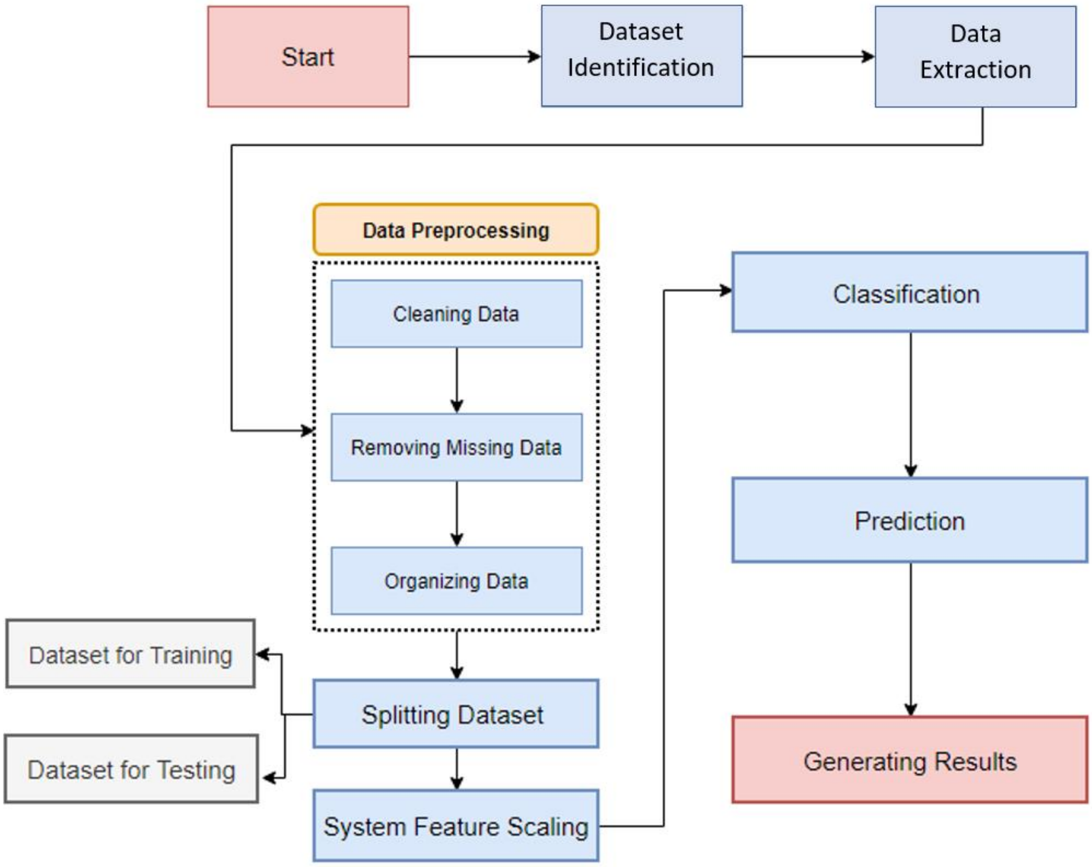


Figure 6. Business Logic diagram of the system

The overall business logic of the system is represented by listing the activities of both actors in the system. In general, it shows how the system works for both types of users, for franchise owners/ coaches and for the administrator of the system.

Franchise owners can access the web-based application without log in. They directly access the dashboard of the website using application home page, search players, access visualizations of several graphs and the about page which gives information about the latest news regarding cricket and cricket players.

The main functionality offered by the system is finding the best player for each playing position based on user requirements. During the search process, the end users can search about bowlers, batsmen, all-rounders and wicket keepers. If the system responds properly to the user's requirements, data is fetched for the player's rating which is generated by the trained machine learning model. In the case of a successful search, the system generates data from the central database. Otherwise, if the system cannot generate a rating about the searched player it will lead to an error page. As for the administrator part of the system, it plays a major role by managing all data regarding players. The administrator can update the player's statistics, add data of new players and delete existing player’s statistics if they are out of the auction list.

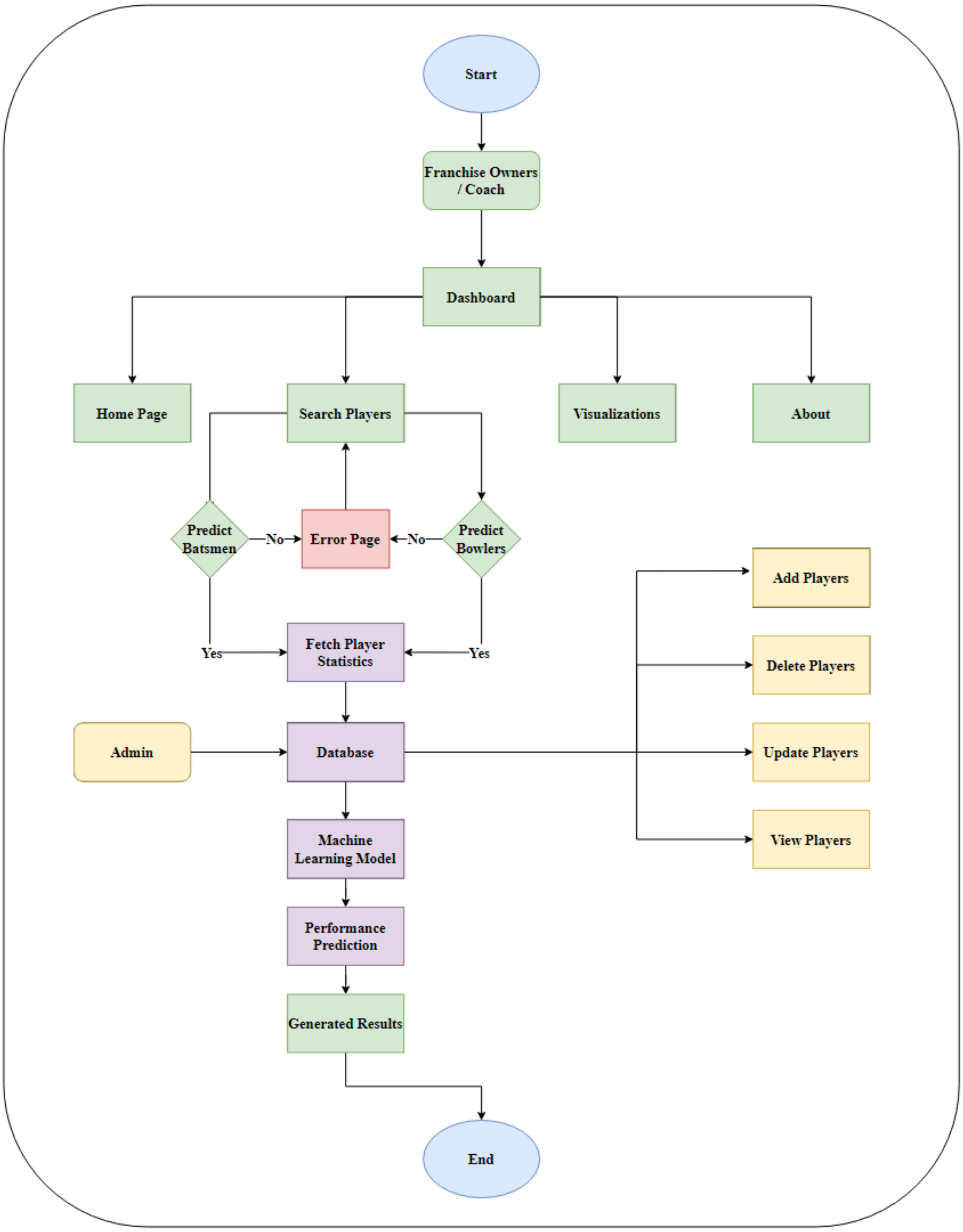


Figure 7. Overall data flow diagram

7. Technologies Used

7.1 Machine Learning Algorithm and Model

Machine Learning: It is a branch of Artificial Intelligence, where applications are developed to learn and improve the accuracy of data over a while. It can also be defined as predefined steps that a system/machine goes through to learn and train itself. Machine learning algorithms, when provided with a data set, eventually turn into a machine learning model. The larger the size of the data given to the machine learning algorithm, the better it can learn or train itself. In other words, it can be stated that the larger the dataset, the higher the accuracy of the model will be.

ML Model: A model in machine learning is a state of a system that has learned or trained itself based on some algorithm and represents what it has learned[11].

The primary step of any machine learning model is to get a dataset from which it can recognize the pattern, structure, attributes of the sample by which it is going to learn. It saves the rules, numbers, or any other data structures associated with the training dataset parameters in its own data structures that can be as simple as a nested if-else statement or as complex as layers of the neurons present in the brains.

7.1.1 Data Collection

Data is the most basic and essential component in machine learning and artificial intelligence. Two major data sets are required for the digital cricket scouting system, namely the train model dataset and auction dataset. Each of these datasets contains a subdivision of the batsmen dataset and bowling dataset.

Data was collected from various cricket information sites. Web crawling of the websites and direct download was used for collecting the data[12]. Data collection was done based on the requirement for our machine learning model, where batsmen and bowlers match information is collected from the very first season of the IPL, i.e., 2007 to 2021. The obtained data is focused on both local and international players of the IPL.

The train model dataset consists of approximately 13000 rows of data for batsmen and 8000 rows of data for bowlers. This data was used to train the machine learning model and also test the model's learning using evaluation parameters. It contains statistics of T20 league cricket players from the year 2007 to 2019.

The auction dataset consisted of approximately 175 rows of data for batsmen and 150 rows of data for bowlers. This data consisted of statistics of players participating in the recent IPL cricket league. It was used to predict the performance of those players for the upcoming IPL to verify the implementation of a digital cricket scouting system. It consisted of statistics of T20 league cricket players from the year 2020 to 2021.

Table 2. Sources of Data

Data Source	Source Description
www.espncricinfo.com	Data for most of cricket league
https://www.iplt20.com	Data for IPL matches - India, Auction list of year 2020
https://www.cricket.com.au/big-bash	Data for Big Bash matches - Australia
https://www.ecb.co.uk/county-championship	Data for English county matches - England
https://www.cplt20.com/	Data for Caribbean country matches - Caribbean
http://www.cricmetric.com/index.py	Data for Auction list Statistics
http://bigbashboard.com/	Data for multiple leagues

7.1.1.1 Description of the data set features

Batting Attributes

Batsman is the player who strikes the ball with a bat and makes runs between the wickets through the field or can strike for four or six runs by hitting the ball towards the boundary. Batsmen are required to score the runs per over and play with different techniques and they are responsible for setting or chasing the target runs to win matches for the team. Following are the attributes used to calculate the performance of the batsman[41].

Innings: It is the total number of times the batsman got a chance for batting, and the experience of the batsmen is derived based on the total number of innings played.

Average: It is the most important attribute for any batsman to showcase as the best player, the average determines the number of runs scored by a batsman and divided by the number of dismissed innings.

Strike rate: It is the second most important attribute for a T20 batsman to figure as a best player. It is calculated as the number of runs scored per 100 balls faced by the batsmen.

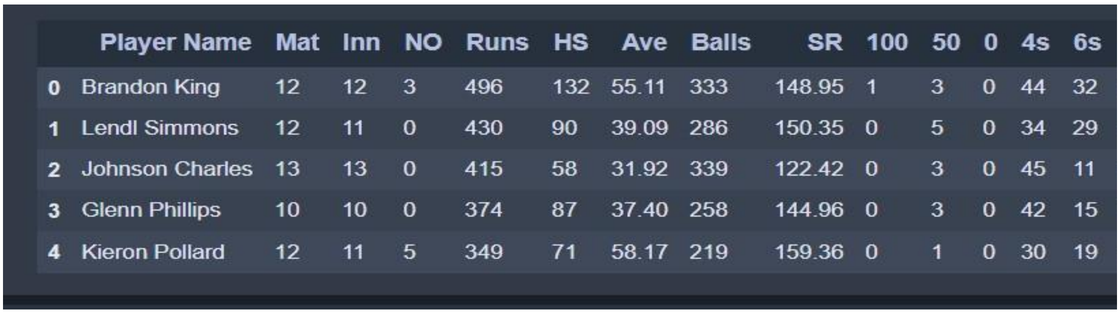
No of Hundreds: Total number of hundreds scored by batsmen per innings.

No of fifties: Total number of the fifties scored by batsmen per innings,

Highest Score: Batsmen’s highest score throughout his matches played.

No of fours: Total no of 4’s scored by batsmen per innings.

No of sixes: Total no of 6’s scored by batsmen per innings.



	Player Name	Mat	Inn	NO	Runs	HS	Ave	Balls	SR	100	50	0	4s	6s
0	Brandon King	12	12	3	496	132	55.11	333	148.95	1	3	0	44	32
1	Lendl Simmons	12	11	0	430	90	39.09	286	150.35	0	5	0	34	29
2	Johnson Charles	13	13	0	415	58	31.92	339	122.42	0	3	0	45	11
3	Glenn Phillips	10	10	0	374	87	37.40	258	144.96	0	3	0	42	15
4	Kieron Pollard	12	11	5	349	71	58.17	219	159.36	0	1	0	30	19

Figure 8. Attributes of batsmen dataset

Bowling Attributes

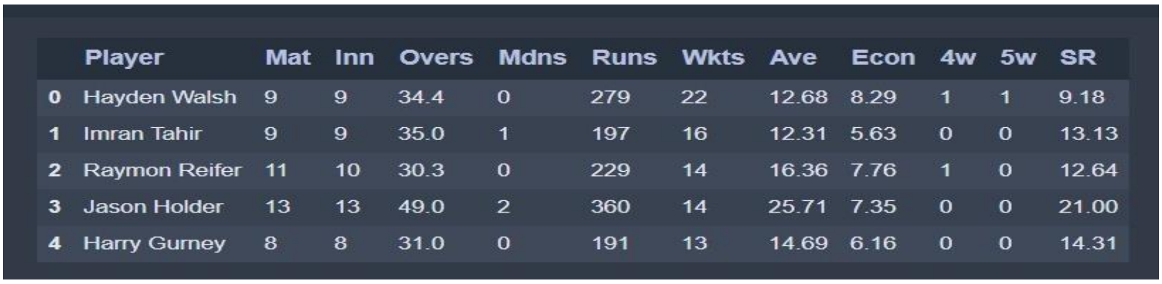
Bowler is the one who is responsible for defending the runs against the batsmen and taking wickets with good seam position and ball swinging techniques, any bowler to showcase as a best player, they should have the ability to contribute to fewer runs and take maximum wickets per innings.

Bowler Innings: It is the total number of chances the player got to bowl, and the experience of the bowler is measured by the total number of innings played.

Overs: The total number of overs bowled by the bowler; this is also another attribute where the experience of the bowler is measured. One over is equal to six balls.

Average: It is the important attribute for any bowler to maintain a good bowling average. It is calculated as Total number of runs saved by the bowler per wicket[40].

Economy rate: It is the next important attribute for any bowler, It defines the average number of runs conceded per overs bowled[40].



	Player	Mat	Inn	Overs	Mdns	Runs	Wkts	Ave	Econ	4w	5w	SR
0	Hayden Walsh	9	9	34.4	0	279	22	12.68	8.29	1	1	9.18
1	Imran Tahir	9	9	35.0	1	197	16	12.31	5.63	0	0	13.13
2	Raymon Reifer	11	10	30.3	0	229	14	16.36	7.76	1	0	12.64
3	Jason Holder	13	13	49.0	2	360	14	25.71	7.35	0	0	21.00
4	Harry Gurney	8	8	31.0	0	191	13	14.69	6.16	0	0	14.31

Figure 9. Attributes of bowlers data

7.1.2 Data Pre-processing

Data set consists of Cricketers batsman and bowlers' records from the year 2007 to 2019. It is analysed to obtain the target variable through different methods, for feature reduction. Many attributes without any importance on target variables are dropped. Python's libraries such as pandas and numpy were used to clean and analyse these datasets as a data frame[13].

7.1.3 AHP

The Analytic Hierarchy Process (AHP) is a comprehensive theory of measurement developed by T. L. Saaty, a mathematics professor from Iraq, in 1971- 1975 (University of Pennsylvania, Philadelphia, Pa). AHP is a nonlinear framework for inference-based and instigative reasoning by analysing several attributes simultaneously, allowing for dependence and feedback, and making mathematical trade-offs to deduce a conclusion. It is used in multicriteria decision-making to obtain relative measures from both discrete and continuous paired comparisons[14].

Steps involved in AHP

There are several steps involved in AHP for finding the optimum weight allocation to the attributes during multi-criteria decision-making. The steps are as follows:

1. Develop the hierarchical structure with three levels
    - ➢ Level One – Goal
    - ➢ Level Two – Attributes/Criteria
    - ➢ Level Three – Alternatives

Several player roles are there while choosing a cricket team. Each player role has its unique attributes, which helps identify the best player for that role[15].

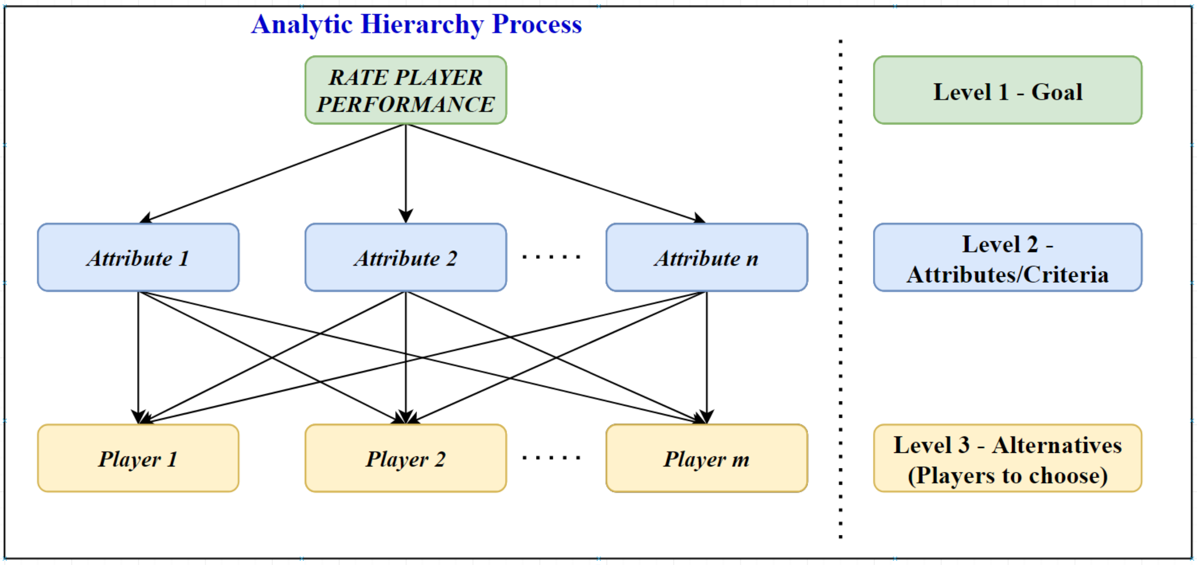


Figure 10. AHP basic structure[15]

2. Formulate a pair-wise comparison matrix and determine the relative importance of various attributes or criteria concerning the goal.

There are specific approaches to make an analytical comparison of two elements using AHP. Nevertheless, the relative weight scale between the two alternatives(attributes) proposed by Saaty (SAATY, 2005) is the most used. The scale finds the relative importance of an option compared to another option by assigning values that range from 1 to 9. A pair-wise matrix should be created with the help of the relative scale of importance, as shown in table 3.

Table 3. Saaty's Relative Scale of Importance[16]

Fundamental Scale (Row vs Column)	**Rating**
Extremely less important	**1/9**
Extremely to Very strongly less important	1/8
Very strongly less important	**1/7**
Very strongly to strongly less important	1/6
Strongly less important	**1/5**
Strongly to Moderately less important	1/4
Moderately less important	**1/3**
Moderately less to Equally important	1/2
Equally important	**1**
Equally important to Moderately more important	2
Moderately more important	**3**
Moderately to Strongly more important	4
Strongly more important	**5**
Strongly to Very strongly more important	6
Very strongly more important	**7**
Very strongly to Extremely more important	8
Extremely more important	**9**

Table 4. Pair-wise comparison matrix

	Attribute 1	**Attribute 2**	**…**	**Attribute n**
Attribute 1	$A_{11} = 1$	A_{12}	…	A_{1n}
Attribute 2	A_{21}	$A_{22} = 1$	…	A_{2n}
…	…	…	…	…
Attribute n	A_{n1}	A_{n2}	…	$A_{nn} = 1$

The decision-maker should fill the Pairwise matrix. For example, the first pair of attributes should be compared, i.e., Attribute 1 and Attribute 2. The decision-maker should answer the question, how relevant is Attribute 1 when compared to Attribute 2. Using Saaty's relative scale, the decision-maker thinks Attribute 1 is very strongly more important than Attribute 2, so A_{12} will be assigned a value of 7. Correspondingly A_{21} will be given a value of 1/7 since A_{21} is inversely proportional to A_{12}. Subsequently, all the matrix columns should be filled similarly. All the diagonal elements will have a value of 1 because the attribute is compared to itself[17].

3. The pairwise matrix with assigned weights should be normalized and updated with the new normalized weights.

Sum of the weights of each attribute should be calculated column-wise. Let's consider n as the total number of attributes, row as i, column as j, where i = 1, 2, …, n and j = 1, 2, …, n.

Table 5. Pairwise comparison matrix with Sum of weights

	Attribute 1	Attribute 2	…	Attribute n
Attribute 1	$A_{11} = 1$	A_{12}	…	A_{1n}
Attribute 2	A_{21}	$A_{22} = 1$	…	A_{2n}
…	…	…	…	…
Attribute n	A_{n1}	A_{n2}	…	$A_{nn} = 1$
SUM S_j	S_1= SUM (A_{11}:A_{n1})	S_2= SUM (A_{12}:A_{n2})	…	S_n= SUM (A_{1n}:A_{nn})

$S_j = \sum_{i=1}^{n} A_{ij}$ where S_j is column-wise sum

$W_{ij} = A_{ij} / S_j$ where W_{ij} is Normalized Weight

Matrix should be updated with normalized weights as shown in table 6.

Table 6. Pairwise comparison matrix with Normalized weights

	Attribute 1	Attribute 2	…	Attribute n
Attribute 1	$W_{11}=(A_{11}/S_1)$	$W_{12}=(A_{12}/S_2)$	…	$W_{1n}=(A_{1n}/S_n)$
Attribute 2	$W_{21}=(A_{21}/S_1)$	$W_{22}=(A_{22}/S_2)$	…	$W_{2n}=(A_{2n}/S_n)$
…	…	…	…	…
Attribute n	$W_{n1}=(A_{n1}/S_1)$	$W_{n2}=(A_{n2}/S_2)$	…	$W_{nn}=(A_{nn}/S_n)$

4. Criteria weights of each attribute should be calculated by averaging the normalized weights row-wise.

$C_i = \frac{1}{n} \sum_{j=1}^{n} W_{ij}$ where C_i is criteria weight

Table 7. Normalized Matrix with Criteria Weights

	Attribute 1	Attribute 2	…	Attribute n	Criteria Weights
Attribute 1	W_{11}	W_{12}	…	W_{1n}	C_1
Attribute 2	W_{21}	W_{22}	…	W_{2n}	C_2
…	…	…	…	…	…
Attribute n	W_{n1}	W_{n2}	…	W_{nn}	C_n

5. After identifying the criteria weights, it is mandatory to check the consistency of the weightage allocation. The identified criteria weights and initially created a non-normalized pairwise matrix in step 2 should be used for calculating weighted sum value.

$$V_i = \sum_{j=1}^{n} A_{ij} * C_i \qquad \text{where } V_i \text{ is weighted sum value}$$

Table 8. Pairwise Comparison Matrix with Weighted Sum Value

Criteria Weights	C_1	C_2		C_n	
	Attribute 1	**Attribute 2**	**…**	**Attribute n**	**Weighted Sum Value**
Attribute 1	$A_{11} = 1$	A_{12}	…	A_{1n}	V_1
Attribute 2	A_{21}	$A_{22} = 1$	…	A_{2n}	V_2
…	…	…	…	…	…
Attribute n	A_{n1}	A_{n2}	…	$A_{nn} = 1$	V_n

6. Once the weighted sum values are calculated, the ratio of weighted sum value to criteria weight should be calculated. The average of these ratios will be lambda max ($\boldsymbol{\lambda}_{max}$) value.

Table 9. Source table for λmax calculation

	Weighted Sum Value	**Criteria Weights**	**Ratio**
Attribute 1	V_1	C_1	V_1/ C_1
Attribute 2	V_2	C_2	V_2/ C_2
…	…	…	…
Attribute n	V_n	C_n	V_n/ C_n

$$\boldsymbol{\lambda}_{max} = \frac{1}{n} \sum_{i=1}^{n} (V_i / C_i)$$

7. $\boldsymbol{\lambda}_{max}$ is used for finding the Consistency Index (C.I). Consistency Index[18] can be calculated as follows:

$$C.I = \frac{\lambda_{max} - n}{n - 1}$$

8. Random Index should be identified based on the attributes count from Saaty's random index table. The random index value for attribute counts of 1 to 12 is mentioned in table 10.

Table 10. Saaty's Random Index[19]

n	1	2	3	4	5	6	7	8	9	10	11	12
R.I	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48

9. Consistency ratio (C.R) is the ratio of Consistency Index to Random Index.

$$C.R = \frac{C.I}{R.I}$$

10. The weightage allocation is feasible only if the Consistency ratio is less than 10%, and then the identified criteria weights of the attributes can be used to achieve the target goal. The weights should be recalibrated until a consistency ratio of 10% is achieved[20] (i.e., repeat steps 2 to 9 again).

Formula creation

AHP calculations were carried out through Microsoft Excel. AHP was used for creating the target class variables for evaluating batsman and bowler performance. Overall, attribute weights were calculated using AHP and utilized for creating four formulas, three for the batsman and one for the bowler. During the attribute weightage allocation, consistency was monitored. Table 11 shows the AHP consistency ratio, which is less than 10% for all the cases. Hence our identified weights were feasible.

Table 11. AHP Consistency table

	AHP Consistency Ratio (<10%)
Top-order Batsman	7%
Middle-order Batsman	7%
Finisher Batsman	9%
Bowler	6%

<u>Formula for Batsman:</u>

Among the 11 attributes chosen for the batsman performance evaluation, attributes such as batting average, strike rate, fours, sixes, the fifties, hundreds, balls faced, total runs scored, innings, and not outs are contributing to positive aspect, and getting out without scoring runs (i.e., zeroes or ducks) contributes to the negative aspect of the player evaluation. Based on the player's batting position, the batsman's role will change, and skill sets are also different. Hence, the performance of a batsman is also evaluated differently for each batting position. Based on the batting position, batsmen can be categorized as shown in table 12[21]:

Table 12. Batsmen Category

Player Type	Playing Position
Top-Order Batsman	1, 2
Middle-Order Batsman	3, 4, 5
Finisher-Batsman	6, 7, 8

Top-Order Batsman: This category of batsmen will have the possibility to face more balls than the rest of the batsmen since they play during the early stage of the batting innings. So, an excellent top-order batsman should have the skill to score more runs and play consistently. A top-order batsman needs to play longer innings and score safe runs by analysing the game situation rather than getting dismissed by trying to hit every ball from the beginning. Therefore, during AHP weightage allocation, for this batsman category, almost 25% weightage was given to batting average, 18% weightage to fours (safe runs), 15% weightage to strike rate, 10% weightage to total runs, and 5% weightage to balls faced.

*Top-Order Batsman Rating = (0.250*BattingAverage) + (0.152*StrikeRate) + (0.184*Fours) + (0.128*Sixes) + (0.099*TotalRunsScored) + (0.053*BallsFaced) + (0.030*50s) + (0.021*100s) + (0.036*Innings) + (0.021*NotOuts) - (0.025*Zeroes)*

Middle Order Batsman: A good middle-order batsman should have both the qualities of the top-order batsman and finisher batsman. If a top-order batsman gets dismissed early, then a middle-order batsman will have a chance to face more balls. In this case, he should play like a top-order batsman, which means that he should have the ability to score more runs and fifties, play consistently, and maintain a good batting average. On the other hand, sometimes, if a top-order batsman plays well for a long time, then a middle-order batsman will get a chance to face only fewer balls. Therefore, a good middle-order batsman should additionally have the skills to strike the ball for fours and sixes. Thus, during AHP weightage allocation, for this batsman category, an equal weightage of almost 23% each was given to batting average and strike rate, and an equal weightage of 15% each was given to fours and sixes.

*Middle-Order Batsman Rating = (0.230*BattingAverage) + (0.232*StrikeRate) + (0.150*Fours) + (0.154*Sixes) + (0.063*TotalRunsScored) + (0.047*BallsFaced) + (0.027*50s) + (0.023*100s) + (0.026*Innings) + (0.024*NotOuts) - (0.023*Zeroes)*

Finisher Batsman: This type of batsman will have a lower possibility to face more balls while batting. A good finisher batsman should have an excellent ball-striking ability to hit more fours and sixes and score quick runs in the fewer balls he faces. The chance of scoring more runs and the fifties is difficult for this kind of batsman. Based on all this, during AHP weightage allocation, almost 19% weightage was given to strike rate, 20% weightage to fours, and 24% weightage to sixes.

*Finisher Batsman Rating = (0.103*BattingAverage) + (0.187*StrikeRate) + (0.206*Fours) + (0.240*Sixes) + (0.080*TotalRunsScored) + (0.020*BallsFaced) + (0.035*50s) + (0.017*100s) + (0.028*Innings) + (0.056*NotOuts) - (0.028*Zeroes)*

Formula for Bowler:

Among the seven attributes chosen for bowler performance evaluation, innings, overs bowled, wickets were taken, maidens, four-wicket hauls, and five-wicket hauls contribute to the positive aspect and economy to the negative aspect of the player evaluation. Irrespective of the skillset or player style, the primary goal of the bowler is common for all, which is bowling defensively (i.e., ability to restrain batsmen from scoring runs) and taking more wickets (i.e., dismissing more batsmen). Therefore, during AHP weightage allocation, almost 37% weightage was given to wickets taken and 32% weightage to the economy.

*Bowler Rating = (0.103*Innings) + (0.187*OversBowled) + (0.206*WicketsTaken) + (0.080*Maidens) + (0.020*4WicketHaul) + (0.035*5WicketHaul)*
*- (0.240*Economy)*

After creating the formulas, the target class variable called player rating was created for the entire dataset. In case of all-rounders, both batting and bowling formulas were applied individually to identify the player's performance. The dataset with the updated target class variable was further used for data training of the machine learning model[22].

7.1.4 Data Training

To train the cleaned data from the previous step, complete analysis is employed, before selecting the algorithm, in order to best suit the model and the idea of the project. Dataset is split into training and test data and it is fed into the algorithm which best fits this data. The Python train-test-split module is utilized to train the dataset. Technologies used in python for the algorithm are:

Pandas: It is a fundamental building block to analyse the data. It is a simple, versatile package that makes it easy to handle and visualise the data. Every data in a dataset is represented in dictionary format and it is stored as a pandas.series.object. The two-dimensional data frame makes it easy to work with data in any stream[23].

Sklearn: It is an open source, simple, efficient and reusable package, consisting of various regression and classification algorithm modules to train and test the data. The modules are very easy to use and learn. It is built on Numpy, Scipy, and matplotlib. The algorithms can also be visualized through graphs using matplotlib[24].

Numpy: It is a mathematically distributed library for computations. Numpy is versatile, fast and easy to use. It can be used to compute the random generations, fourier transforms, linear regressions etc. It consists of high-level syntax which makes it possible for every user with less knowledge[25].

The underlying principle was to quantify the players according to their performance which is based on certain parameters of their performance, so that the players can be easily classified by the machine learning algorithms whether they are a good fit for the team or not. The formula that was used for the rating of the players was developed by the help of the Analytic Hierarchy Process. Then with the help of the performance formula, a new column was added in the dataset by the name of “performance” which was later used as a target parameter to train the model. Once the model is trained and validated, it is ready for use in the real world of sport. It will take an input of players' recent performance statistics and then the trained model will predict/rate the chances of players success which will be very useful for the teams’ coaches or managers.

Below are the algorithms that are differently handled to obtain the accuracy of prediction.

Naive Bayes:

The Naive Bayes algorithm works on the probability of a condition. It uses principles of the Bayes theorem for predicting models. Every attribute has its own class variable and it is independent of other attribute values, which is called class conditional independence. Bayes theorem determines the probability of a model on a condition[26].

Based on Bayes theorem:

- **$P(A|B) = P(B|A)\ P(A) / P(B)$**

  $P(A|B)$ is a posterior probability of hypothesis A forgiven event B

  $P(B|A)$ is a posterior probability of hypothesis B for given event A

  $P(A)$ is a prior probability

  $P(B)$ is a Marginal Probability
- Naive Bayes uses sklearn.naive_bayes to develop the model.
- The train_test_split module is used to test and train the data.
- The estimation of the probability is wrong sometimes for the trained data.

Decision Tree:

A Decision Tree is a tree-structured model with different branches to it, where every leaf node represents the class variable, and the internal tree node represents the attributes. The first node or the initial node is called the root node. Every leaf node is the outcome of the decision. It handles both regression and classification problems. It is easy to understand and visualize compared to other models. Training time for the dataset in a decision tree is faster than a neural network[28].

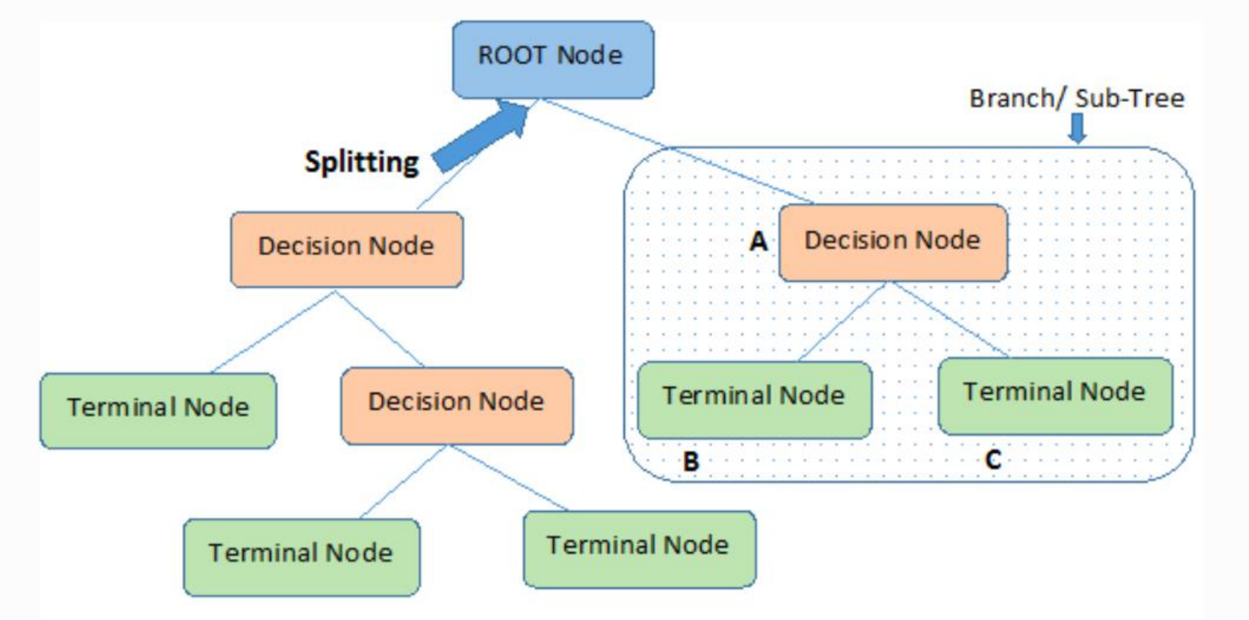


Figure 11. Working of Decision Tree algorithm [27]

Algorithm:

- The root node is split into many branches and the process is called splitting. Split is decided based on the condition on the attribute variable
- After splitting into nodes, a subset of the larger data is generated.
- The cycle is repeated recursively until the condition is met:
  - End of the attribute
  - All the values in the tuple are the same as the attribute.
- Unwanted branches are removed during dataset training called pruning.
- Inbuilt python library “scikit-learn” is used to develop the Decision Tree model.
- The Training and test dataset is generated using the train_test_split module, which is further used for prediction and accuracy of the model.

Random Forest:

Unlike the Decision tree, Random Forest does not use a single tree to decide the performance of the data. A minute change in dataset training would result in the change of the decision tree structure. Decision trees are expensive in computation, and it is overfitting, due to which Random Forest is chosen to train the dataset.

Random Forest is a classification algorithm where multiple decision trees are implied on the dataset to predict the accuracy of the data trained. As the number of trees grows in the algorithm, the more accurate the prediction and overfitting is resolved. Furthermore, every decision tree created runs parallelly without any collision[30].

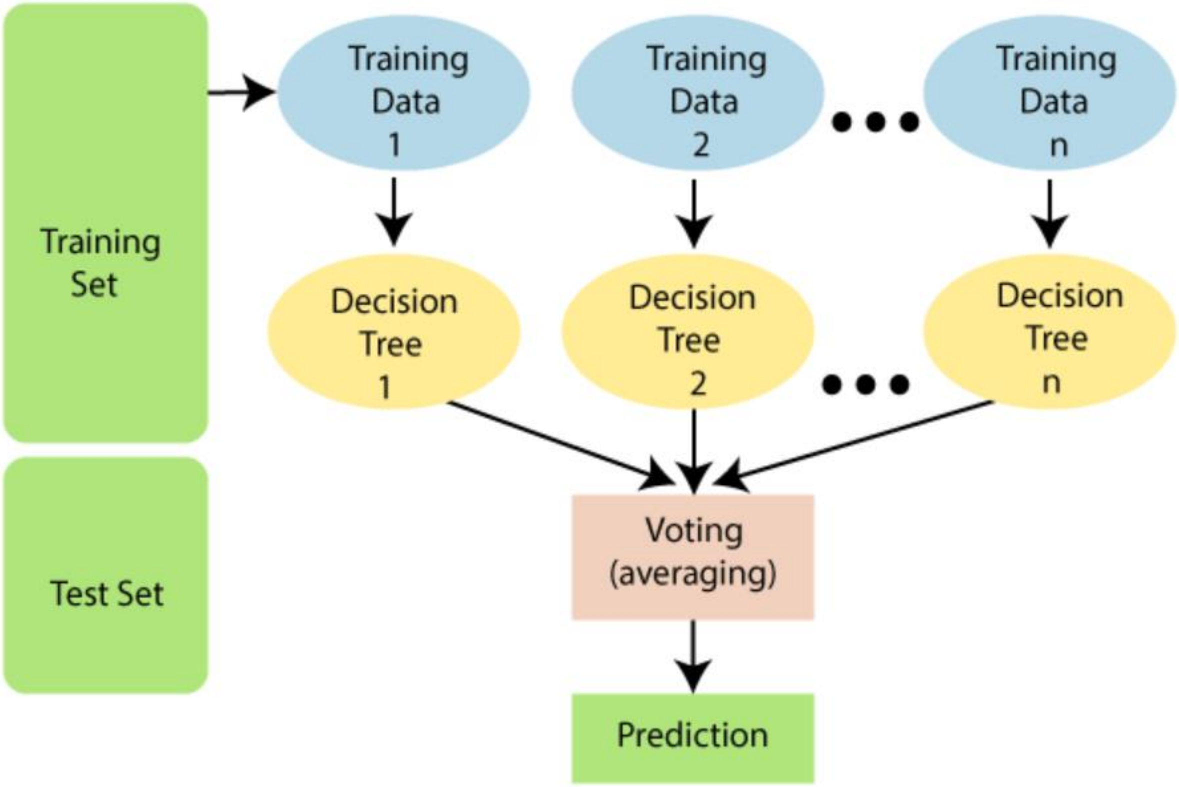


Figure 12. Working of Random Forest algorithm [29]

Algorithm:

- Every Data set is divided or split into a trained and tested dataset.
- The Trained dataset is further divided as a subset, every subset is a decision tree.
- The aggregation is decided on the voting method, and finally, the score is predicted.
- sklearn.ensembl.RandomForestClassifier library is used to build the Random Forest Classifier model.
- Random data k, is selected from the dataset.
- Decision tree is built from the selected data.
- Number N, is chosen from a decision tree that has to be built.
- The steps are repeated to find the new data point from the decision tree and based on voting the tree is decided for prediction.

7.2 Web App Development

7.2.1 Back-end Development

Python Flask Framework

The Flask framework in python is a popular and trending microweb framework, lightweight and easy to use. It communicates through a web server gateway interface which is called WSGI, which provides dynamic development and Testing. It helps in eliminating the dependency on the structure and monolithic architecture in Django Projects. Flask_Restful, Flask-sqlalchemy, psycopg2 are the libraries of python that make back-end API and other databases such as Postgresql to set up more comfort[31].

7.2.2 Database

Postgres

PostgreSQL is an Open-Source database management system, with the support of a large community, owned by a single company. PostgreSQL provides online support to debug the errors easily. And it contains a Postgres image that makes it easy to deploy with docker. Digital cricket scouting system uses PostgreSQL database to store the data sets trained through the model because of its easy debugging technology for the developers[32].

7.2.3 Front-end Development

React was first created and deployed for Facebook in the year 2011 by Jordan Walkie. It is an open-source front-end library for creating interactive user interfaces and reusable user interfaces for both web and mobile development. React has a larger community to support and it uses component-based logic for data passing. ReactJS improves the performance of the application through DOM features. Digital Scouting System contains various functionality that is being accessed by the Cricket franchise, hence the look and feel of the React improves the readability. Digital Scouting System should be very responsive, which is possible through React, which renders creating responsive screens and reusable components that make it easy for the application using React themes, across all the components[33].

Axios

To access the backend APIs, React provides a component called Axios with child function call-back. Axios contains features such as error, response, loading, async, awaiting while accessing the APIs, along with call-back props such as onSuccess, onError, and OnLoading. Digital Scouting System web application requires consuming and displaying several APIs fetched from the backend, which is accomplished through the Axios[34].

Reactstrap

Reactstrap is a ReactJs package which can be installed in a project with the help of Node Package Manager (NPM) and it has inbuilt support for the components of bootstrap like buttons, form elements, input fields etc. Having inbuilt bootstrap capabilities helps in the rapid and quick development of intuitive User Interface as the bootstrap does not need to be included or called externally. Reactstrap also has inbuilt form handling and validation capabilities[35].

8. Implementation

The system was implemented in two major parts: the machine learning model and the web application. The machine learning model was implemented for three algorithms, and the one with better results was chosen for the web application services. Web application comprises front-end user interface, back-end services, and a relational database management system for storing data. Front-end and back-end were integrated through Axios.

8.1 Machine Learning Model

For the implementation of the model, prebuilt python libraries (Imblearn and scikit-learn) were used which have the inbuilt functions to train the model[36][37. The dataset that has been used was processed via AHP, which generated the target variable with different derived formulas for both batsmen and bowlers. The training dataset consists of approximately 13000 rows, which was splitted in different ratios in form of training and testing data, and the splitting is done in the iteration of different ratios to get the machine learning algorithm performance so that it can be compared and decided which ratio of split gives the best output.

The splitting of the data set was done using the train_test_split function from the library. The splitting ratio is passed to this function as a parameter. Now once the data is splitted, the algorithm for which the model has to be trained is called via the scikit-learn library and the splitted dataset is passed to the model for the training. So, this process of calling the model and passing the splitted dataset was done for all the three algorithms (Decision Tree, Naive Bayes, Random Forest) for both batsmen and bowler. The model will take some time to train itself depending on the size of the dataset and once the model is trained it is saved in a pkl format so that this model is called via the APIs to give back the results.

During the training of the model(classification), sometimes there might be an issue of class mismatch between the classified classes, so in order to avoid this issue of mismatch, the library Imblearn is used. It up samples or down samples the classes dynamically according to the samples present in the dataset.

After the repetitive implementation, it was observed that the split ratio of 80:20 (Train:Test) for Random Forest gives the accurate output for the model. For batsmen we created 5 classes in the target variable with 5 being the highest and 1 being the lowest. For bowlers we created 8 classes in the target variable with 8 being the highest and 1 being the lowest.

8.2 Application Functionalities

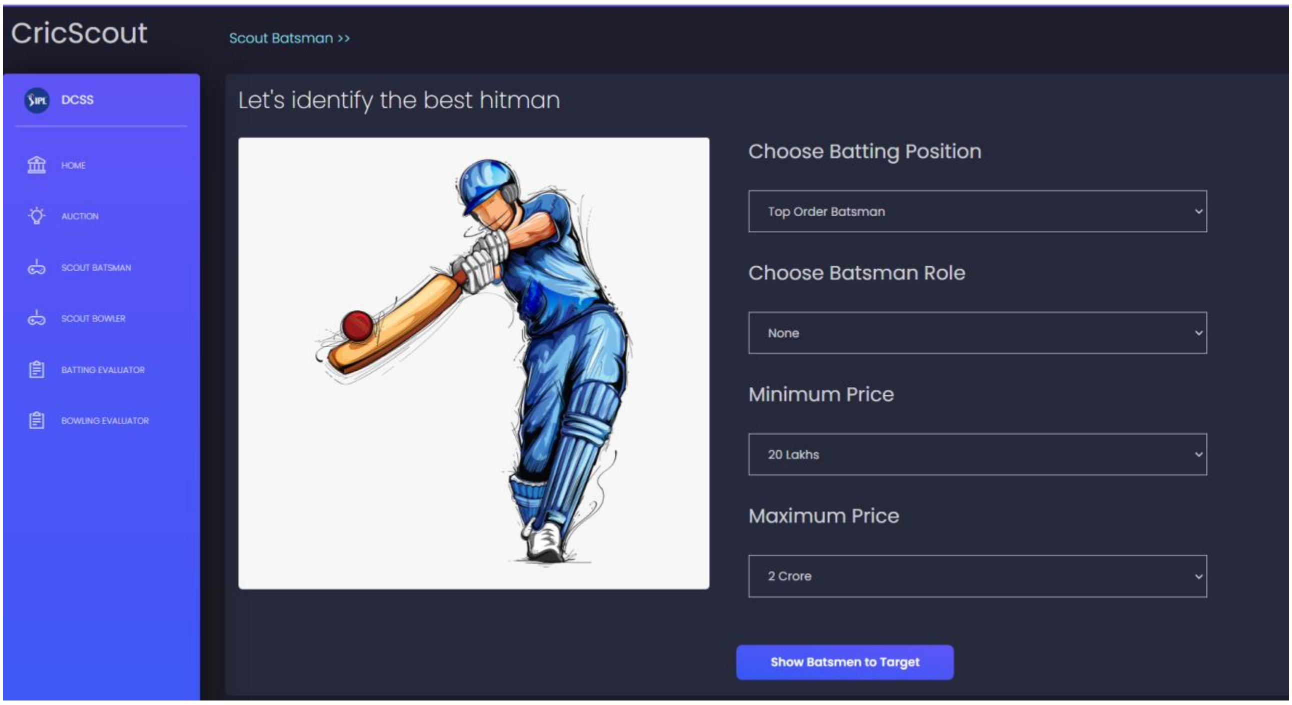
8.2.1 Scout Batsman

The first significant functionality of the Digital Cricket Scouting System enables the coaches or franchise owners to search the predicted performance of batsmen who are on the auction list. In addition, the user is provided with options to choose the batting position, specific role, minimum, and maximum base price to make their query more detailed.

User Interface:

As shown in figure 13, on clicking the scout batsman tab from the navigation bar, users can see the form to choose the options for getting the predicted batsmen list. The form contains three options:

1. The user can select the batting position of the batsman from the "choose batting position dropdown," and the possibilities are top-order batsman, middle-order batsman, and finisher batsman.
2. The user can choose the specific role of batman from the "choose batting role dropdown," and the options are none (no specific role), wicket-keeper, and batting all-rounder.
3. The user can select the minimum base price and maximum base price options for additional filters.



Let's identify the best hitman

Choose Batting Position: Top Order Batsman

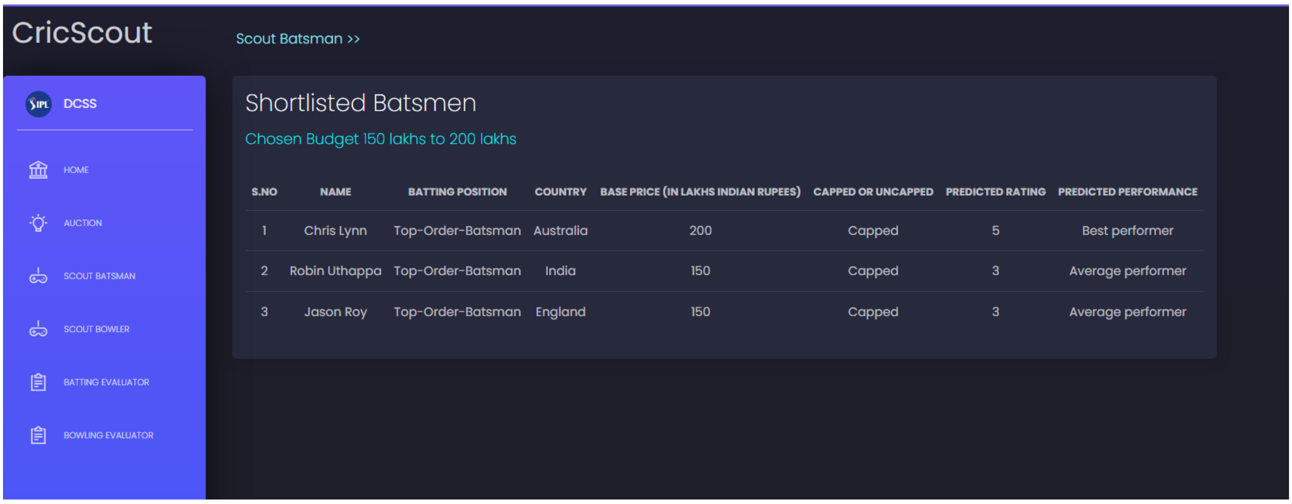
Choose Batsman Role: None

Minimum Price: 20 Lakhs

Maximum Price: 2 Crore

Figure 13. Landing Page of Application

After filling the form, on clicking the "Show Batsmen to Target" button, a list of batsmen with their predicted performance sorted by rating and auction base price will be fetched using the predict batsman back-end web service. The user can view the fetched list as shown in figure 14.



Shortlisted Batsmen

Chosen Budget 150 lakhs to 200 lakhs

S.NO	NAME	BATTING POSITION	COUNTRY	BASE PRICE (IN LAKHS INDIAN RUPEES)	CAPPED OR UNCAPPED	PREDICTED RATING	PREDICTED PERFORMANCE
1	Chris Lynn	Top-Order-Batsman	Australia	200	Capped	5	Best performer
2	Robin Uthappa	Top-Order-Batsman	India	150	Capped	3	Average performer
3	Jason Roy	Top-Order-Batsman	England	150	Capped	3	Average performer

Figure 14. Shortlisted Batsmen

Back-end Web Service:

This API is used to predict the performance of Batsmen who are in the auction list whose statistics are managed by the administrator in the backend database. All the batsmen are assigned with their respective batting positions such as top-order, middle-order and finisher/lower-order batsman and they have their own individual base price. This API helps to predict and display the top wise performance of the players which are predicted to be in form. This API predicts the results based on the chosen batting position, role and base price.

Request Type: POST

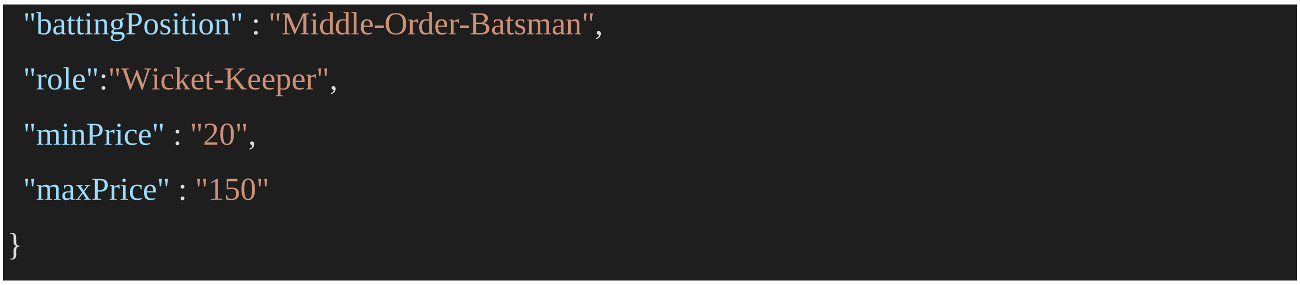
Request URL: /predictBatsman

Request Structure for only Batsman:

```
{
 "battingPosition" : "Top-Order-Batsman",
 "role":"none",
 "minPrice" : "20",
 "maxPrice" : "150"
}
```

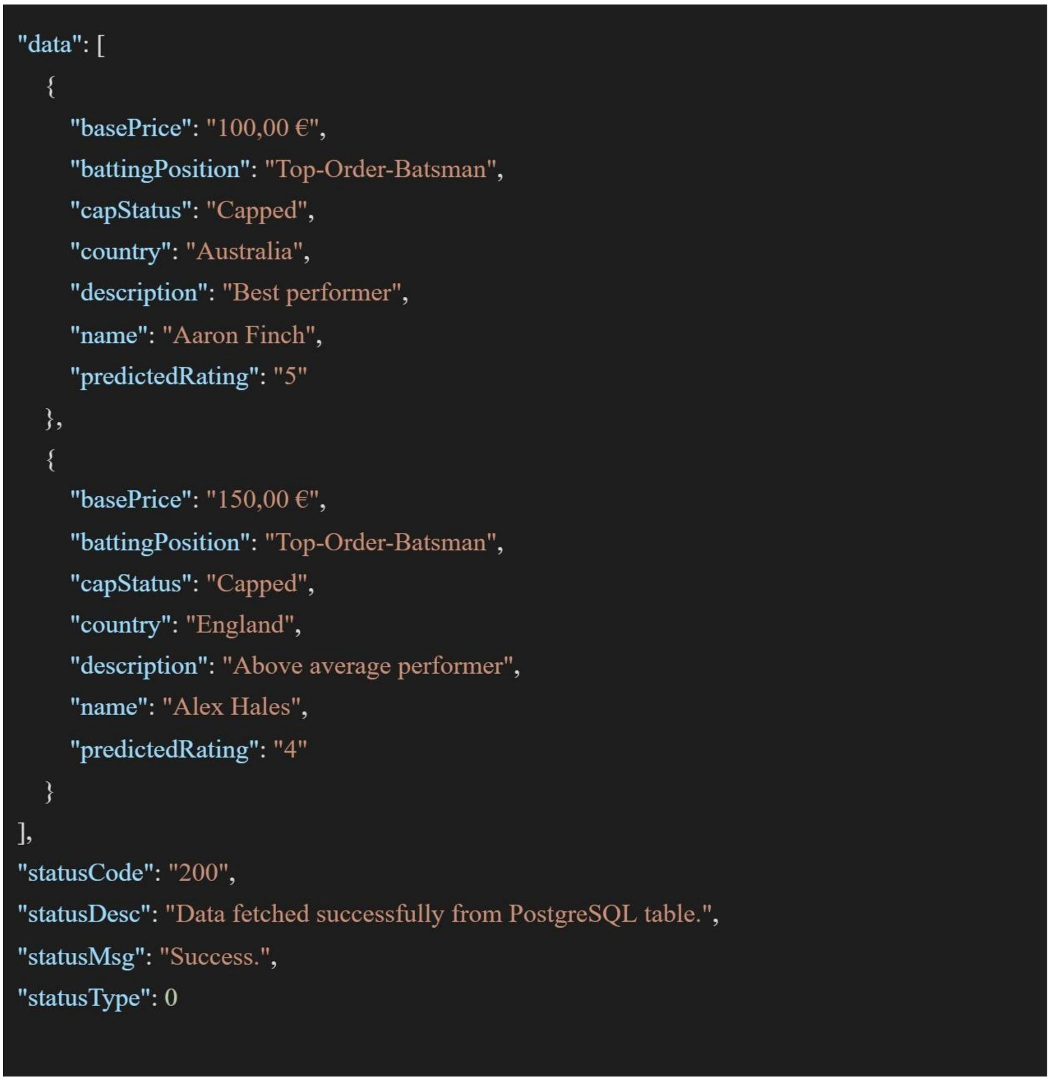
Request Structure for Wicket-Keeper Batsman:

```
{
```



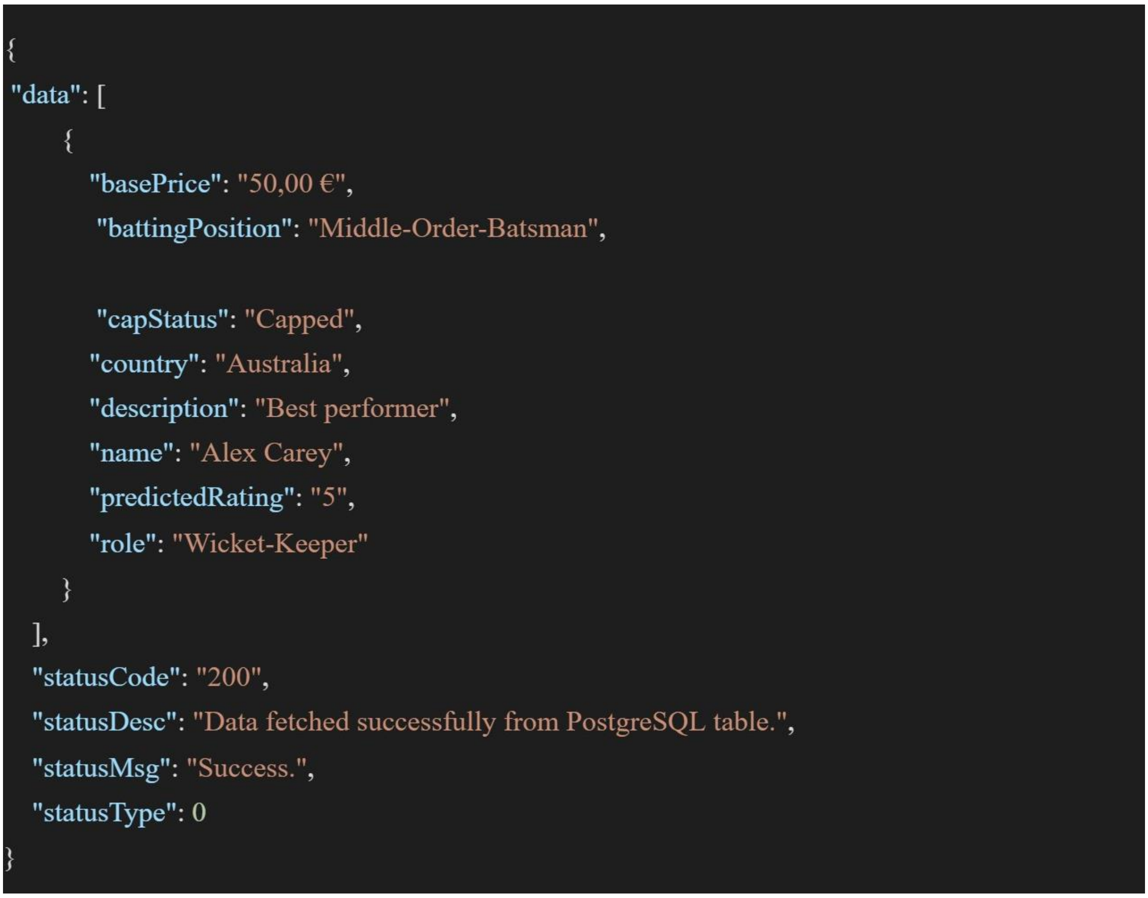
```
 "battingPosition" : "Middle-Order-Batsman",
 "role":"Wicket-Keeper",
 "minPrice" : "20",
 "maxPrice" : "150"
}
```

Success Request-Response: For all Batsman



```
"data": [
   {
      "basePrice": "100,00 €",
      "battingPosition": "Top-Order-Batsman",
      "capStatus": "Capped",
      "country": "Australia",
      "description": "Best performer",
      "name": "Aaron Finch",
      "predictedRating": "5"
   },
   {
      "basePrice": "150,00 €",
      "battingPosition": "Top-Order-Batsman",
      "capStatus": "Capped",
      "country": "England",
      "description": "Above average performer",
      "name": "Alex Hales",
      "predictedRating": "4"
   }
],
"statusCode": "200",
"statusDesc": "Data fetched successfully from PostgreSQL table.",
"statusMsg": "Success.",
"statusType": 0
```

Success Request - Response: For only Wicket-Keeper Batsman



```
{
 "data": [
      {
         "basePrice": "50,00 €",
          "battingPosition": "Middle-Order-Batsman",

          "capStatus": "Capped",
         "country": "Australia",
         "description": "Best performer",
         "name": "Alex Carey",
         "predictedRating": "5",
         "role": "Wicket-Keeper"
      }
   ],
   "statusCode": "200",
   "statusDesc": "Data fetched successfully from PostgreSQL table.",
   "statusMsg": "Success.",
   "statusType": 0
}
```

Validations covered:

1. The database connection should be active.
2. The user shall choose the player batting position and role as none to display only the performances of all batsmen.
3. The user shall choose the player batting position and role as wicket-keeper to display only the performances of the wicket keepers.

Actions in the System:

This API enables the user to fetch the prediction rating of the batsman based on the selection criteria of the batting position and the base price. Initially before fetching any players, this API checks whether the selected criteria exist in the table named tbl_batsmanauction on the Postgres database. If the selected criteria exist, this API returns the players in top wise prediction rating and along with the description. If the selection criteria do not exist, this API returns as “No players available in these criteria”.

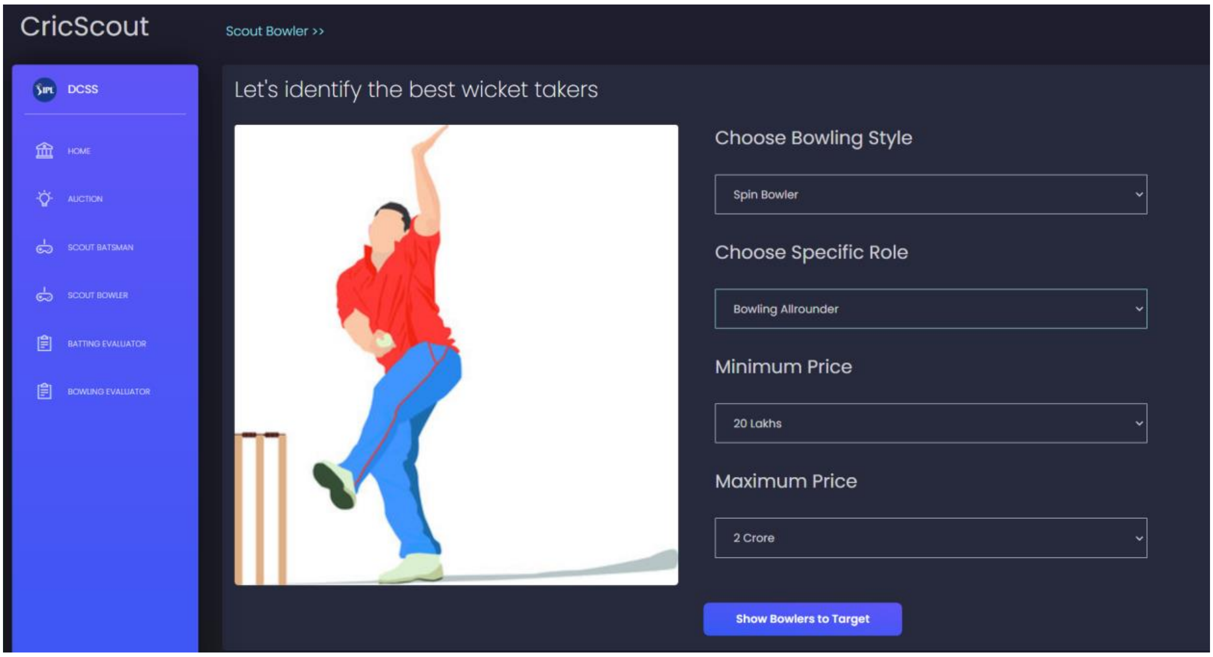
8.2.2 Scout Bowler

The Digital Cricket Scouting System facilitates the coaches or franchise owners to search the predicted performance of bowlers on the auction list. In addition, the user is provided with options to choose the bowling style, specific role, minimum, and maximum base price to make their query more detailed.

User Interface:

As shown in figure 15, users can see the form to choose the options for getting the predicted bowlers list by clicking the scout bowler tab from the navigation bar. The form comprises three options:

1. The user can select the bowling style of the player from the "choose bowling style dropdown," and the possibilities are a spin bowler and a fast bowler.
2. The user can choose the specific role of bowler from the "choose bowling role dropdown," and the options are either a none (no specific role) or bowling all-rounder.
3. The user can select the minimum base price and maximum base price options for additional filters.



Let's identify the best wicket takers

Choose Bowling Style: Spin Bowler

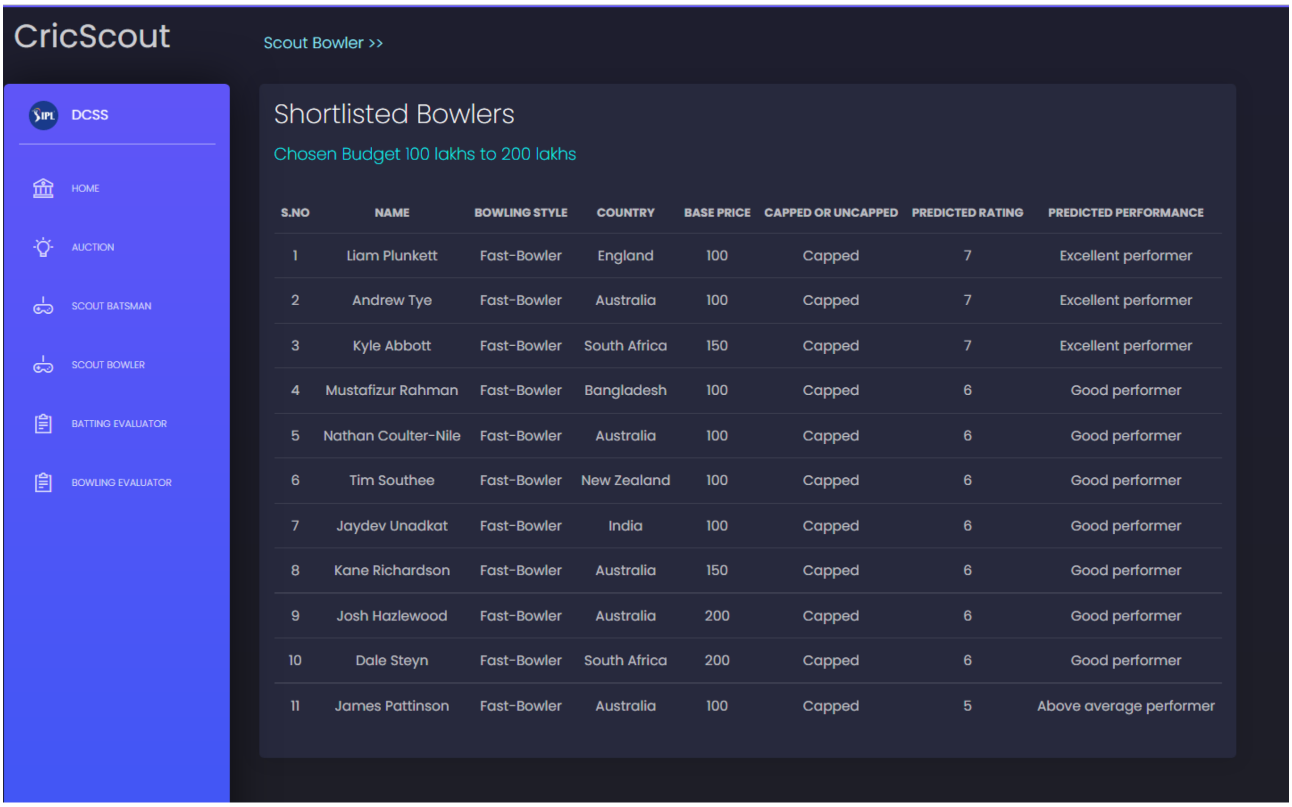
Choose Specific Role: Bowling Allrounder

Minimum Price: 20 Lakhs

Maximum Price: 2 Crore

Figure 15. UI for Scout Bowler

After filling the form, on clicking the "Show Bowlers to Target" button, a list of bowlers with their predicted performance sorted by rating and auction base price will be fetched using the predict bowler back-end web service. The user can view the fetched list as shown in figure 16.



Shortlisted Bowlers

Chosen Budget 100 lakhs to 200 lakhs

S.NO	NAME	BOWLING STYLE	COUNTRY	BASE PRICE	CAPPED OR UNCAPPED	PREDICTED RATING	PREDICTED PERFORMANCE
1	Liam Plunkett	Fast-Bowler	England	100	Capped	7	Excellent performer
2	Andrew Tye	Fast-Bowler	Australia	100	Capped	7	Excellent performer
3	Kyle Abbott	Fast-Bowler	South Africa	150	Capped	7	Excellent performer
4	Mustafizur Rahman	Fast-Bowler	Bangladesh	100	Capped	6	Good performer
5	Nathan Coulter-Nile	Fast-Bowler	Australia	100	Capped	6	Good performer
6	Tim Southee	Fast-Bowler	New Zealand	100	Capped	6	Good performer
7	Jaydev Unadkat	Fast-Bowler	India	100	Capped	6	Good performer
8	Kane Richardson	Fast-Bowler	Australia	150	Capped	6	Good performer
9	Josh Hazlewood	Fast-Bowler	Australia	200	Capped	6	Good performer
10	Dale Steyn	Fast-Bowler	South Africa	200	Capped	6	Good performer
11	James Pattinson	Fast-Bowler	Australia	100	Capped	5	Above average performer

Figure 16. UI Shortlisted Bowlers

Back-end Web Service:

This API is used to predict the performance of a Bowlers who are in the auction list whose statistics are managed by the administrator in the backend database. Bowlers have different styles such as fast-bowlers and spin-bowlers. Each bowler has their own individual base price. This API helps to predict and display the topwise performance of the players which are predicted to be in form in the upcoming matches. This API predicts the results based on the chosen role, bowling style, and base price.

Request Type: POST

Request URL: /predictBowler

Request Structure for Fast-Bowler:

```
{
  "role" : "Bowler",
 "bowlingStyle" : "Fast-Bowler",
  "minPrice" : "20",
  "maxPrice" : "150"
}
```

Request Structure for Spin-Bowler:

```
{
  "role" : "Bowler",
  "bowlingStyle" : "Spin-Bowler",
  "minPrice" : "20",
  "maxPrice" : "200"
}
```

Success Request-Response: For Fast-Bowler

```
{ "data": [ {
            "basePrice":"20",
            "bowlingStyle":"Fast-Bowler",
            "capStatus":"Uncapped",
            "country":"India",
            "name":"KartikTyagi",
             "predictedRating":"6"
        }, {
            "basePrice":"150",
            "bowlingStyle":"Fast-Bowler",
            "capStatus":"Capped",
            "country":"South Africa",
            "name":"Kyle Abbott",
            "predictedRating": "7" }],

    "statusCode":"200",
    "statusDesc":"Data fetched successfully from PostgreSQL table.",
    "statusMsg": "Success.",
    "statusType": 0}
```

Success Request-Response: For Spin-Bowler

```
{"data":[{
            "basePrice":"75",
            "bowlingStyle":"Spin-Bowler",
            "capStatus":"Capped",
            "country":"New Zealand",
            "name": "Ish Sodhi",
            "predictedRating": "6"   },{

            "basePrice":"150",
            "bowlingStyle":"Spin-Bowler",
             "capStatus":"Capped",
            "country":"Australia",
            "name":"Adam Zampa",
            "predictedRating": "8" } ],
   "statusCode":"200",
   "statusDesc": "Data fetched successfully from PostgreSQL table.",
   "statusMsg": "Success.",
   "statusType": 0}
```

Validations covered:

1. The database connection should be active.

2. The user shall choose the player's bowling style as Fast-Bowler or Spin-Bowler and role as Bowler to display the performances of bowlers.

3. The user shall choose the base price range to view the players within their budget.

Actions in the System:

This API enables the user to fetch the prediction rating of the bowlers based on the selection criteria of bowling style, role and the base price. Initially before fetching any players, this API checks whether the selected criteria exist in the table named tbl_bowlingauction on Postgres Database. If the selected criteria exist, this API returns the players in topwise prediction rating and along with the description. If the selection criteria do not exist, this API returns as “No players available in this criterion”.

All-Rounders Back-end web service:

This API is used to predict the performance of All-Rounders who are in the auction list whose statistics are managed by the administrator in the backend database. All-rounders can play different roles such as finisher-batsmen, fast-bowlers or spin-bowlers. Every all-round has their own individual base price. This API helps to predict and display the topwise performance of the allrounders who are predicted to be in-form in the upcoming matches. This API predicts the results based on the chosen role, bowling style for allround bowler or battingpostion for allround batsman, and base price.

Request Type: POST

Request URL for All-Rounder Batsman: /predictBatsman

Request URL for All-Rounder Bowler: /predictBowler

Request Structure for All-rounder Spin-Bowler:

```
{
   "role":"All-Rounder",
   "bowlingStyle":"Spin-Bowler",
   "minPrice":"40",
   "maxPrice" : "40"}
```

Success Request for All-rounder Spin-Bowler:

```
{
   "data": [{
         "basePrice": "40",
         "bowlingStyle": "Spin-Bowler",
         "capStatus": "Uncapped",
         "country": "India",
         "description": "Above average performer",
         "name": "Deepak Hooda",
         "predictedRating": "5"
      }],
   "statusCode": "200",
   "statusDesc": "Data fetched successfully from PostgreSQL table.",
   "statusMsg": "Success.",
   "statusType": 0
}
```

Request Structure for All-rounder Fast-Bowler:

```
{
   "role":"All-Rounder",
   "bowlingStyle":"Fast-Bowler",
   "minPrice":"40",
   "maxPrice":"40"
 }
```

Success Request for All-rounder Fast-Bowler:

```
{
   "data": [
      {
         "basePrice": "40",
         "bowlingStyle": "Fast-Bowler",
         "capStatus": "Uncapped",
         "country": "England",
         "description": "Good performer",
         "name": "James Fuller",
         "predictedRating": "6"
      },
      {
         "basePrice": "40",
         "bowlingStyle": "Fast-Bowler",
         "capStatus": "Uncapped",
        "country": "England",
        "description": "Above average performer",
        "name": "Benny Howell",
        "predictedRating": "5"
      } ],
   "statusCode": "200",
   "statusDesc": "Data fetched successfully from PostgreSQL table.",
   "statusMsg": "Success.",
   "statusType": 0
}
```

Request Structure for All-round Batsman:

```
{
   "battingPosition" : "Finisher-Batsman",
   "role":"All-Rounder",
   "minPrice" : "40",
   "maxPrice" : "40"
 }
```

Success Request for All-rounder Batsman:

```
{
   "data":[{
        "basePrice":"40",
        "battingPosition":"Finisher-Batsman",
        "capStatus":"Uncapped",
        "country":"India",
        "description":"Average performer",
        "name": "Deepak Hooda",
        "predictedRating": "3",
        "role": "All-Rounder"
      }, {
        "basePrice": "40",
        "battingPosition": "Finisher-Batsman",
        "capStatus": "Uncapped",
        "country": "England",
        "description": "Average performer",
        "name": "James Fuller",
        "predictedRating": "3",
        "role": "All-Rounder"
      } ],
   "statusCode": "200",
   "statusDesc": "Data fetched successfully from PostgreSQL table.",
   "statusMsg": "Success.",
   "statusType": 0
}
```

Validations covered:

1. The database connection should be active.
2. The user shall choose the player's bowling style as Fast-Bowler or Spin-Bowler and role as All-rounder to display the performances of all-rounder bowlers.
3. The user shall choose the player role as All-rounder and batting position as Finisher-Batsman to display the performances of all-rounder batsmen.
4. The user shall choose the base price range to view the players within their budget.

Actions in the System:

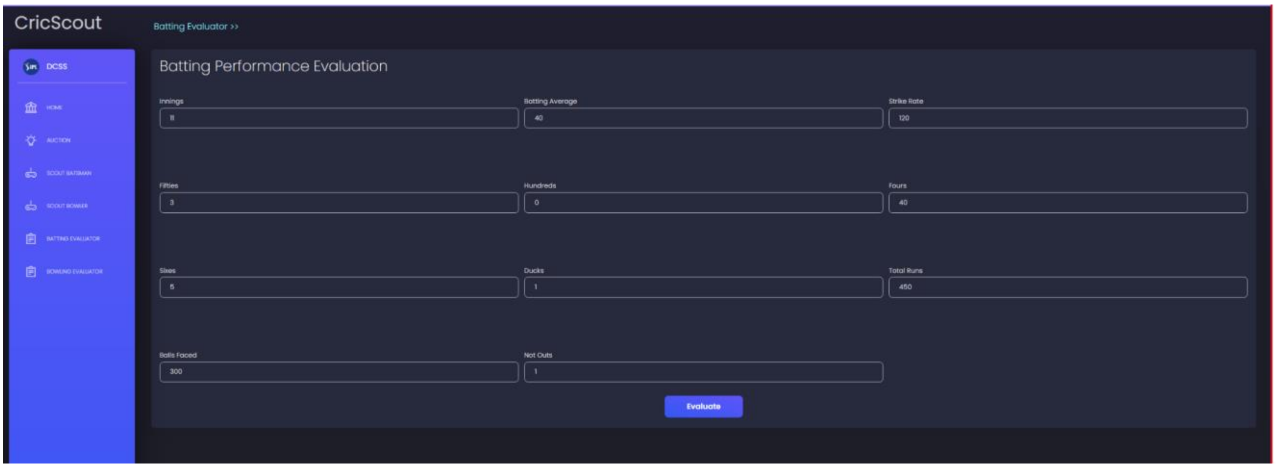
The API enables the user to fetch the prediction rating of the all-rounder on the selection criteria of bowling style, role and the base price for bowler and for batsman it based on batting position, role and base price. Initially before fetching any players, this API checks whether the selected player criteria exist in the table named tbl_bowlingauction for all-round bowlers and tables named tbl_batsmanauction for all-rounder batsmen on Postgres Database. If the selected criteria exist, this API returns the players in top wise prediction rating and along with the description. If the selection criteria do not exist, this API returns as “No players available in this criterion”.

8.2.3 Batting Evaluator

Apart from predicting the auction players list, the digital cricket scouting system can also evaluate any batsman's performance by providing their statistics as input. This functionality will assess the player statistics for all the batsman categories and offer better insight into the player's suitable batting position. This functionality is implemented to evaluate a relatively new player whose statistics are not available in the database.

User Interface:

By clicking the batting evaluator tab, users can see the form to provide the batsman's input statistics for getting the player's performance evaluation. As shown in figure 17, the form comprises eleven fields such as Innings, Batting Average, Strike Rate, the Fifties, Hundreds, Fours, Sixes, Ducks, Total Runs, Balls Faced, Not Outs. Each form field is afforded with respective helper text for better perception.

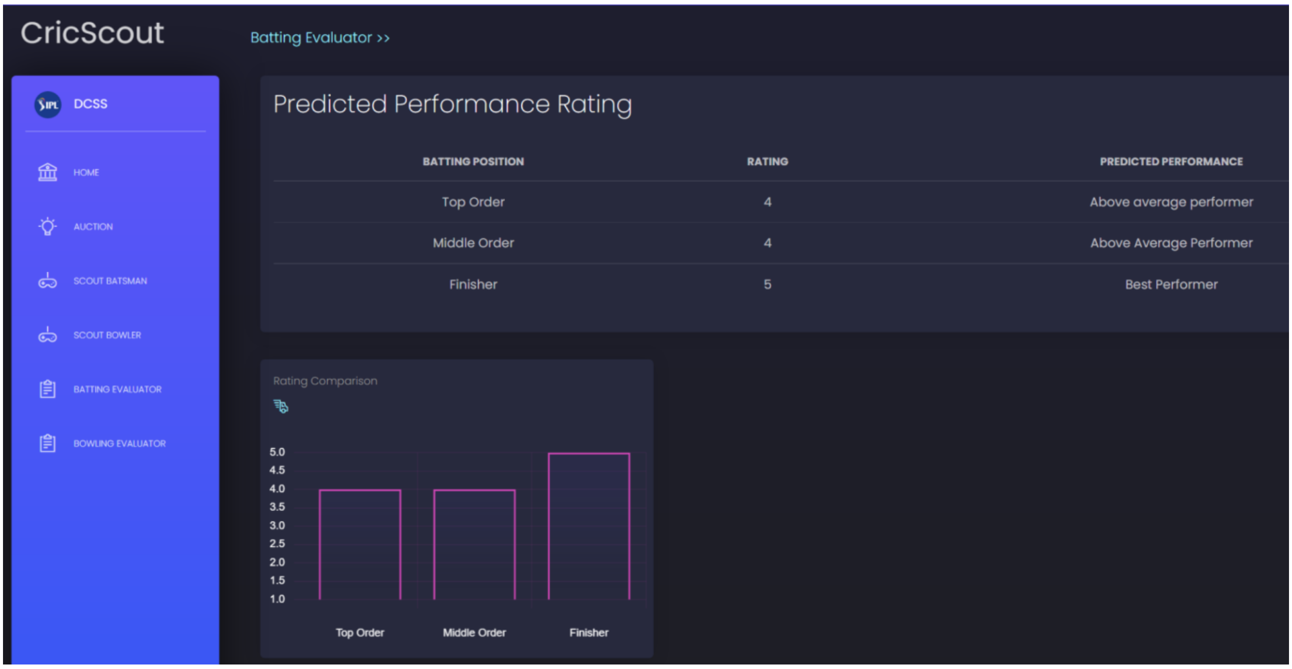


Batting Performance Evaluation

Innings 11	Batting Average 40	Strike Rate 120
Fifties 3	Hundreds 0	Fours 40
Sixes 5	Ducks 1	Total Runs 450
Balls Faced 300	Not Outs 1	

Figure 17. Batting Performance Evaluation UI

After filling the form, with one click of the "Evaluate" button, the predicted performance details for three different batsman categories will be fetched using the "fetch batsman ratings" back-end web service. The obtained details are illustrated in the table format and charts for better visualization, as shown in figure 18.



Predicted Performance Rating

BATTING POSITION	RATING	PREDICTED PERFORMANCE
Top Order	4	Above average performer
Middle Order	4	Above Average Performer
Finisher	5	Best Performer

Figure 18 Result of Batting performance evaluator

Back-end Web Service:

This API is used to predict the performance of a player whose statistics are entered in the user interface manually. This helps to predict the performance of the players who are not in the auction list or are not mainstream players.

Back-end Web Service:

Request type: POST

Request URL: /fetchRatings/batsman

Request Structure:

```
{
"innings" :  <int>,
"average" : <float>,
"strikeRate" : <float>,
"fifties" : <int>,
"hundreds": <int>,
"fours" : <int>,
"sixes" : <int>,
"ducks" : <int>,
"totalRuns" : <int>,
"ballsFaced" : <int>,
"notOuts" : <int>

}
```

Success Request Response:

```
{
   "ratings": {
      "finisher": {
         "Description": "Best performer",
         "rating": "5"
      },
      "middleOrder": {
         "Description": "Best performer",
         "rating": "5"
      },
      "topOrder": {
         "Description": "Best performer",
         "rating": "5"
      }},
   "role": "batsman"
}
```

Actions in the system:

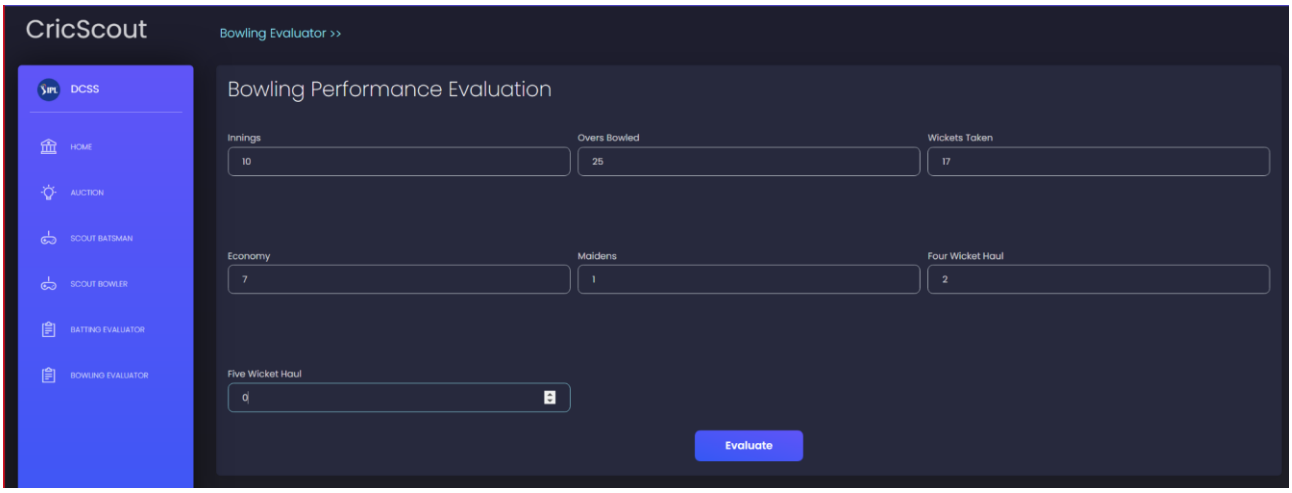
This API feeds the cricket player statistics into the machine learning model. The model predicts the performance score of the player based on these statistics and returns the performance score and performance description to the UI. The performance scores are predicted for three types of player roles for the same player.

8.2.4 Bowling Evaluator

Much like the batting evaluator option, the digital cricket scouting system also evaluates any bowler's performance by giving their statistics input. This functionality will assess the player statistics and offer better insight into the player's ability. This functionality is implemented to determine a new player whose match statistics are not available in the database.

User Interface:

By clicking the bowling evaluator tab, users can see the form to provide the bowler's input statistics for performing the player's performance evaluation. As shown in figure 19, the form comprises seven fields: Innings, Overs Bowled, Wickets Taken, Economic, Maidens, Four Wicket Haul, and Five Wicket Haul. Each form field is supported with individual helper text for better perception.

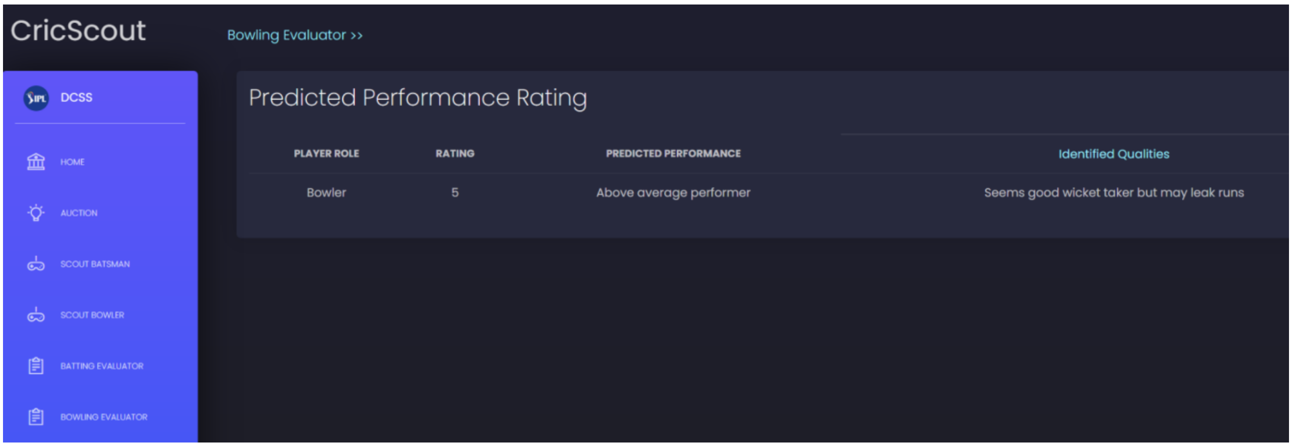


Bowling Performance Evaluation

Innings 10	Overs Bowled 25	Wickets Taken 17
Economy 7	Maidens 1	Four Wicket Haul 2
Five Wicket Haul 0		

Figure 19. Bowling Performance Evaluation UI

After filling the form, with one click of the "Evaluate" button, the predicted performance details will be fetched using the "fetch bowler ratings" back-end web service. The obtained details are illustrated in the table format and charts for better visualization, as shown in figure 20.



Predicted Performance Rating

PLAYER ROLE	RATING	PREDICTED PERFORMANCE	Identified Qualities
Bowler	5	Above average performer	Seems good wicket taker but may leak runs

Figure 20. Result of Bowling performance evaluator

Back-end Web Service:

This API is used to predict the performance of any bowler whose statistics are entered in the user interface manually by the user. This helps to predict the performance of the players who are not in the auction list or are not mainstream players.

Request type: POST

Request URL: /fetchRatings/bowler

Request Structure:

```
{
"innings":"10",
"ballsBowled" : "409",
"wickets" : "14",
"economy" : "10.0",
"maidens": "1",
"fourWicketHaul" : "0",
"fiveWicketHaul" : "1"

}
```

Success Request Response:

```
{
   "ratings": {
      "Description": "Above average performer",
      "rating": "5" },
   "role": "bowler"
}
```

Actions in the system:

This API feeds the cricket player statistics into the machine learning model and calculates the performance rating using the bowler formula that was created using AHP tool. The model predicts the performance score of the bowler based on these statistics and returns the rating and performance description to the UI.

8.2.5 Admin API - Training the batsman model

This API is used to train the machine learning model of the batsman using a train dataset of players whose role is batsman. This helps to train the model for predicting the performance rating of the players.

Back-end Web Service:

Request type: GET

Request URL: /trainbatsman

Actions in the system:

This API trains the machine learning model with batsman player statistics. The performance of the player is calculated using the Batsman formula that was created using the AHP tool. Each batting position has its own formula, and the players' statistics are fed to the formula and calculated by the performance rating. The model predicts the performance score of the player based on the statistics and batting position. The machine learning algorithm that is used to train the model is Random Forest.

8.2.6 Admin API - Training the bowler model

This API is used to train the machine learning model of the bowler using a train dataset of players whose role is bowler. This helps to train the model for predicting the performance rating of the players.

Back-end Web Service:

Request type: GET **Request URL:** /trainbowler

Actions in the system:

This API trains the machine learning model with bowler player’s statistics. The performance of the player is calculated using the Bowling formula that was created using the AHP tool. These players' statistics are fed to the formula and calculated the performance rating. The model predicts the performance score of the player based on the statistics and bowling role. The machine learning algorithm that is used to train the model is Random Forest.

9. Deployment

For the deployment of the Digital Cricket scouting System, Firebase has been used. Firebase is a Google owned software service that offers many tools like hosting, developing environment, analytics, authentication, performance, testlab, crashlytics[38].

For the deployment of the application, below mentioned steps were followed[39]:

- First of all, the firebase needs to be installed in the project which can be done via NPM command “npm install firebase-tools -g”
- Now the firebase account should be logged in and initialized with the below command. “firebase init”
- After initializing the firebase, it asked a bunch of questions related to the project hosting like the name of the project that needs to be hosted, the location of the project files in the directory, and overwriting of some files. After answering the question the firebase will generate a “firebase.json” file that consists of all the configurations of the firebase hosting.
- Once the “firebase.json” file is generated the project can be deployed via the command “firebase deploy”
- And after running the deploy command, teh firebase returns the unique URL where the app is deployed and for the case of the Digital Cricket Scouting System it was deployed in *https://digital-cricket-scout.web.app/*

10. Evaluation

10.1 Machine Learning Model Evaluation

- After creating the target variable using AHP formulas, the reliability of the variable was verified manually.
- The entire trained dataset which was used for the training of the model was processed via AHP (Analytical Hierarchy Process). For each playing position of a player a separate formula was given and it was trained differently to achieve the result for a target variable.
- The model was trained for all the three algorithms with different split ratios and results were compared as shown in the table 13. The best case was achieved by the split ratio of 80:20 (Training:Testing) for Random forest with 99% performance output(Accuracy, Precision, Recall, F1 score). Theoretically this result also supports the fact that random forest is highly accurate in this case because it uses a forest of decision trees, which ultimately helps the algorithm to make more accurate decisions as comparison of decision trees which uses a single tree to estimate the results.

Table 13. Algorithm performance comparison

Algorithm Name	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Naive Bayes	81%	81.4%	81%	81%
Random Forest	99.12%	99.12%	99.12%	99.12%
Decision Tree	98.44%	98.44%	98.44%	98.43

- Figure 21 is the plot which represents the learning curve of the model for the Random Forest algorithm. The below plot represents how well the trained model is performed when going through the validation data. Initially the training score is one, when the model was trained and when the model went through the validation data it started to predict 85 percent of the case correctly and later with the increase in the validation data, the learning curve increased gradually representing the accuracy of the model.

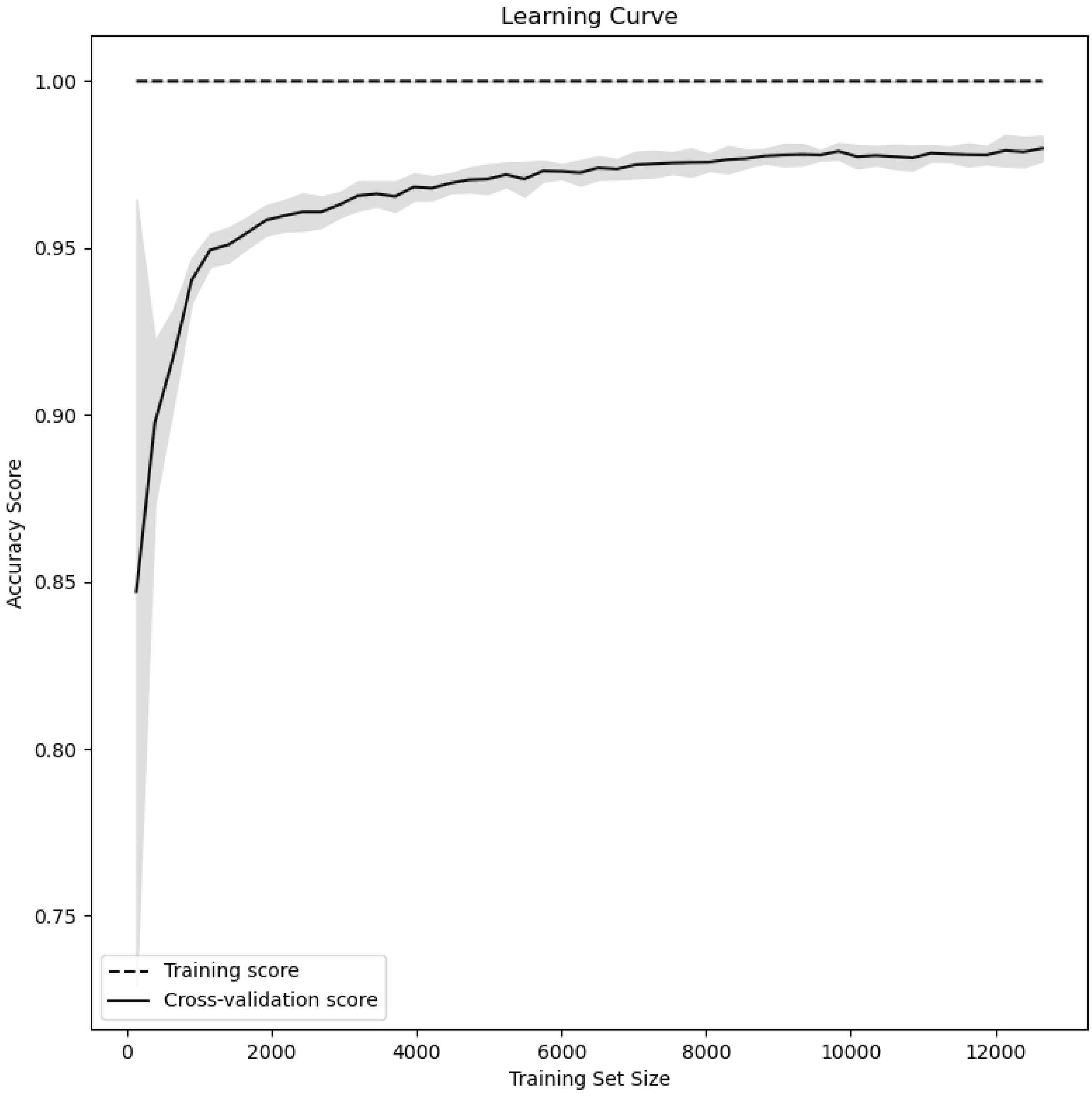


Figure 21. Learning curve of Random Forest

- To make sure the model is working fine, manually evaluation was also done. For manual evaluation, the recent data of the players playing in 2020 IPL was collected manually and was fed to the system and the predicted results were highly accurate.

10.2 Front-end Evaluation

- For the user Interface evaluation, different types of error, success and warning message are displayed in the appropriate place in the screen.
- For better and clear visuals, the results of the players fetched from the back end are shown in the tabular format.

- Interactive and responsive User Interface for better understanding of the user

10.3. Back-end Evaluation

- All of the Back-end APIs are independent of each other, so they were tested independently after each stage of development
- Pycharm, the IDE used for writing back-end APIs, default debugger was also used to check the stability of the APIs
- External tools, such as Postman application is also used to check whether the APIs are responding properly or not in according to the UI requests
- Try catch blocks were frequently used to avoid the crash of the APIs.

11. System Limitation

- This model only predicts the best players according to each playing position; however, it might also be possible that the best-predicted player might not perform well because it tells the best players on the basis of the past performance and the randomness included in the game might overrun that prediction.
- Is not able to identify the team coordination ability of a player.

12. Conclusion

Digital Scouting System is a machine learning model which can be used as a web app by end-users, helps the team owners or coaches to identify the best players according to the different playing positions for the game of cricket. Its accuracy is also very high. This model also filters the best players according to the price tags which helps team owners to manage their team budget in a way more effective manner. This model helps the team to perform well by assembling a good performing team and to keep the team in the competition of winning trophies in today's world of highly competitive sports.

The ideology used in this application for predicting players performance is highly accurate as it is a combination of machine learning algorithms and the mathematical formula derived through the AHP process. This concept can prove to be revolutionary and has the potential to be applied to any other sport, the only thing that needs to be changed are the parameters (players performance variable) related to that specific sport and then basically it will work the same.

13. Future Scope

- As of now, the application is only limited to Cricket, but in future, it can be extended to scout best players in other sports also.
- Can be partnered with professional sports teams/clubs to get clean and rich data to train the model and achieve better and accurate results.
- The same approach can also be implemented using deep learning algorithms like Artificial Neural Network or Conventional Neural network and the results can be compared.

References

[1]. A. I. Anik, S. Yeaser, A. G. M. I. Hossain and A. Chakrabarty, "Player's Performance Prediction in ODI Cricket Using Machine Learning Algorithms," 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), 2018, pp. 500-505, doi: 10.1109/CEEICT.2018.8628118.).

[2]. (Davis, Jack & Perera, Gamage & Swartz, Tim. (2015). Player evaluation in Twenty20 cricket. Journal of Sports Analytics. 1. 19-31. 10.3233/JSA-150002.)

[3]. Michonski, Z., By, & Michonski, Z. (2020, December 3). 3 popular sports growing across the globe. EF Tours Blog. https://blog.eftours.com/inspiration/education/sports-that-are-globally-growing-beyond-their-common-audiences.

[4]. S. Muthuswamy and S. S. Lam, "Bowler Performance Prediction for One-day International Cricket Using Neural Networks," in Industrial Engineering Research Conference, 2008.

[5]. Mr. Mujamil Dakhani 1, Umme Habiba Maginmani2, Predicting Accuracy of Players in the Cricket Using Machine Learning, International Research Journal of Engineering and Technology (IRJET), May 2020

[6]. S. R. Iyer and R. Sharda, "Prediction of athletes performance using neural networks: An application in cricket team selection," Expert Systems with Applications, vol. 36, pp. 5510-5522, April 2009.

[7]. H. H. Lemmer, "The combined bowling rate as a measure of bowling performance in cricket," South African Journal for Research in Sport, Physical Education and Recreation, vol. 24, no. 2, pp. 37-44, January 2002.

[8]. Lewis, A. J. (2005). Towards fairer measures of player performance in one-day cricket. Journal of the Operational Research Society, 56, pp.804-815.

[9]. World leaders in sports data. Opta Sports. (n.d.). https://www.optasports.com/.

[10]. Football Professional Videos and Data Platform. Wyscout. (n.d.). https://wyscout.com/.

[11]. Brownlee, J. (2020, August 19). Difference Between Algorithm and Model in Machine Learning. Machine Learning Mastery. https://machinelearningmastery.com/difference-between-algorithm-and-model-in-machine-learning.

[12]. (Atakan KantarciAtakan is an industry analyst of AIMultiple. He has a background in consulting at Deloitte, & *, N. (2021, April 4). Top 18 Web Scraper / Crawler Applications & Use Cases in 2021. AIMultiple. https://research.aimultiple.com/web-scraping-applications/).

[13]. Real Python. (2021, June 17). Pythonic Data Cleaning with Pandas and NumPy. Real Python. https://realpython.com/python-data-cleaning-numpy-pandas/.

[14]. Saaty, T. L. (2011, January 13). How to make a decision: The analytic hierarchy process. European Joournal of Operational Research. https://www.sciencedirect.com/science/article/pii/037722179090057I.

[15]. Cheng, E. W. L., & Li, H. (2001, September 1). Analytic hierarchy process: an approach to determine measures for business performance. Measuring Business Excellence. https://www.emerald.com/insight/content/doi/10.1108/EUM0000000005864/full/html

[16]. Saaty, T. L. (2011, January 13). How to make a decision: The analytic hierarchy process. European Journal of Operational Research. https://www.sciencedirect.com/science/article/pii/037722179090057I.

[17]. (1999) The AHP, Pairwise Comparisons. In: Lootsma F.A. (eds) Multi-Criteria Decision Analysis via Ratio and Difference Judgement. Applied Optimization, vol 29. Springer, Boston, MA. https://doi.org/10.1007/978-0-585-28008-0_3

[18]. Taherdoost, Hamed, Decision Making Using the Analytic Hierarchy Process (AHP); A Step by Step Approach (2017). International Journal of Economics and Management Systems, Vol. 2, 2017, Available at SSRN: https://ssrn.com/abstract=3224206

[19].https://www.researchgate.net/publication/316736276_Analytic_Hierarchy_Process_AHP_for_business_site_selection

[20]. Saaty, T. L. (2011, January 13). How to make a decision: The analytic hierarchy process.European Journal of Operational Research. https://www.sciencedirect.com/science/article/pii/037722179090057I.

[21]. A criterion for comparing and selecting batsmen in limited overs cricket. Taylor & Francis. (n.d.). https://orsociety.tandfonline.com/doi/abs/10.1057/palgrave.jors.2601800?journalCode=tjor20#.YNNO82gzaM8.

[22]. Ahmed, F., Deb, K., & Jindal, A. (2012, September 3). Multi-objective optimization and decision making approaches to cricket team selection. Applied Soft Computing. https://www.sciencedirect.com/science/article/pii/S1568494612003584#bib0070.

[23]. pandas. (n.d.). https://pandas.pydata.org/.

[24]. Installing scikit-learn¶. learn. (n.d.). https://scikit-learn.org/stable/install.html.

[25]. NumPy. (n.d.). https://numpy.org/.

[26]. 1.9. Naive Bayes¶. scikit. (n.d.). https://scikit-learn.org/stable/modules/naive_bayes.html.

[27]. Decision Tree Algorithm, Explained. KDnuggets. (n.d.). https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

[28]. 1.10. Decision Trees¶. scikit. (n.d.). https://scikit-learn.org/stable/modules/tree.html.

[29].Machine Learning Random Forest Algorithm - Javatpoint. www.javatpoint.com. (n.d.). https://www.javatpoint.com/machine-learning-random-forest-algorithm

[30].sklearn.ensemble.RandomForestClassifier.scikit.(n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

[31]. Welcome to Flask. Welcome to Flask - Flask 0.10.1 documentation. (n.d.). https://flask-doc.readthedocs.io/en/latest/.

[32].PostgreSQL Python. PostgreSQL Tutorial. (n.d.). https://www.postgresqltutorial.com/postgresql-python/.

[33]. React – A JavaScript library for building user interfaces. – A JavaScript library for building user interfaces. (n.d.). https://reactjs.org/.

[34]. axios. npm. (n.d.). https://www.npmjs.com/package/axios.

[35]. DigitalOcean. (2021, June 11). How To Build Forms in React with Reactstrap. DigitalOcean. https://www.digitalocean.com/community/tutorials/react-fancy-forms-reactstrap.

[36]. imbalanced-learn. PyPI. (n.d.). https://pypi.org/project/imbalanced-learn/.

[37]. learn. scikit. (n.d.). https://scikit-learn.org/stable/.

[38]. Rosencrance, L. (2019, April 25). What is Google Firebase? - Definition from WhatIs.com. SearchMobileComputing.
https://searchmobilecomputing.techtarget.com/definition/Google-Firebase.

[39]. Richards, J. (2019, September 7). How to Deploy a React App with Firebase Hosting. Medium. https://medium.com/swlh/how-to-deploy-a-react-app-with-firebase-hosting-98063c5bf425.

[40]. Thiraviam, B. (n.d.). ‘How to calculate Bowling Average, Strike Rate & Economy Rate in Cricket?’, CricIndeed. Available at: https://www.cricindeed.com/how-to-calculate-bowling-average-strike-rate-economy-rate-in-cricket/ [Accessed 15 June 2021]

[41]. Katewa, S. (n.d.). ‘Cricket Batting Average: How Does it Work & Should You Care?’. CricketMastery. Available at: https://cricketmastery.com/cricket-batting-average/ [Accessed 15 June 2021]

Appendix