**MINOR PROJECT 1**

**“Player Selection (IPL Team) using Clustering Algorithm”**

**by**

**NAME Roll Number**

**Lakshay Vasuja R172218064**

**Lakshay Sharma R172218063**

**Divyansh Chandna R172218045**

**Under the guidance of**

**Mr. Deepak Kumar Sharma**

Assistant Professor

Department of Informatics



**School of Computer Science**

**University of Petroleum and Energy Studies**

**Dehradun-248007**

**2018-22**

**TABLE OF CONTENTS**

**Contents**

1. **Introduction 3**
2. **Flowchart 3**
3. **Problem Statement 3**
4. **Objective 4**
5. **Methodology 4**
6. **Algorithm 5**
7. **Implemented Code 5**
8. **Work flow 6**
9. **Output Screens 6**
10. **Scope of Project 13**
11. **Contribution of each member 13**
12. **Conclusion 14**
13. **References 14**
    * + 1. **Introduction**

IPL also known as Indian Premiere league launched in 2008 by BCCI is a professional Twenty20 cricket league in India. IPL consists of 8 different teams representing 8 different cities in India. Players from all over the world take part in this tournament.

During the Auction Phase, Players are selected for every team so some situations may arise where the team doesn’t get their desired player as the bid has maxed out the team’s budget for the particular player and there can be a number of situations where this can happen.

* + - 1. **Flow Chart**

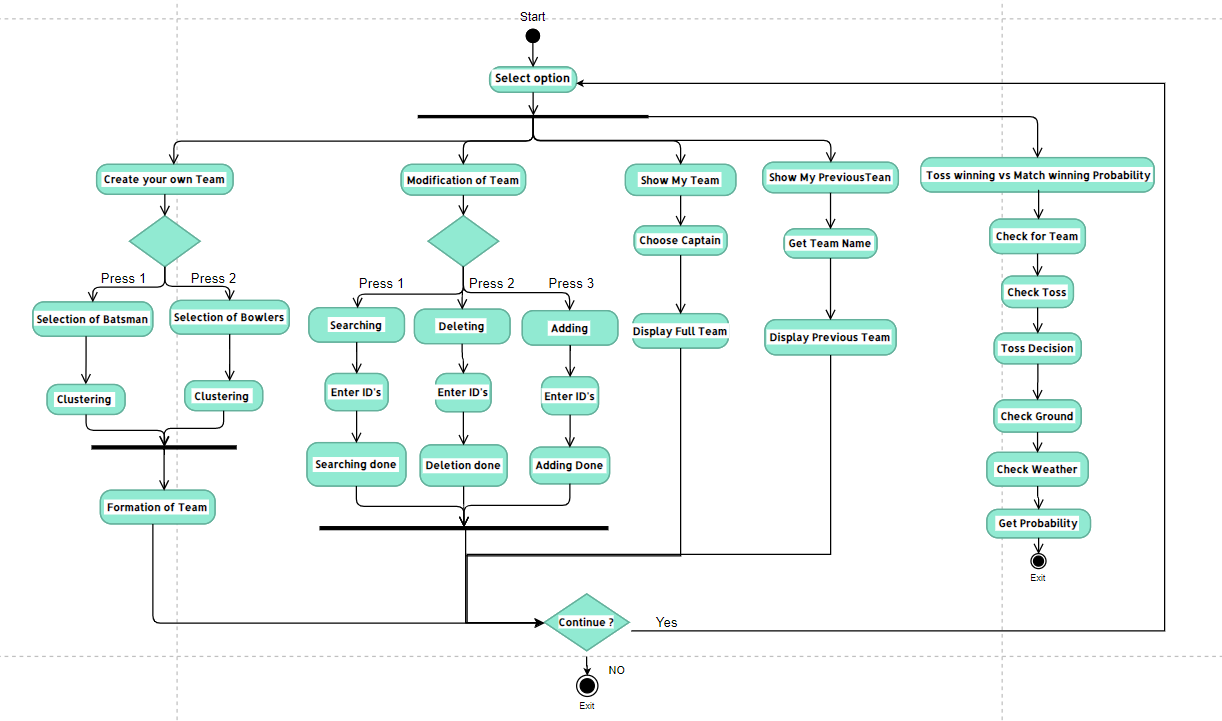


Fig 2.1

* + - 1. **Problem Statement**
* There can be a number of situations where desired players cannot be picked during auction so a substitute player has to be picked.
* To make a good and versatile team with different properties is a hard task as team management have to keep track of all the players and bid on them.
* Sometimes captains cannot make a well informed decision on what to choose after winning the toss.
  + - 1. **Objectives**
* Use K-Means clustering algorithm to create clusters of players based on their similar properties.:

Sub objective:-

* + 1. Implementation of K-Means to create clusters of players
    2. Making an IPL team for the user and adding functionalities like modifying the selected teams and deleting previous selected teams and searching players in the teams etc.
    3. Capping of players based on their bid to create a budgeted Team.
    4. Calculate the probability between toss winning vs match winning taking in account different parameters like toss decision, venue, weather conditions etc.
       1. **Methodology**
       2. Predicting the events that may happen IPL follows the past. Cleaning thedata and making it ready for the analysis will be the first step.
       3. A menu-driven user interface depicting functionalities like
* Create your own team
* Modification in your team
* Show my team
* Show previous team
* Probability of game winning vs toss winning
  + - 1. In “Create your own team” first of all user is asked to name the team following which a choice is given to select batsman or bowler. The user can specify the number of desired batsman or bowlers. Clustering is performed on the account of various parameters for batsman(Total runs, Out, Average, Strike rate) and bowlers(Economy, Wickets, Matches, Count, Balls per wicket, SR) separately. User can enter the number of desired clusters(3 to 6). After the clustering you can select your players create team.
      2. “Modification in your team” provides you with 3 options: Searching, Adding, Deleting of players.
      3. “Show my team” displays the players selected in the team along with the total cost of the team and selection of captain can also be done here.
      4. Show previous team displays the previously created teams and the user can view the players in those teams by entering the team name.
      5. Probability of game winning vs toss winning can calculate the probability of game winning using the dataset which has the parameters Team1, Team2, toss winner, toss loser, toss decision, Venue, Weather. The probability can only be calculated of the teams that have played in the IPL and are present in the dataset.
      6. **Algorithm**
* K-Means algorithm

1. Choose the number of clusters(K) and obtain the data points
2. Place the centroids c\_1, c\_2, .....c\_k randomly
3. For each data point x(i):
   * + 1. find the nearest centroid(c\_1, c\_2 .. c\_k) using Euclidean distance
       2. assign the point to that cluster
4. For each cluster I = 1…k
   * + 1. new centroid = mean of all points assigned to that cluster
5. Repeat steps 4 and 5 until convergence or until the end of a fixed number of iterations.
   * + 1. **Implemented Code**

Functions:

1. **Random()**

The first step of k-means is carried out with the help of this function. It selects a random index. All the values at that random index are taken as the initial values and then the Euclidean distance from that point to others is calculated and an array of those distances is returned. This function is used for selecting random value from batsman dataset.

1. **fileWriting()**

Here 3 inbuilt functions of file handling: fwrite,fseek,fread are used.

fwrite- Creation of text file of selected batsman

fseek- Traversing of selected players in text file

fread- Reading of selected player

1. **getProbability()**

First of all it asks for a number of inputs like selecting the team, outcome of the toss, their playing conditions and then it will calculate the probability of the match winning.

1. **displayProbability()**

This is an output function that displays the matches won, matches lost and total matches played and then if there is data to calculate the probability it will display the probability of match winning and match losing.

1. **Bowler\_Random()**

This function is used for selecting random value from bowlers dataset. Although it has the same functionality as the Random function the only difference is that it works on the bowlers dataset which has different attributes than batsman.

1. **Create\_Team()**

This is the key function for batsman. Functionalities like clustering, batsman selection, modification of batsman in your team are a part of this function.

1. **bowlers()**

This is the key function for batsman. Functionalities like clustering, bowlers selection, modification of bowlers in your team are a part of this function.

**8. Work flow (PERT Chart)**

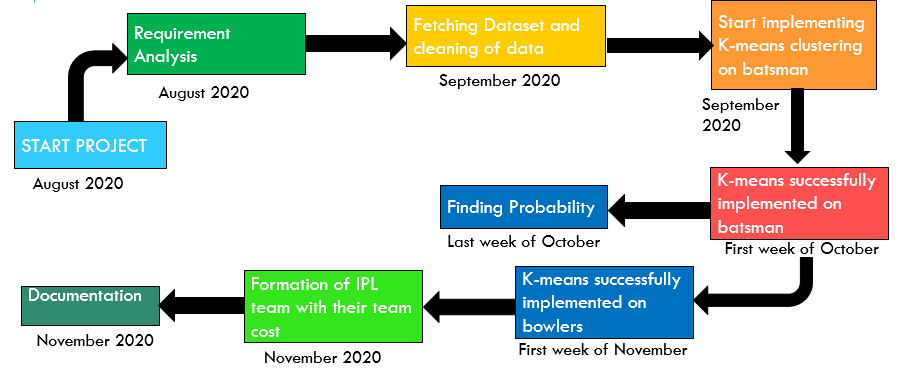
****

Fig 8.1

**9. Results**

Menu Driven

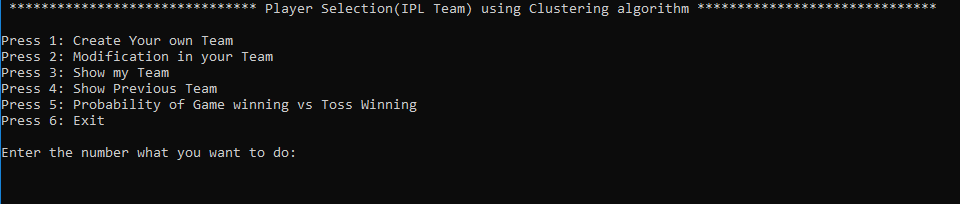


Fig 9.1

Above figure 9.1 describes the menu driven function of the project

Create your own Team

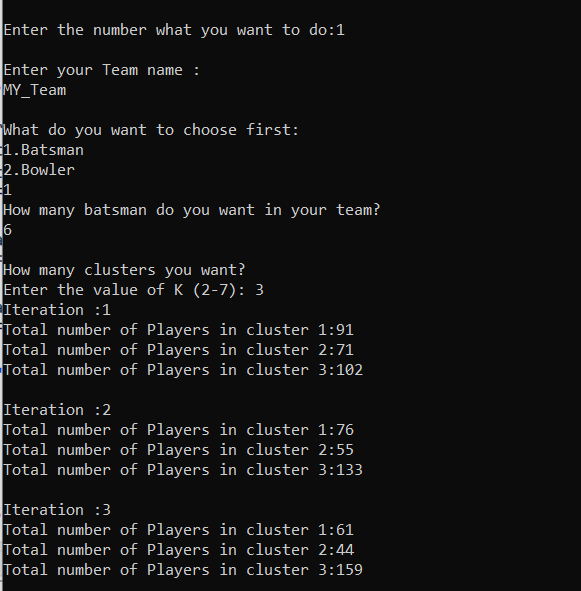


Fig 9.2

Fig 9.2 shows the clustering of batsman when the selected clusters are 3.

Displaying cluster 2 Players

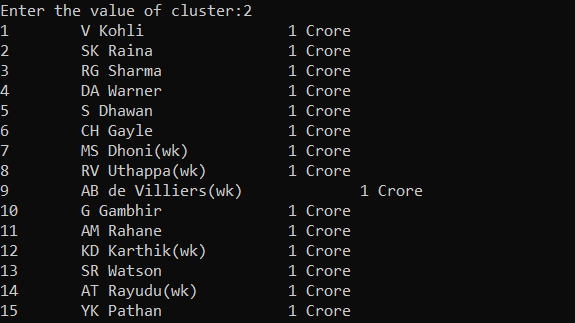


Fig 9.3

Fig 9.3 shows the batsman present in cluster 2 at that instance.

Selection of Batsman

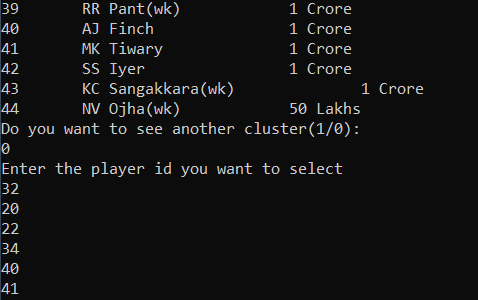


Fig 9.4

Fig 9.4 shows selected batsman which were selected by the user.

Selecting cluster value for selection of bowlers

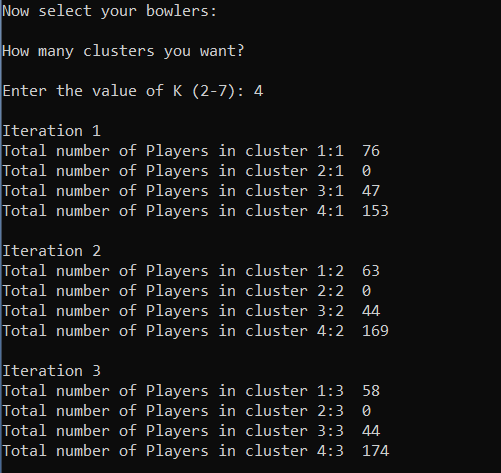


Fig 9.5

Fig 9.5 shows the clustering of bowlers

Displaying cluster 4 bowlers

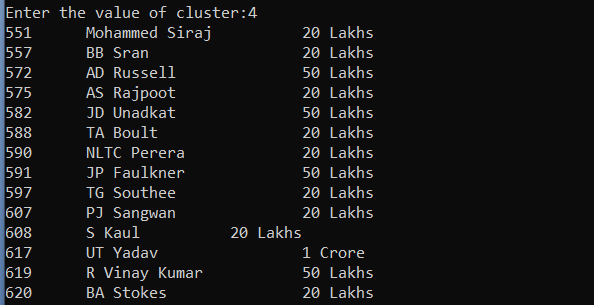


Fig 9.6

Fig 9.6 displays the bowlers of cluster 4 at that instance

Selection of bowlers for team

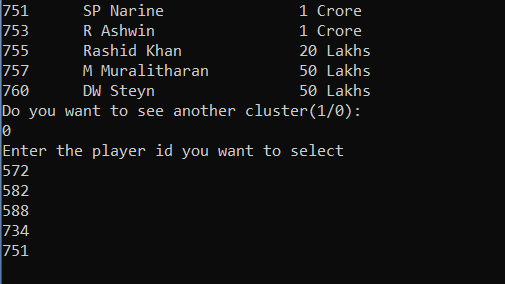


Fig 9.7

Fig 9.7 shows selected bowlers which were selected by the user.

Show My Team

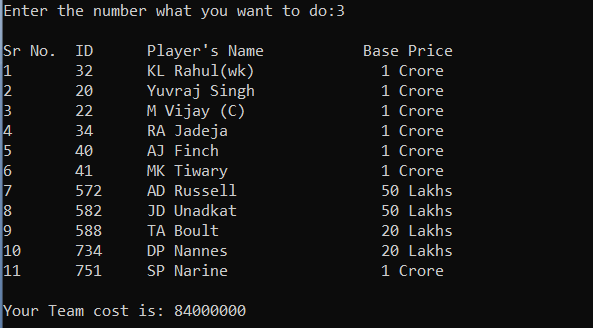


Fig 9.8

Fig 9.8 shows the final team representation selected by the user.

Modification in my Team

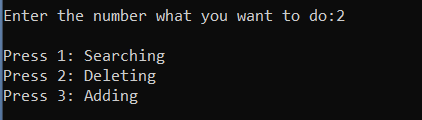


Fig 9.9

Fig 9.9 shows the menu options for the modification of team.

Searching of a Team player

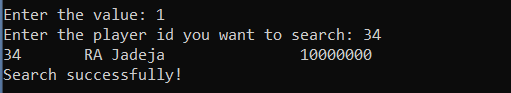


Fig 9.10

Fig 9.10 shows the successful search of a player in the team

Deletion of a player

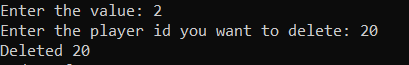


Fig 9.11

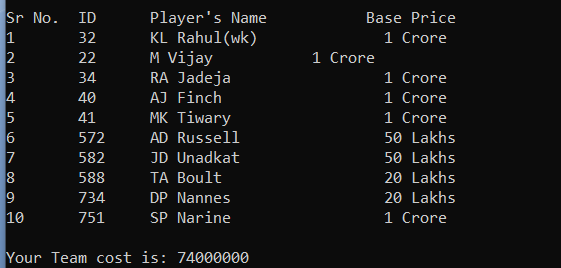


Fig 9.12

Fig 9.11 and Fig 9.12 shows the successful deletion of the player and shows the remaining players in the team.

Adding new player in Team

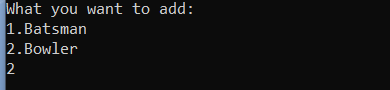


Fig 9.13

13.PNG

Fig 9.14

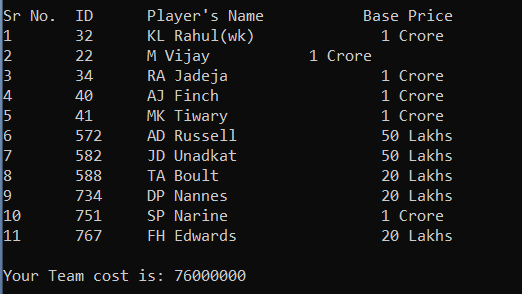


Fig 9.15

Fig 9.13, 9.14 and 9.15 shows the successful addition of player over the previously deleted player.

Show my previous Team

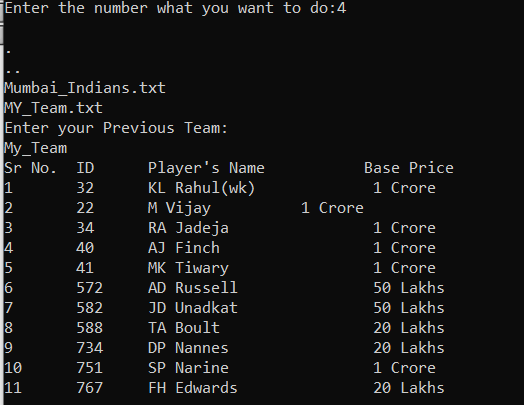


Fig 9.16

Fig 9.16 shows the previous created team that were created by the user.

Probability of game winning vs toss winning

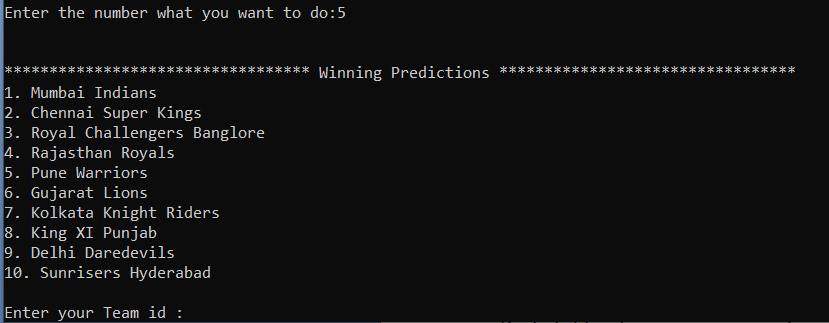


Fig 9.17

Results of probability

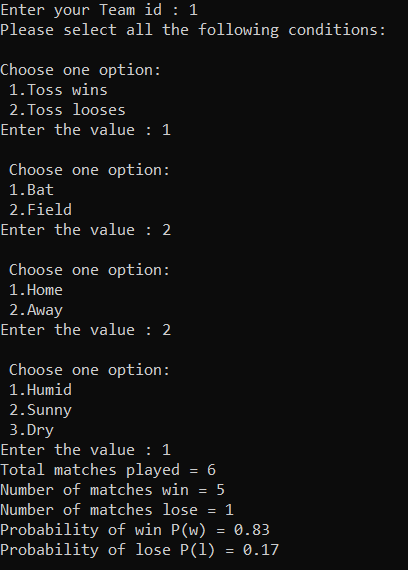


Fig 9.18

Fig 9.17 and 9.18 shows the result of the probability of game winning vs toss winning. In the above instance the probability of winning is higher than losing.

**10. Scope of project**

1. Selecting a team having best players comprises an important part of IPL auctions. We covered this scenario with the scope that there may arise situations where desired players are already booked, creating clusters of players having similar properties will help in choosing the substitute of the desired player.
2. In an IPL match, captains have an important responsibility of choosing field or bat after winning Toss, having a proper knowledge of pitch, weather conditions in which they have played will help captains to understand their hold over the match.
3. Budget and the performance of the team can be determined.
4. Every year many young players (say 300) present themselves in the auction, a few around (say 200) gets the chance to play. Teams are categorized having 16 players each, here arises the problem of grouping the data of all players such that we create a balanced team having good to average players considering the bidding amount.

**11. Contribution of each member**

**LakshayVasuja**

1. Finding and cleaning dataset
2. First iteration of batsman K-means
3. Probability of game winning vs toss winning
4. Making program menu-driven.
5. Clustering of Bowlers
6. Modification in team functionality

**Lakshay Sharma**

1. Read data from csv and storing into array of structures.
2. First iteration of batsman K-means
3. Creation of header files for linking
4. Testing and debugging
5. Previous team functionality
6. Modification in team functionality

**DivyanshChandna**

1. Making Random Function for selecting initial random values.
2. Second and third iteration of batsman K-means
3. Probability of game winning vs toss winning
4. Testing and debugging
5. Making team captain and wicket keeper
6. Clustering of Bowlers

**12. Conclusion**

This is a standalone application for formation of IPL team with base (minimum) team cost and the algorithm used for clustering of players is K-Means clustering applied successfully.

**13. References**

1. K. Alsabti S. Ranka and V. Singh, “An Efficient k-means Clustering Algorithm,” Proc. First Workshop High Performance Data Mining, Mar. 1998.
2. S. Arora P. Raghavan and S. Rao, “Approximation Schemes for Euclidean k-median and Related Problems,” Proc. 30th Ann. ACM Symp. Theory of Computing, pp. 106-113, May 1998.
3. Dataset Links:

<https://data.world/raghu543/ipl-data-till-2016-set-of-csv-files/workspace/file?filename=Toss_Decision.csv>

[https://data.world/raghu543/ipl-data-till-2017/workspace/file?filename=Match.csv](https://meet.google.com/linkredirect?authuser=1&dest=https%3A%2F%2Fdata.world%2Fraghu543%2Fipl-data-till-2017%2Fworkspace%2Ffile%3Ffilename%3DMatch.csv)

[https://www.kaggle.com/harsha547/indian-premier-league-csv-dataset?select=Player.csv](https://meet.google.com/linkredirect?authuser=1&dest=https%3A%2F%2Fwww.kaggle.com%2Fharsha547%2Findian-premier-league-csv-dataset%3Fselect%3DPlayer.csv)