**“Data Analysis and Recommender System in IPL”**

**Mini Project**

**In**

**Sixth Semester**

**By**

|  |  |
| --- | --- |
| **1MS16IS016** | **Atul R H** |
| **1MS16IS028** | **Hitesh M** |

**Under the guidance of**

**Dr.Vijaya Kumar B P**

Professor and Head

Dept. of ISE, RIT

****

**DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING**

**RAMAIAH INSTITUTE OF TECHNOLOGY**

**(AUTONOMOUS INSTITUTE AFFILIATED TO VTU)**

**VIDYA SOUDHA**

**M. S. RAMAIAH NAGAR, M. S. R. I. T. POST, BANGALORE – 560054**

**2018-2019**

RAMAIAH INSTITUTE OF TECHNOLOGY

(Autonomous Institute Affiliated to VTU)

VIDYA SOUDHA

M. S. Ramaiah Nagar, M. S. R. I. T. Post, Bangalore – 560054

## DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

****

**CERTIFICATE**

This is to certify that the project work entitled **“Data Analysis and Recommender System in IPL”** is a bonafide work carried out by **Atul R H** bearing **USN: 1MS16IS016** and **Hitesh M** bearing **USN: 1MS16IS028** in partial fulfilment of requirements of Mini Project course of Sixth Semester B.E. It is certified that all correction/suggestions indicated for internal assessment has been incorporated in the report. The project has been approved as it satisfies the academic requirements in respect of project work prescribed by the above said course.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Signature of the Guide Signature of the HOD

**Dr. Vijaya Kumar B P Dr. Vijaya Kumar B P**

Professor and Head Professor and Head Dept. of ISE, RIT, Dept. of ISE, RIT Bangalore-54 Bangalore-54

##### Other Examiners

**Name of the Examiners: Signature**

1.

2.

**DECLARATION**

I Student of sixth semester BE, Dept. of Information Science and Engineering, Ramaiah Institute of Technology, Bangalore, hereby declare that the project entitled “Data Analysis and Recommender System in IPL**”,** thesis completed and written by me under the guidance of **Dr. Vijaya Kumar B P,** Dept. of Information Science and Engineering, Ramaiah Institute of Technology, Bangalore.

Place: Bangalore

Date:

Atul R H 1MS16IS016

Hitesh M 1MS16IS028

**ACKNOWLEDGEMENT**

First and foremost, our utmost gratitude to *Dr. Vijaya Kumar B P* dept of ISE, MSRIT whose sincerity and thoughtfulness we will never forget. He has been our inspiration as we overcame all the obstacles in the completion of this project work. He had kind concern and consideration regarding project work and we would like to thank him for his continuous support and immeasurable patience.

We would like to thank our beloved principal Dr. N.V.R Naidu for his support and encouragement.

This work would not have been possible without the guidance and help of several individuals who in one way or another contributed their valuable assistance in preparation and completion of this study.

We would like to express sincere thanks to all the teaching and non-teaching faculty of ISE Department and our dear friends who helped in preparing the Report.

# ABSTRACT

In sports, recruitment and selection strategies against other teams are prone to human errors due to biases, incomplete data, insufficient analysis and other such oversights. In order to mitigate such issues, implementing a data driven approach is important as it would result in more objective and efficient selection processes.

This project utilizes 3 datasets consisting of various features of the Indian Premier League(IPL) including ball by ball data, the outcome of every match etc. The objective of the project is to process this data so as to uncover patterns and trends to draw helpful conclusions like finding the best batsmen against particular styles of bowling or specific bowlers. Such information is invaluable for teams to strategize against other teams and players. This project also implements nearest neighbour search using K-D Trees to identify outstanding players and help discover and recommend players similar to them. These outcomes can be presented to sports organizations who can take appropriate measures to ensure that selection of players is more optimal. Successful implementation of such technology could help provide teams with competitive benefits over others resulting in better performances overall.

#### CONTENTS

***Declaration i***

***Acknowledgements ii***

***Abstract iii***

**Page No**

1 **INTRODUCTION**

* 1. Motivation 1
  2. Scope 1
  3. Objectives 2
  4. Proposed Model 2
  5. Organization of Report 3

1. **LITERATURE SURVEY**

**2.1** Introduction 4

**2.2**  Related Works with citation of References 4

**2.3** Conclusion of Survey 5

1. **SYSTEM ANALYSIS AND DESIGN**

**3.1** Software Process Models 6

**3.2**  Roles and Responsibilities 6

**3.3** Product Overview 7

**3.4** External Interface Requirements 7

**3.5**  Functional Requirements 9

**3.6** Non**-**Functional Requirements 9

**3.7** Performance Requirements 11

**3.8**  Risk Identification and Assessment 11

1. **MODELLING AND IMPLEMENTATION**

**4.1** Introduction 13

**4.2** Architecture Diagram 13

**4.3** Sequence Diagram 14

**4.4** Use Case Diagram 15

**4.5** Tools Introduction 16

**4.6** Technology Introduction 16

**4.7** Explanation and Implementation 17

Algorithms

**4.8** Pseudo Code 20

1. **TESTING, RESULTS AND DISCUSSION**
   1. Testing 23
   2. Test Cases 23
   3. Results and Discussion 24
2. **CONCLUSION AND FUTURE WORK** 31
3. **REFERENCES** 32

***List of Figures***

Fig. 4.2.1 Architecture Design 14

Fig. 4.3.1 Sequence Diagram 15

Fig. 4.4.1 Use Case Diagram 16

Fig. 5.3.1 Average runs v/s Batsmen 24

Fig. 5.3.2 Strike rate v/s Batsmen 24

Fig. 5.3.3 Innings progression: Strike rate v/s runs 25

Fig. 5.3.4 Innings progression: Strike rate v/s balls 26

Fig. 5.3.5 Top Batsmen v/s Right arm med bowlers 27

***List of Tables***

Table 3.2 Roles and Responsibilities 6

Table 3.4.1 Software Requirements 8

Table 3.8 Risk Identification 11

**Chapter 1**

**INTRODUCTION**

**1.1 Motivation**

A large percent of human beings inherently identify with sporting events. We may even draw a parallel between fans of a sporting club to individuals of a nation; just as citizens of a nation share a common identity, similar traditions and culture, in sports, individuals share a common identity with fellow supporters and similar tradition and culture that pertains to the team they support and the sport they follow. Sports play a huge role of a large number of people’s lives around us which is why it’s a global, multi-billion dollar industry. Technology has revolutionized sport to a large degree with devices like smart helmets to goal-line technology. Our motivation is to attempt to help integrate the power of data with sports. This involves analyzing vasts amounts of data to gain helpful insights and conclusions that may help modernize the way in which sports are played and perceived. Given the fact that sports is a domain where the potential of data and machine learning isn’t completely explored, as ardent sports fans and data enthusiasts, we believe it’s our duty to fill the gaps and produce work that can help further the advent of helpful technology into the field of sports. In status quo, data driven approaches are implemented by only a select few leagues within a select few sports such as the NBA, Major League Baseball(MLB) etc. Machine learning is even more sparsely implemented and is often only used to create prediction models that determine outcomes of matches. This project attempts to study the work done on data in other sports and attempt to apply it to cricket given its widespread following in India. The secondary focus of this project is to incorporate machine learning into the sport and formulate a recommender system that may be helpful at almost all points of time.

**1.2 Scope**

The scope of this project includes the following:

* Analyzing the dataset we get important information about team and player performances elucidating weaknesses to exploit or work on and strengths to harness or strategize against.
* Recommending alternative players that can emulate players that exhibit optimal performance levels. This simplifies processes like acquiring players into teams via auctions and replacing injured/unavailable players.
* The potential to improve on this model to make predictions and identify trends more accurately depends on availability of more specific data. In future, when there is more data on an increased number of factors such as speed of bowling, degree of spin etc., we can formulate far more accurate and targeted models.
* As datasets grow and provide us with more data from different avenues such as Ranji Trophy and other intra-national level cricketing tournaments, we will be able to harness this model to better optimize scouting and recruitment strategies.

**1.3 Objectives**

* To analyze,sanitize and/or combine datasets to a degree to which we are able to directly apply datasets to machine learning models optimally.
* To extrapolate as much relevant information as possible from the given dataset based on requirements.
* To identify players who exhibit high levels of performance and recommend other players who are able to emulate similar levels of performance.

**1.4 Proposed Model**

Our dataset contains ball by ball data ranging from IPL seasons 2008-2018. Our model uses Python Libraries like Pandas and NumPy to convert this raw data into meaningful projections of data. This data aggregation is important as the raw dataset containing ball by ball information cannot directly be used. At this stage, we can extrapolate useful information from the dataset based on requirements. Our model then incorporates an unsupervised machine learning algorithm in order to facilitate a recommender system that can recommend players with similar levels of performance. In this project, our recommendation query is essentially a search problem. Thus, we use K-Dimensional trees to reorganize the dataset into a tree structure in order to reduce the complexity of the search problem instead of directly using K-nearest neighbour despite easier implementation as KD trees project more accurate results. The KD tree is traversed until the datapoints about the player closest to the searched player is found and corresponding values are returned.

Using this model, we are able to achieve the following **deliverables:**

* Analysis of various important player and team statistics.
* Easier visualization of data.
* A recommendation of similar performing players to a given player.
* A recommendation for which batsman to field to play against a particular bowler.

**1.5 Organisation of Report**

In order to explain the developed system, the following sections are covered :

* **Literature Review** describes the study of the existing systems and techniques taken into account prior to development of the proposed system.
* **System Analysis and Design** provides a detailed walkthrough of the software engineering methodology adopted to implement the model, an overview of the system and the modules incorporated into the system.
* **Modelling and Implementation** provides a deeper insight into the working of the model. The various modules and their interactions are depicted using relevant descriptive diagrams.
* **Testing** the model to ensure bug/error free model along with the **Results** obtained. **Discussion** then provides detailed analysis on quality assurance measures.
* **Conclusion** about the Results obtained after successfully running the model and **Future Scope** of the model is highlighted.

**Chapter 2**

**LITERATURE SURVEY**

**2.1 Introduction**

The ubiquity of sports coupled with the huge market it offers has led to numerous papers published on sports. There is a trend of individuals and organizations publishing research on predictor models that claim to accurately predict large scale events such as World Cups. However, for the purposes of this project, we primarily focus on papers that help arrive at attributes that help us identify key players as well as papers which attempt to discern similarities between players, which is comparatively less explored.

**2.2 Related Works with the citation of the References**

A study to identify optimal attributes that impact the result of a cricket match[1] was carried out. The study documents the use of Extra Tree algorithm, Naive Bayes classifier and Support Vector Machine(SVM) algorithm, with the SVM algorithm being the most accurate, discerning that a combination of the following attributes were contribute the most to winning a cricket match: high individual wickets, number of bowled deliveries, number of thirties, total wickets, wickets in power-play, runs in death overs, dots in middle overs, number of fours, singles in middleovers. A research paper on Cricket Sabermetrics and Data Mining analysis for cricket[2] whose main aim was to explore the statistical aspect of cricket in a selected domain and find key performance metrics that contribute towards the outcome of matches also provided a visualization feature, which shows how some specific features affect the end result of a match. A paper[3] by A J Lewis explores identifying better metrics of evaluation of cricketers than current statistical metrics such as economy rate, strike rate etc. as they do not consider the stage of the innings a player is performing in which, needs to be reflected for the metric to be fair according to him.

N P Kurade’s research paper[4] for selection and recommendation of players to build a sports team involves ranking players and using an Active Window Filtration system to recommend players to build a team. Another research paper[5] uses binary integer programming to create optimal sequences of teams in fantasy sports leagues, particularly cricket.

**2.3 Conclusion of Survey**

Using the existing literature on creating teams for fantasy league, we are able to understand how players are viewed on a relative scale to each other. Metrics like base score, impact score, milestone scored etc. are used to capture the context of the data about players. However this demarcation strategy could still be insufficient as details such as match conditions like weather, nature of the pitch etc. could also impact the performance of players but are left out. The paper by A J Lewis gives us insight into the fact that the context of certain cricket games are important to keep in mind while determining the worth of players based on their performance. For example, this could mean that statistics like the strike rate of players in matches could depend less on the merit of the player and more on factors that contribute to the overall nature of the game, like chasing a small target or the stage of the game that they are playing in, power-play or middle stages of the innings etc. These papers also give us insight on how to visualize data optimally to arrive at far reaching inferences. With this information, we can better create a model that optimally selects and recommends players that are closely related to the key players we identify using data visualization models. The existing literature covering identification of what traits or attributes of a cricket player impacts the game gives us information based on which we can better select or better model identification of key players using the data that we possess. We can model a system to identify players who stand out in those specific traits that are considered most impactful. We can also use these traits as the base parameters for the K-D Tree and nearest neighbour search algorithm to identify similar players based on these most impactful traits to improve our model. As discussed earlier, literature on player recommender models is lacking, therefore, we conclude our survey by focusing on similar studies to improve our model instead of trying to resolve problems with existing models of recommender systems for IPL.

**Chapter 3**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 Software process Models**

The chosen software process model for this project is Agile Methodology. This methodology is advantageous to us as it allows for flexibility for change and for improvement regardless of inconsistent requirements. It has a lower cost of working specifically for this project because we are able to deliver multiple returns specific to our requirements. This method encourages open communication among team members, and clients at every step of software development. This ensures changes can be made quicker and throughout the development process by having consistent evaluations to assess the product with the expected outcomes requested and reducing overall evaluation time. It keeps each project transparent by having regular consistent meetings that allow everyone involved to access the project data and progress.

**3.2 Roles and Responsibilities**

|  |  |
| --- | --- |
| Literature Survey | Atul and Hitesh |
| Domain Analysis | Atul and Hitesh |
| Resource Planning | Hitesh |
| Synopsis | Atul |
| Project Planning | Atul and Hitesh |
| Design of system architecture | Hitesh |
| Coding | Atul and Hitesh |
| Testing | Atul and Hitesh |
| Report | Atul and Hitesh |

**Table 3.2 Roles and Responsibilities**

**3.3 Product Overview**

The project basically emphasises on finding several inferences about different players, teams etc. in the IPL with the help of various data visualization tools. This project also includes a functional recommender system which on searching for a player, returns the closest possible player(s) with attributes as similar as possible to the original player.

First, ball by ball data from all editions of the IPL between 2008-2018 is studied. This raw is processed to a more palatable form. Using this new modified dataset, we are able to make several important inferences from the data regarding certain key features of the IPL. For example, from this data, we are able to discern the best batsman vs spin bowling in the IPL. Such information is crucial to sporting teams and can provide a marginal benefit which is often crucial in the field of sports.

This project has the additional functionality; we can use the processed dataset along with unsupervised machine learning algorithms to create a recommender system. The processed data is learned into a K-Dimensional tree format. We then use Nearest Neighbour algorithm to determine the nearest neighbours to a particular datapoint. When we search for a player, the algorithm traverses the KD Tree and until it finds a player with the closest possible attributes as the searched player and returns the players name and statistics. Thus, we are able to analyze and make inferences from data and implement a recommender system.

**3.4 External Interface Requirement**

**Hardware Requirements**

* 32- or 64-bit Computer
* Operating System: Windows / Linux / MacOS
* Minimum 3GB of disk space for download and install
* RAM: Minimum of 4GB
* Processors: Intel® Core™ i3 processor onwards
* Python 3.6.x
* Pandas, Numpy, Scikit-Learn and Matplotlib libraries

**Software Requirements**

|  |  |
| --- | --- |
| **Software Used** | **Description** |
| 1. Anaconda | Anaconda is a freemium open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment. |
| 2.PyCharm | PyCharm is an integrated development environment (IDE) used in computer programming, specifically for Python.  Uses include: code analysis, a graphical debugger, an integrated unit tester,specialized project views, refactoring and multiple plugins. |
| 3. Pandas Library | The python pandas library is an open source project that provides a variety of easy to use tools for data manipulation and analysis. |
| 4. NumPy | NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. |
| 5. Scikit-learn Library | Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting. |

**Table 3.4.1 Software Requirements**

**3.5 Functional Requirements**

**Functional Requirement 3.5.1 (Processing of data)**

Using Pandas:

* Input: Ball-by-Ball data
* Output: Processed Data

**Functional Requirement 3.5.2 (Data Visualization)**

Using Matplotlib and seaborn:

* Input: Processed Data
* Behavior: The system projects different attributes from the processed data for the purposes of a better understanding of the statistical elements, a main part of it being the variation of a batsman’s performance according to bowling actions
* Output: Data Visualization

**Functional Requirement 3.5.3 (Recommending Players)**

Using Sklearn:

* Input: Processed Data
* Behavior: The system incorporates the data in a K-D Tree and recommends a player with similar characteristics using NearestNeighbors
* Output: Processed Data And Visualization

**3.6 Non-Functional Requirements**

* **Handling Data**

The data requirements are very simple and direct. The data is to be supplied to the model in the form of a dataset which will be worked upon using the Pandas library. A database may be used to retrieve the data in which case we may use an SQL based framework, however since this is a purely machine learning-based project we may use data available directly for the model to work upon.

* **Scalability**

The scalability of the model depends on the amount of data provided to be studied. The model is only as accurate in prediction as the quality of the data set provided. With more data a better working model can demonstrate more accurate predictions.

* **Availability**

The model will be available for testing and by direct access to the source code and data set.

* **Usability**

The model will allow the us to understand players that are similar to each other in specific areas and the accuracy will depend on the training data provided.

* **Correctness**

The model will be using various parameters to tabulate and visualize data to extrapolate relevant conclusions. This model also uses several algorithms to process the data and obtain the most accurate predictions for similar players in a particular category.

* **Maintainability**

The frameworks and technology used is up to date and can be easily maintained to satisfy needs that may arise in the future.

* **Portability**

The model will be portable across systems that support the required software specifications and consist of the tools used to develop the model.

**3.7 Performance Requirements**

* The data regarding the sport required for testing and drawing the required inferences must be collected and fed to the model.
* The data provided must be legitimate and must be collected by trustworthy procedures.
* The inferences drawn are to be studied from the point of view of analytical information which may help in drawing further conclusions which may change which players a team chooses to field.
* The model should then be used to recommend a player that has similar levels of performance as a given player.

**3.8 Risk Identification and Assessment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Risk** | **Probability** | **Impact** | **Exposure** | **Mitigation Plan** |
| 1 | Incorrect Inferences   * There are many factors that lead to incorrect inferences drawn such as assuming co-relation is the same as causation. | High | High | High | * Proper integration of code. * Use of efficient algorithms for analysis. * Availability of correct data sets pertaining to the study. * Cognizance of risks while making inferences. |
| 2 | Insufficient data - speed of ball, spin of ball, weather conditions. | High | High | High | * Algorithms that require less training data can be implemented. * Results adjusted for error. |
| 3 | Accuracy and Completeness of Data.   * Data may not be accurate and may not properly describe the required attributes for analysis. | Medium | High | Medium | * Proper methods of data collection. * Use reliable sources for obtaining datasets. |
| 4 | Incomplete data set   * Certain values may be missing in the chosen dataset. | Medium | Medium | Medium | * Use methods such as mean imputation and imputation by regression to fill in missing values.Use of unbiased attribute as target variables. |

**Table 3.8 Risk Identification**

**Chapter 4**

**MODELLING AND IMPLEMENTATION**

**DESIGN**

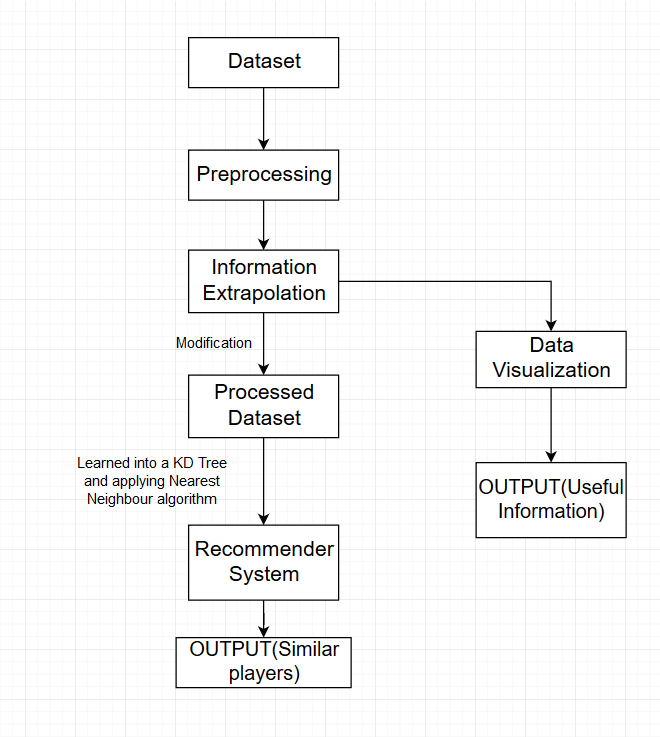
**4.1 Introduction**

The system is designed to analyze data and implement a machine learning model, we modify the dataset in the preprocessing stage per requirements. This process includes scaling and aggregation. From this dataset, we extrapolate important information. We then store the dataset in a tree format as our recommender system can be reduced to a search problem. We use unsupervised machine learning methods (Nearest Neighbour algorithm) to obtain a recommendation from the model.

Keeping in mind our requirements, we are able to implement a model which follows the architecture given below:

**4.2 ArchitectureDiagram**

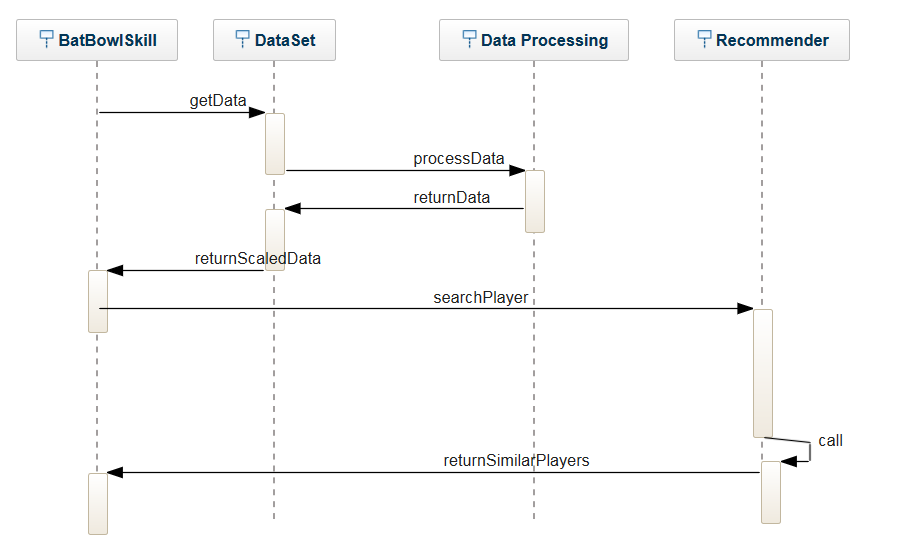
The architecture of our model requires acquisition of required datasets from available public domain sources as the first step. These datasets are preprocessed which involves using methods like joining, scaling, aggregation etc. These procedures are carried out using the Pandas library. The processed data is then visualized to allow us to make inferences and draw conclusions from it. This useful information is a resultant of data processing techniques and forms one part of the core outputs of our project. This processed dataset also functions as the input into our recommender system. The dataset is arranged in a K-D Tree structure and using nearest neighbour algorithm, the recommender system gives us player recommendations which forms the other part of the core outputs of this project. This recommender system utilizes Sci-kit Learning library.

****

**Fig. 4.2.1 Architecture Diagram**

**4.3 SEQUENCE DIAGRAM**

The recommender system requires statistical averages like average runs per match, strike rates in different innings, to perform the task of finding similar players to the given player. The ball by ball data is fed to and handled by the data processing sections of the program which generates a new csv file that contains all the required statistical metrics. This new csv file forms the input for the Recommender System. The Recommender System then uses this data to identify similar players to the given player and returns them as output.

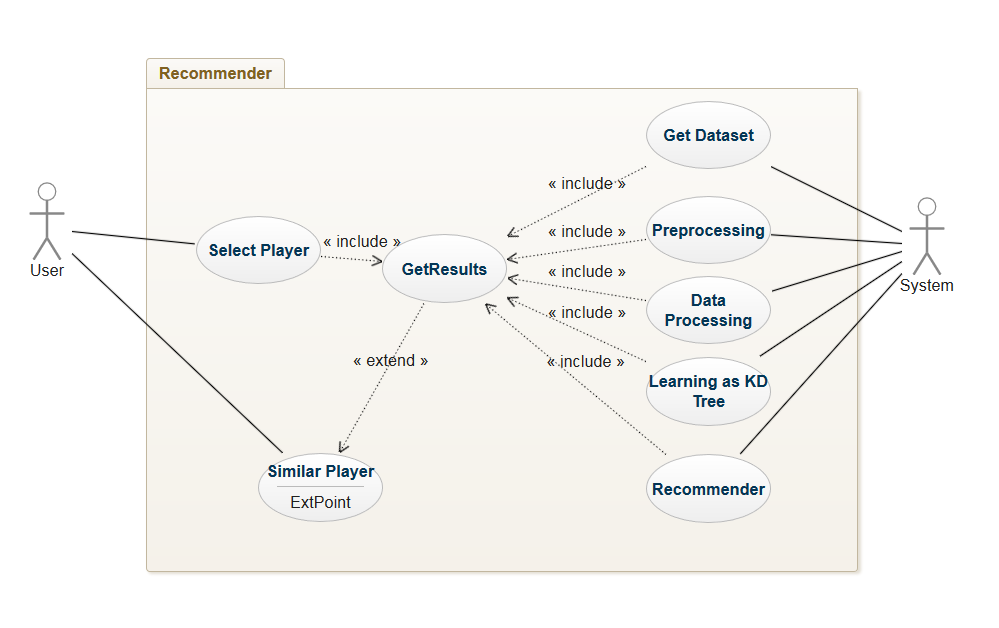
****

**Figure 4.3.1**

**4.4 USE CASE DIAGRAM**

The System side of our project gets the dataset and preprocesses data. This processed data gives us meaningful information and is then learned into a K-D Tree structure which functions as the input for the recommender system to recommend players.

The User side then provides input in the form of a search query. The User must provide the name of the player, and the system identifies similar players based on the metrics arrived at during preprocessing.

****

**Figure 4.4.1**

**IMPLEMENTATION**

**4.5 Tools Introduction**

The languages used in the project were Python 3.x. They were coded using the PyCharm IDE as part of the Anaconda environment. The essential libraries used in handling of data were NumPy and Pandas. It was essential to visualize data to draw appropriate inferences, and for this matplotlib and seaborn were used. The K-Nearest Neighbours algorithm was executed using the scikit-learn library.

**4.6 Technology Introduction**

**Scikit-learn:** It is an open source machine learning library which consists of simple and efficient tools for data mining and data analysis. The manipulations and computations were extremely useful in plotting the data. The library functions are built on NumPy and SciPy.

**PyCharm Notebook:** PyCharm is an integrated development environment (IDE) used in computer programming, specifically for Python.

Uses include: code analysis, a graphical debugger, an integrated unit tester,specialized project views, refactoring and multiple plugins.

**Anaconda**: Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine learning related applications, that aims to simplify package management and deployment.

**4.7 EXPLANATION AND IMPLEMENTATION ALGORITHMS**

**PREPROCESSING**

As part of the preprocessing, null values are removed and semantic changes are made to the dataset to ensure uniformity.

**PROCESSING OF DATA FOR VISUALIZATION**

Inorder to process data efficiently, we calculate certain important values from the preprocessed dataset. This includes figures of statistical importance such as run rate, strike rate, economy rate, average runs per game etc. Using these figures, we are able to visualize data to identify players that exhibit optimal levels of performance.

**MODELING INTO K-D TREE**

The k-d tree is a binary tree in which every leaf node is a k-dimensional point. Every node that isn’t a leaf node can be imagined to be generating a bifurcating hyperplane that divides the space into two parts called half spaces. Points that appear to the left of the bifurcating hyperplane are represented by the left subtree and points to the right of the hyperplane are represented to by the right subtree of that node. We need to model the data in the form of a K-D tree as the recommender system can be simplified as a search problem. As we use nearest neighbour algorithm, formatting the data as a tree gives us the benefit of eliminating a large portion of the search space, thereby simplifying the process.

### **CONSTRUCTION OF KD TREE**

As there are many possible ways to choose axis-aligned splitting planes, there are multiple different ways to construct k-d trees. The method of k-d tree construction has the following constraints:

* As we traverse down the tree, we cycle through the axes used to select the splitting planes. (For example, in a 3-dimensional tree, the root would have an *x*-aligned plane, the root's children would both have *y*-aligned planes, the root's grandchildren would all have *z*-aligned planes, the root's great-grandchildren would all have *x*-aligned planes, the root's great-great-grandchildren would all have *y*-aligned planes, and so on.)
* Points are inserted by selecting the median of the points being put into the subtree, with respect to their coordinates in the axis being used to create the splitting plane. (Note the assumption that we feed the entire set of *n* points into the algorithm up-front.)

This method leads to a balanced k-d tree, in which each leaf node is approximately the same distance from the root. However, balanced k-d trees aren’t optimal for all purposes. In our project, the different dimensions of the KD tree include all metrics identified during the processing stages.

**NEAREST NEIGHBOUR ALGORITHM**

The nearest neighbour search (NN) algorithm aims to find the point in the tree that is nearest to a given input point. This search is done efficiently using tree properties to eliminate large areas of the search space and optimize search time.

The following steps explains how searching for a nearest neighbour in a *k*-d tree works :

1. Starting from the root node, the algorithm recursively moves down the tree, in the same manner that it would if the search point were being inserted ( it goes left or right depending on whether the point is lesser than or greater than the current node in the split dimension).
2. Once a node is reached by the algorithm, the node point is checked and if the distance is better, that node point is saved as the "current best".
3. The recursion of the tree is unwound by the algorithm, and at each node the following steps are performed :
   1. If the current node is closer than the current best, then it is now considered the current best.
   2. A check is carried out by the algorithm for points on the other side of the splitting plane that are closer to the search point than the current best. Theoretically, this is done by intersecting the splitting hyperplane with a hypersphere around the search point that has a radius equal to the current nearest distance. As all the hyperplanes are axis-aligned, this is implemented in the form of a comparison to test if the distance between the splitting coordinate of the search point and current node is lesser than the distance (overall coordinates) from the search point to the current best.
      1. If the plane is crossed by the hypersphere, nearer points could exist on the other side of the plane, hence, the algorithm must look for closer points and continue to move down the other branch of the tree from the current node, following the same recursive process as the entire search.
      2. If the hypersphere doesn't intersect the splitting plane, then the algorithm continues walking up the tree, and the entire branch on the other side of that node is eliminated.
4. When this process is finished for the root node in this algorithm, then the search is complete.

In our project, upon receiving the input of the name of the player, the algorithm traverses the KD tree to find a player who resembles the attributes of the original player as closely as possible. We are able to compute the distance between the datapoints for these two players. We are also able to generate multiple neighbours to a particular datapoint which means the system recommends multiple players as alternatives to the searched player.

# 4.8 Pseudo Code

The following steps show the preprocessing that is required to ensure the the output doesn’t suffer from error due to simple data collection errors

>Import all the required packages.

>Read ball-by-ball data set as deliveries, match data set as matches.

>Perform initial preprocessing for data visualization

-Drop Umpire3 as too many NULL values

-Change Pune Supergiant to Pune Supergiants

-Change all occurences of Bangalore to Bengaluru.

The next step is to extract all the necessary information from the deliveries data frame, by performing aggregation operations to determine statistics like run, scored, wickets, strike rate, etc

The following shows how the dataset of NewBowl, NewBat are formed.

Batsman Dataframe

Runs 🡨 Group by batsman, aggregate sum of ‘batsman\_runs’ from deliveries

Balls🡨 Group by batsman, aggregate count of ‘ball from deliveries

Matches\_Played🡨 Group by batsman where Match\_id is unique

Boundaries\_4 🡨 Group by batsman, aggregate sum of ‘batsman\_runs’ where ‘batsman\_runs’ = 4

Boundaries\_6 🡨 Group by batsman, aggregate sum of ‘batsman\_runs’ where ‘batsman\_runs’ = 6

Dot\_balls 🡨 Group by batsman, aggregate sum of ‘batsman\_runs’ where ‘batsman\_runs’ = 0

Batsman\_score 🡨 Use inner join to merge the columns

These metrics vary according to the matches each player has performed in, so determining the statistical averages of these columns more accurately represent what type the player is

Batsman Dataframe

Strike\_rate 🡨 ((batsman\_runs / balls played))\*100

Avg\_runs 🡨 (batsman\_run/matches\_played)

Percent\_of\_boundaries\_4 🡨 Boundaries\_4 / balls

Percent\_of\_boundaries\_6 🡨 Boundaries\_46/ balls

Percent\_of\_dot\_balls 🡨 Dot\_balls / balls

The Final step is take the batting hand of players from another dataset DIM\_PLAYER.csv

Batsman Dataframe

>Read DIM\_PLAYER.csv as player\_data

>Drop all columns except name and batting hand

>Merge with batsman\_score over player\_name using Inner join

>Save batsman\_score dataframe to CSV file ‘NewBat.csv’

Similar steps need to be performed for bowlers, except the metrics vary as follows

Bowler Dataframe

Runs 🡨 Group by bowler, aggregate sum of ‘batsman\_runs’ from deliveries

Balls🡨 Group by bowler, aggregate count of ‘ball from deliveries

Extras🡨 Group by bowler, aggregate sum of ‘extra\_runs’ from deliveries

Matches\_Played🡨 Group by bowler, where Match\_id is unique

Boundaries\_4 🡨 Group by bowler, aggregate sum of ‘batsman\_runs’ where ‘batsman\_runs’ = 4

Boundaries\_6 🡨 Group by bowler, aggregate sum of ‘batsman\_runs’ where ‘batsman\_runs’ = 6

Dot\_balls 🡨 Group by bowler, aggregate sum of ‘batsman\_runs’ where ‘batsman\_runs’ = 0

Bowler\_score 🡨 Use inner join to merge the columns

Statistical averages as the next part:

Bowler Dataframe

Economy\_rate 🡨 ((batsman\_runs / balls played))\*6

Avg\_runs 🡨 (batsman\_run/matches\_played)

Percent\_of\_boundaries\_4 🡨 Boundaries\_4 / balls

Percent\_of\_boundaries\_6 🡨 Boundaries\_46/ balls

Percent\_of\_dot\_balls 🡨 Dot\_balls / balls

Getting Bowling styles from DIM\_PLAYER.csv

Bowler Dataframe

>Read DIM\_PLAYER.csv as player\_data

>Drop all columns except name and Bowling\_style

>Merge with bowler\_score over player\_name using Inner join

>Save bowler\_score dataframe to CSV file ‘NewBowl.csv’

## **Recommender System**

This part of the code deals with recommendation of similar types of player with respect to the metrics we have arrived at above. We work with the new CSV file that has been generated

>Import necessary sklearn library modules

>Read NewBat.csv into data and data\_dup

>Input the name of the player whose similar player needs to be found into ‘finding’

>Set index of dataframes to the names of player (a way to preserve name even after dropping it

>Drop non-statistical columns

>Generate tree as KDTree with leaf size = 3

>Locate the nearest neighbour of ‘finding’ player

>Search data\_dup to get all columns corresponding to the name

**Chapter 5**

**TESTING, RESULTS AND DISCUSSION**

**5.1 TESTING**

**5.1 Testing tools and Environment**

Testing tools – PyCharm Notebook

Testing Environment – Ubuntu or Windows 7 onwards with at least 4GB RAM

**5.2 Test Cases**

**5.2.1 Missing/noisy data**

The missing values and noise of all the datasets have to be dealt before applying any algorithm.

**5.2.2 Size of Dataset**

If the size of the dataset is above 5-10 MB then the number of tuples in dataset is over several millions which might contains noisy data that can lead to wrongful predictions.

**5.2.3 Inferences across several datasets**

When data from multiple datasets are put together, the attributes removal and selections need to be done with caution, as important attributes can be disregarded, and there might be several new attributes which were initially not present in old dataset.

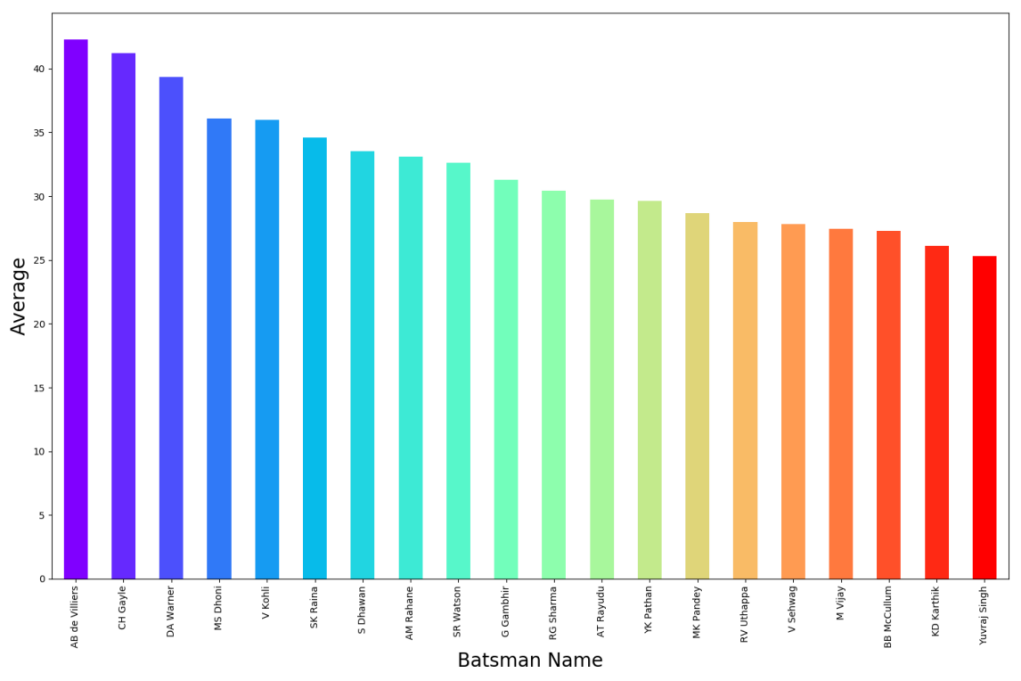
**5.3 RESULTS AND DISCUSSION**

After the first step of processing raw data, we are able to understand our data better using data visualization models in order to make concrete and helpful conclusions. Some of the important conclusions we draw are presented below.

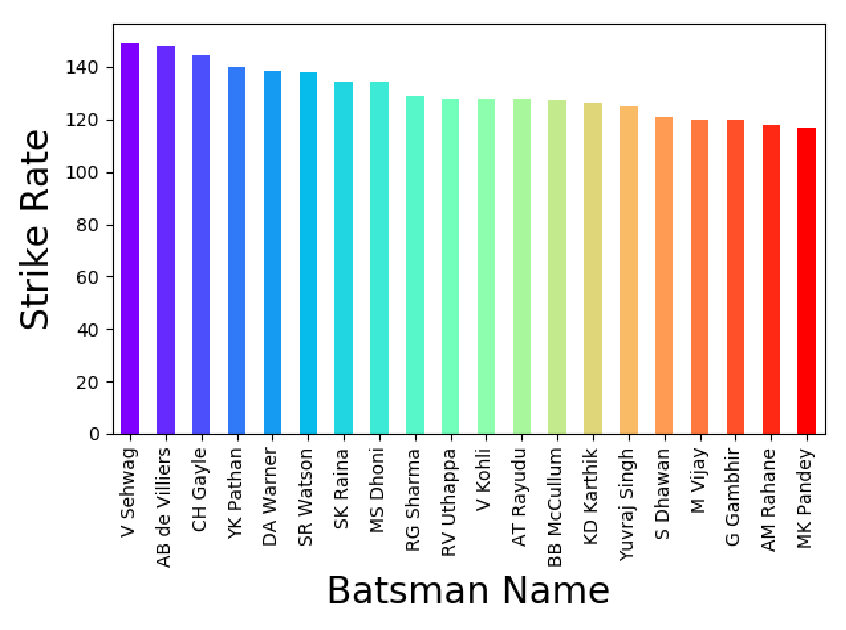
**Top Batsmen:**

Using strike rate and batting average of each player, it’s easy to immediately identify players who have been consistently performing well in this format.

The following data shows the best performing batting players:



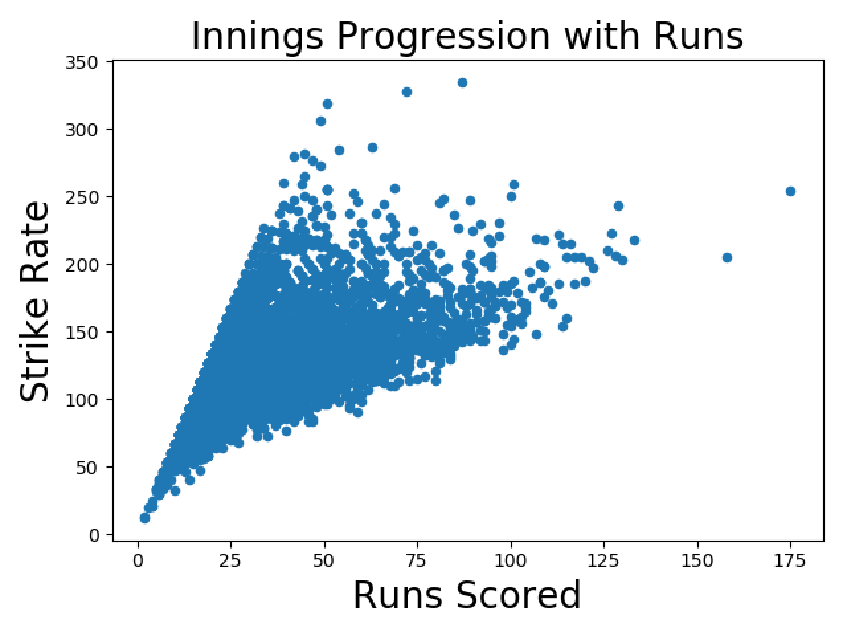
**Fig. 5.3.1**



**Fig. 5.3.2**

From this data, it becomes evident the top batsmen not only have a high strike rate, but also a good batting average, so the representation of players in both paradigms will have significant overlap.

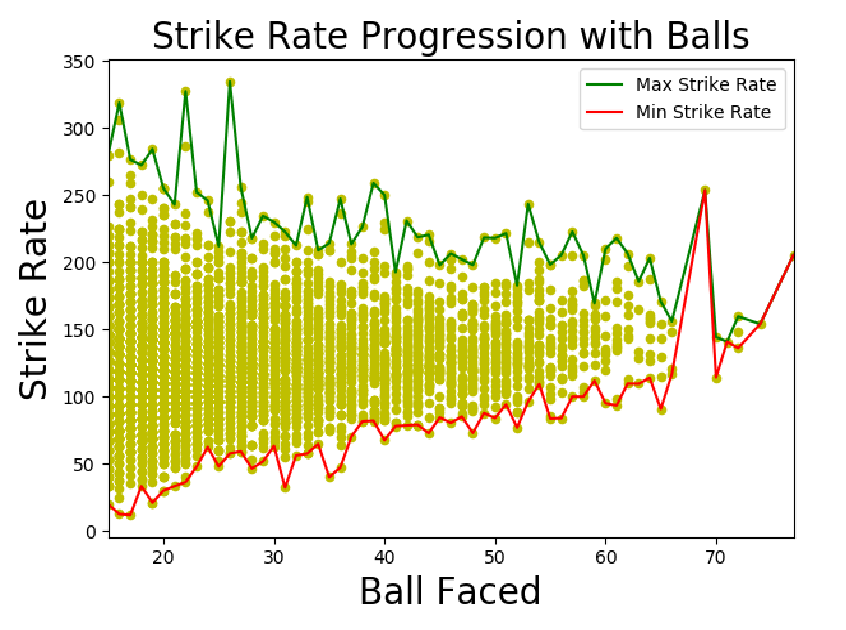
The next observation made was with respect to strike rate as the innings progressed :



**Fig. 5.3.3**

Using this data, we can conclude that increase in the number of runs scored tends to co-relate to an improved strike rate. This can be tied down to the fact that as a player faces more balls from different bowlers, the player is acclimatized to the match at hand and hence the rate of scoring runs increases. The ratio tends to increase and we get a graph resembling the one displayed above. We notice that variation in strike rate can be caused by the period of the game the batsman has entered the game. In the later stage of the innings, batsmen tend to be more aggressive and try to take bowlers on thereby scoring a significant amount of runs with a higher strike rate than their aggregate. However, we can see that in general when the score is high the lowest limit of the strike rate is increasing and the maximum limit also tend to increase.

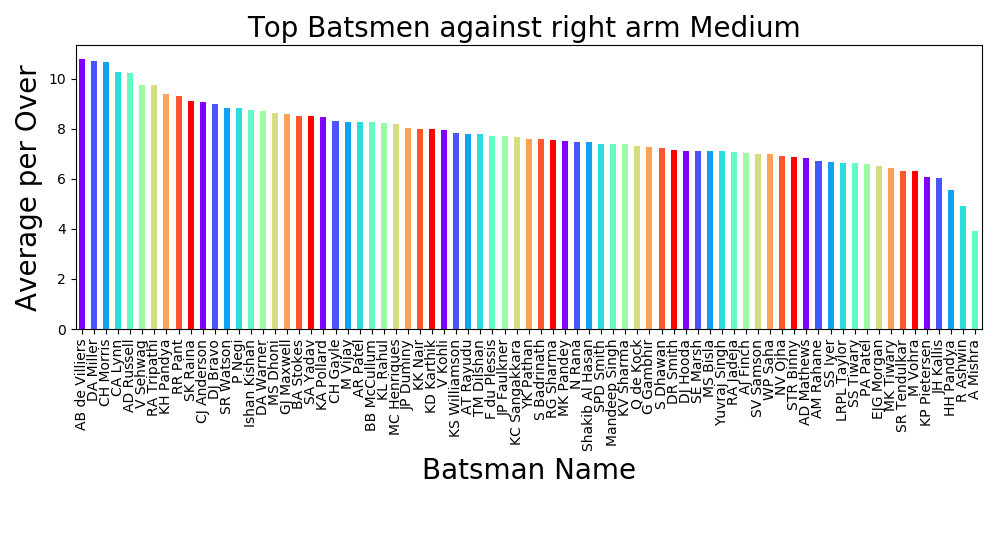
Further analyzing the data, we are able to reach a more comprehensive understanding.



**Fig 5.3.4**

The difference in the strike rates is stark at the beginning as different players opt to strategize differently about starting their innings; going all guns blazing and taking bowlers on or playing conservatively to protect their wicket. However, as the number of balls faced by the batsman increases, the strike rate tends to even out. The window of difference between maxiumum strike rates recorded at a particular number of balls faced and the minimum strike rates recorded keeps on reducing as the number of balls faced increases.

As we move towards the goal of recommending good players, it’s important to take into account the composition of opposing teams and be able to strategize against it. Here, we see the data of the strongest batsmen facing right arm medium paced bowlers, all of whom should be prioritized while facing a team depends on right arm medium paced bowling



**Fig 5.3.5**

Players like AB de Villiers, David Warner and Chris Morris stand out here.

In this manner, we are able to visualize data to combat any style of bowling, by making a graph relevant to the players in a specific team.

We can also modify our recommender system to give us the best possible batsman to field against a particular bowling style based on previous performances.

targname = 'SP Narine'

Best Player:  
  
Player\_Name KL Rahul  
Percent\_dot\_balls 0.3125  
No\_of\_6 1.5  
No\_of\_4 1.5  
batsman\_runs 70  
ball 32  
Matches\_batted 4  
Strike\_Rate 218.75  
Avg\_run\_per\_match 17.5  
percent\_boundaries 1.5  
Name: 21, dtype: object  
Distance from Player: 1.3539797641056206   
  
  
Player\_Name CH Gayle  
Percent\_dot\_balls 0.537037  
No\_of\_6 0.142857  
No\_of\_4 0.714286  
batsman\_runs 46

ball 54  
Matches\_batted 7  
Strike\_Rate 85.1852  
Avg\_run\_per\_match 6.57143  
percent\_boundaries 0.371429  
Name: 4, dtype: object  
Distance from Player: 1.357742328797376

Here, we observe that the best batsmen to field against Sunil P Narine is Chris Gayle and K L Rahul based on previous data.

The key element of our project, the recommender system, derives from these data visualizations to achieve its end results of identifying players who produce similar levels of performance and methods of achieving them.

For Virat Kohli:

Enter Name to find Similar Player:  
V Kohli

Similar Player:  
  
Player\_Name SK Raina  
Percent\_dot\_balls 0.340048  
Percent\_of\_6 1.0814  
Percent\_of\_4 2.60465  
Batting\_hand Left-hand bat  
Country\_Name India  
batsman\_runs 5014  
ball 3723  
Matches\_batted 172  
Strike Rate 134.676  
Avg\_run\_per\_match 29.1512  
Name: 5, dtype: object  
Distance from Player: 165.47776837848454

For Sachin Tendulkar:

Enter Name to find Similar Player:  
SR Tendulkar  
  
Similar Player:  
  
Player\_Name R Dravid  
Percent\_dot\_balls 0.427007  
Percent\_of\_6 0.341463  
Percent\_of\_4 3.28049  
Batting\_hand Right-hand bat  
Country\_Name India  
batsman\_runs 2174  
ball 1918  
Matches\_batted 82  
Strike Rate 113.347  
Avg\_run\_per\_match 26.5122  
Name: 22, dtype: object   
  
  
Player\_Name AM Rahane  
Percent\_dot\_balls 0.376068  
Percent\_of\_6 0.546218  
Percent\_of\_4 3.02521  
Batting\_hand Right-hand bat  
Country\_Name India  
batsman\_runs 3445  
ball 2925  
Matches\_batted 119  
Strike Rate 117.778  
Avg\_run\_per\_match 28.9496  
Name: 8, dtype: object

Although the playing styles of Rahul Dravid and Sachin Tendulkar might be different in all their nuances, using our model, we are able to understand that the levels of performance produced by them in the IPL format is quite similar.

By using this recommender system, we are able to find players who are similar to one another. While this model is currently implemented on IPL, it can be extended to include International Cricket statistics.

The same model functions to find similar bowlers as well, for example :

Enter Name to find Similar Player:  
SR Watson  
  
Similar Player:  
  
Unnamed: 0 82  
Player\_Name IK Pathan  
Wickets\_per\_match 0.980198  
Percent\_dot\_balls 0.451491  
No\_of\_6 0.811881  
No\_of\_4 2.90099  
Bowling\_skill Left-arm medium-fast  
Country\_Name India  
Runs\_Conceded 2569  
ball 2113  
extra\_runs 1.40594  
Matches\_bowled 101  
Economy\_Rate 7.69806  
Name: 82, dtype: object  
Distance from Player: 54.927912161283864   
  
  
Unnamed: 0 31  
Player\_Name Z Khan  
Wickets\_per\_match 1.20202  
Percent\_dot\_balls 0.441564  
No\_of\_6 0.646465  
No\_of\_4 3.15152  
Bowling\_skill Left-arm fast-medium  
Country\_Name India  
Runs\_Conceded 2691  
ball 2276  
extra\_runs 1.70707  
Matches\_bowled 99  
Economy\_Rate 7.53954  
Name: 31, dtype: object  
Distance from Player: 178.00900926864875

**Chapter 6**

**CONCLUSION AND FUTURE WORK**

In sports, it’s important to understand trends and patterns that indicate similarity in performance between players. Therefore, this project attempts process data to identify players who exhibit high levels of performance and recommend other players who can emulate such performances. The project has begun with gathering and refining data for processing and analysis. It involved combining different datasets in order to extrapolate information and visualize data. The combined dataset has been used to facilitate machine learning algorithms that can recommend similar batsmen and/or bowlers to the required player. Thus, we are able to implement a functional recommender system.

As future work, we aim to carry out the following:

* Attempt to apply models that follow similar methodology to different sports where stronger data exists.
* Using clustering, players could be divided into similar groups based on their performance which can be used to identify most skilled players. Attributes can also be used for clustering.
* Using more nuanced data like average pace of ball per bowler, average degree of spin or swig per ball per bowler, performance of bowler under different condition etc. we can use deep learning methods to train a batsman to face a particular bowler.

**REFERENCES**

[1]P. Somaskandhan, G. Wijesinghe, L. B. Wijegunawardana, A. Bandaranayake and S. Deegalla, "Identifying the optimal set of attributes that impose high impact on the end results of a cricket match using machine learning," 2017 IEEE International Conference on Industrial and Information Systems (ICIIS), Peradeniya, 2017, pp. 1-6.

URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8300399&isnumber=8300323

[2]P. A. Gregory, D. H. M. S. N. Herath, D. S. L. Karunasekera, D. S.Deegalla, and A. U. Bandaranayake, “Cricket sabermetrics: A data mininganalysis of cricket,”in Proceedings of the 6th YSF Symposium, pp. 33–38,2017.

1. Lewis, A J (2005). “Towards Fairer Measures of Player Performance in One-Day Cricket,” The Journal of the Operational Research Society, 56(7), 804-815.

[4]”An intelligent method for selecting and recommending best players to help build sports team”, N P Kurade

[5]”Moneyballer: An Integer Optimization Framework for Fantasy Cricket League Selection and Substitution”, Debarghya Das