An Analysis of Best Player Selection Key Performance Indicator: The Case of Indian Premier League (IPL)

Mayank Khandelwal, Jayant Prakash and Tribikram Pradhan

Abstract IPL is the most celebrated T20 cricket festival in the world in which 8 teams give their best to reach the top team in the tournament. In such a contest there are various players from different nationalities playing for different teams. As we know that only a certain amount of players can play one match, so there is a problem for team management to choose the best combination of players for the match. In this paper, we are calculating the *Most Valuable Player (MVP)* by using a novel approach, decision tree is used to classify the players into various classes. Further, bipartite cover is used for selection of bowlers, variance analysis is used to find the similarity among players. Finally, genetic algorithm is used to select the best playing eleven. After selecting the best players, we are predicting individual strike rates with total team scores. This paper is going to give them a solution to eliminate non performing players using customized method of their performance analysis in earlier matches, assembling a decent playing eleven for any match using revolutionary methods and deciding batting order in an efficient manner.

Keywords MVP · Decision tree · Bipartite cover · Co-variance · Genetic algorithm · Regression

1 Introduction

Indian Premier League (IPL) is a Twenty20 (T20) cricket extravaganza started by BCCI (Board of Control for Cricket in India) in 2008, which is held annually in the month of April - June. The first season of IPL was sponsored by DLF, which is a leading real estate company in India. The inaugural season of the tournament took place from 18 April - 1 June 2008. The challenge is that during an IPL auction only

M. Khandelwal · J. Prakash · T. Pradhan
Department of I&CT, Manipal University, Manipal 576104, India
e-mail: {mayankkhandelwal123,jayant.11snh}@gmail.com,
tribikram.pradhan@manipal.edu

S. Berretti et al. (eds.), *Intelligent Systems Technologies and Applications*,
Advances in Intelligent Systems and Computing 384,
DOI: 10.1007/978-3-319-23036-8_16

selected players can play the cricket match, hence the team owner must select the optimal combination of players. As of now, there is no fool-proof solution to this challenge nor is there any solution ranging from selection of bowlers to selection of batsmen maintaining experience within the team. A possible methodology consists of attempting to choose the best payer to buy among all the participants in an IPL auction for the team using a measure called *MVP*. MVP is dynamic in nature, which implies that comparison criteria changes over the proceedings of the auction. We then classify players according to their complete performance measured points, called *TCP*. Then, we attempt to choose the best set of bowlers using bipartite cover concept and batsmen using genetic algorithm. We attempt to decide the best batting order using genetic algorithm and using some interesting measures, we predict the results for the game.

2 Literature Survey

S. Singh, S. Gupta and V. Gupta in [1] proposed an integer programming real-time model for optimal strategy for binding processes. Spreadsheets were used to document & calculate the results since it was the optimal choice considering the flexibility of incorporation for more weight-age based on recent performance of a player to evaluate the final outcome.

S. Singh in [2] uses Data Envelopment Analysis to measure how effective teams are in IPL. The author calculates awarded points, total run rate, profit and returns by determining that total expenses including the wage price of players and staff as well as other expenses. Efficiency score is usually directly related to the performance of the player in the league. On decomposing the inefficiencies into technical and scale inefficiency, it is realize that the inefficiency is primarily due to unoptimized scale of production & unoptimized transformation of the results and the considered data.

P. Kalgotra, R. Sharda and G. Chakraborty in [3] develop predictive models which aid managers to select players for a talented team in the least possible price. This is calculated on the basis of the player's past performance. The optimal model is selected on the basis of the rate of validation data misclassification. This model helps in the selection of players by aiding in the author's bidding equation. This research also facilitates the managers to set the salaries for players.

F. Ahmed, K. Deb and A. Jindal [4] use NSGA-II algorithm to propose a new representation scheme & a multi-objective approach for selecting players in a limited budget considering the batting & bowling strengths along with the team formation. Factors such as fielding further optimize the results. The dataset to define performance is taken from IPL - 4th Edition. The author shows analysis in real-time auction events, selecting players one-by-one. The author argues that the methodology can be implemented across other fields of sports such as soccer etc.

Sonali B. and Shubhasheesh B. [5] focus on how teams strategically decide on the final bid amount based on past player & team performance in IPL and formats similar to IPL. The authors also shed light on how personalities of players can affect team performance. They analyze the possible factors based on which bidders decide and build a predictive model for pricing in the auction. The analysis is done individually for all the teams.

H. Saikia and D. Bhattacharjee [6] classify performances of all-rounders into 'Performer', 'Batting-All Rounder', 'Bowling-All Rounder' and 'Under-performer'. Further, they suggest and consider independent variables that influence an all-rounder's performance by using Step-wise Multinomial Logistic Regression (SMLR). The independent variables are used to predict the class of an all-rounder player using Naive Bayes Classification concept.

F. Ahmed, A. Jindal and K. Deb [4] suggest a multi-objective approach which optimizes and identifies the batting and bowling strengths of a team using NSGA-II algorithm. Information from the trade-off front enables decision making for final team selection. The study uses data from IPL - 4th Edition and player's statistical data as performance parameters. The authors argue that the methodology is generic is extend-able to other sports as well.

3 Methodology

In order to obtain *MVP*, here we have proposed a novel approach consisting of MVP calculation, classification using decision tree, selection of players using bipartite cover & genetic algorithm. Finally, similarity measure among players can be analyzed by using regression analysis to make a comparison between various teams. We propose the following steps in our methodology :

- Analysis of *Most Valuable Player* (MVP) by using a Set of Rules
- Dynamic Nature of MVP Calculation After Each Player's Selection Process
- Calculation of *Total Credit Point* (TCP) by using a Set of Rules
- Classification Among Selected Players of Individual Teams using Decision Tree
- Application of Bipartite Cover for the Selection of Best Bowlers & All-Rounders
- Finding similarity & dis-similarity among players using co-variance analysis
- Selection of Batsmen using Genetic Algorithm
- Deciding the Batting Order using Genetic Algorithm
- Predicting team scores with individual score & strike rate of a player using multiple linear regression analysis.

3.1 MVP Calculation

In this section, we need to find out the player's batting points (PBT), player's bowling points(PBW) and player's experience (PEX). In order to find out the above three formula's, we need to consider the following parameters : Player's Batting Average, Player's Batting Strike Rate, Number of centuries & half-centuries, Bowling Average, Bowling Strike Rate, Economy, Number of 4-wicket & 5-wicket haul and Number of Matches Played. We define the 'Most Valuable Player' (MVP) as the single parameter that can be used to compare any type of player in the auction. MVP is decided on the basis of requirement of type of player selected by the owner. For this, we need the 'Requirement Points' (minimum required in the team) for batting(BARP), bowling(BORP) and experience(ERP). 'Total Requirement' (TRP) is the sum of all requirement points (Batting + Bowling + Experience) i.e TRP = BARP + BORP + ERP

1. PBT = $(((BattingAverage * 0.3) + (BattingStrikeRate * 0.4) + \lceil NumberofHundreds \rceil + (NumberofFifties * 0.2))/10)$
2. **if that the bowler must have bowled minimum 100 bowls in his IPL career, then,**
   $PBW = (((300/BowlingAverage) + (200/BowlingStrikeRate) + (300/Economy) + \lceil Numberof4 - wicketshaul \rceil * 0.1 + \lceil Numberof5 - wicketshaul \rceil * 0.1)/10)$
3. PEX = (Number of Matches Played/Total Number of Matches in IPL so far)

Algorithm 1. Pseudo-code for MVP Calculation

```
procedure MVP(SetA, SetB)
    mvp = 0
    if PBW = 0) then
        mvp = (8*PBT*(BARP/TRP)+(ERP/TRP)*PEX)/10
    else
        if PBT/PBW>=2 then
            mvp = ((7*PBT*(BARP/TRP))+(2*PBW*(BORP/TRP))+(PEX*(ERP/TRP))/10)
        else
            if PBW/PBT>=2 then
                mvp = ((7*PBW*(BORP/TRP))+(2*PBT*(BARP/TRP))+(PEX*(ERP/TRP))/10)
            else
                mvp = ((9*PBW*(BORP/TRP))+(9*PBT*(BARP/TRP))+(2*PEX*(ERP/TRP))/20)
            end if
        end if
    end if
end procedure
```

3.2 Decision Tree

Decision Tree is powerful decisive tool used for Classification and Prediction. Every node is bonded with rules that help the data to be classified according to the nature defined by the rules. It is basically used in Data Warehouse for Knowledge Discovery. Following are the features of a Decision Tree:

- There must be finite number of distinct attributes for classification.
- Target values of data used for classification should be discrete.
- There should not be any missing data which are important for classification.

Algorithm 2. TCP Calculation

```
procedure TCP(PBT, PBW, PEX)
    if PBW = 0 then
        TCP = (8*PBT + 1*PBW + PEX)/10
    else if PBT/PBW>=2 then
        TCP = (7*PBT + 2*PBW + PEX)/10
    else if PBW/PBT>=2 then
        TCP = (2*PBT + 7*PBW + PEX)/10
    else
        TCP = (9*PBT + 9*PBW + 2*PEX)/20
    end if
end procedure
```

Following are the components of a Decision Tree:

- **Decision Node :** It is a non leaf node used to make a decision according to the relevant data taken into consideration for the classification.
- **Leaf Node :** Represents the final classification container that holds data post operations occurred at the Decision Node.
- **Path :** It represent the result used for classification of the data from the decision node.

In Decision Tree Data is classified starting from the root node using top down approach till the leaf node is encountered.

Algorithm 3 Pseudocode for Classification of Players using Decision Tree

```
procedure DETERMINE_CLASS(TCP)
    if TCP>=6.5 then
        return A+
    else if TCP>=6.0 then
        return A
    else if TCP>=5.5 then
        return B
    else if TCP>=5 then
        return C
    else if TCP>=4.5 then
        return D
    else if TCP>=6.5 then
        return E
    else
        return F
    end if
end procedure
procedure CLASSIFICATION(PBT, PBW, TCP)
    if (PBW = 0) or ( PBT / PBW ) >= 4 ) then
        if TCP>=6.5 then
            player_type = Batsman
            determine_class(TCP)
        end if
    else
        if (PBW/PBT)>=1.25 then
            player_type = Bowler
            determine_class(TCP)
        else
            if TCP>=6.5 then
                player_type = All-Rounder
                determine_class(TCP)
            end if
        end if
    end if
end procedure
```

3.3 Bipartite Cover

An Undirected Graph is a Bipartite Graph if it has n vertices's, partitioned into two sets A and B such that no two edges of the same set are connected. This means that all edges exist only between vertices's from Set A to vertices's from Set B. If vertices's of $A1$ which is a subset of A, connects all vertices's of B, then $A1$ covers B. The size of this cover is determined by the number of vertices's in $A1$. $A1$ is said to be minimum cover if no smaller subset of A covers B. Hence, Set C gives minimum cover with size x. Conclusively, the bipartite cover problem aims at getting minimum cover for a set of vertices in a bipartite graph. We have shown the pseudo-code for bipartite cover in *Algorithm 4*.

Algorithm 4. Pseudo-code for Bipartite Cover

```
procedure BIPARTITE(SetA, SetB)
    x = 0                                   ▷ initial number of selected vertices for cover
    C = new_cover_set                       ▷ Cover Set
    for all i in A do
        a_set[i]=Degree[i]                  ▷ Populating degrees of vertices in A
    end for
    for all i in B do
        b_set[i]=false                      ▷ marking all vertices in set B unreached
    end for
    while all a_set[i] > 0 for all nodes in A do
        r = vertex i of A with max value in a_set[i]
        append r to C
        increment x by 1
        for all vertex s adjacent from r do
            if r is not reached then
                mark r is reached in b_set[r]
                for all vertex t adjacent from r do
                    decrement a_set[t] by 1
                end for
            end if
        end for
    end while
end procedure
```

3.4 Covariance of Numeric Data

Correlation & covariance are measures to find how much attributes vary in accordance to each other. Consider A and B, and a set of n observations $\{(a_1, b_1), ..., (a_n, b_n)\}$ The mean or expected values of A and B:

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n} \quad and \quad E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n} \tag{1}$$

The covariance between A and B is defined as

$$Cov(A,B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n} \tag{2}$$

We observe that, $r_{A,B}$ (Correlation Coefficient)

$$r_{(A,B)} = \frac{Cov(A,B)}{\sigma_A \sigma_B} \tag{3}$$

where $\sigma_A \sigma_B$ are the standard deviations of A and B, respectively. We show that

$$Cov(A,B) = E(A.B) - \bar{A}\bar{B} \tag{4}$$

For A and B which are likely to change together, if A is greater than $\bar{A}$ (the expected value of A), then B will tend to be greater than $\bar{B}$ (the expected value of B). Hence, the covariance between A and B is *positive*. On the contrary, if one of the attributes is likely to be greater than its expected value when the other attribute is lower than its expected value, then the covariance of A and B is *negative*.

If A and B are independent, then E(A.B) = E(A).E(B). Hence, the covariance is Cov(A,B) = E(A.B) - $\bar{A}\bar{B}$ = E(A).E(B) - $\bar{A}\bar{B}$ = 0. Converse isn't true. Some pairs of random variables (attributes) may have a covariance of 0 but are dependent.

3.5 Genetic Algorithm

Genetic algorithms have proved to be particularly efficient for searching, optimizing, machine learning and the like. Genetic Algorithms have proved to be very robust when searching in complex spaces both experimentally as well as theoretically. The iterative process of Genetic Algorithm creates new outcomes using the following steps:

- Selection: Select individuals randomly for reproduction with a probability which depends on fitness of others, relatively. This helps in achieving a good chance in selecting the best required individuals.
- Cross-Over: The selected individuals fuse to generate an offspring. Recombination and mutation techniques can be used to generate new chromosomes.
- Evaluation: The fitness of the generated chromosomes is computed.
- Mutation:The initially selected individuals are discarded by the new ones. Thus, the iteration begins to generate and evolve more off-springs.

We randomly select a cross site and swap the genes (bits in our case) of two parent chromosomes to generate two new offspring chromosomes. For example:

Offspring1=10011011<u>11010011</u>	Chromosome1=10011011<u>10010010</u>
Offspring2=01101010<u>10010010</u>	Chromosome2=01101010<u>11010011</u>

Cross-over is followed by Mutation.

We randomly select a gene and complement on that position. For instance, in a binary string 0 will be converted to 1 and 1 will be converted to 0. Mutation is required to generate even more possibilities in the Genetic Algorithm. Let us assume that we mutate the 5th & 6th position of Offspring 1 and Offspring 2 respectively. We obtain the following:

Mutated Offspring1= 1001**0**01111010011
Mutated Offspring2= 01101**1**1010010010

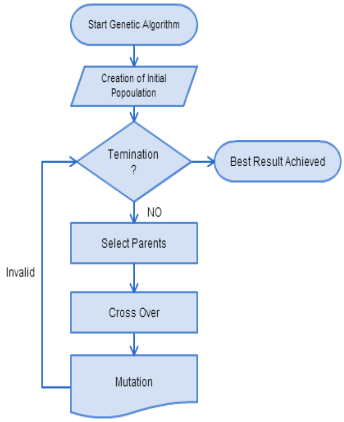


Fig. 1 Life Cycle of Genetic Algorithm

3.6 Regression

Regression Analysis involves a single predictor variable, *x* and a response variable, *y*. It is the form of:

$$y = b + wx \tag{5}$$

where we assume constant variance of *y* and *b* (Y-intercept) & *w* (Slope of the Line) are regression coefficients.

The regression coefficients, *w* and *b* can be analogized as weights :

$$y = w_0 + w_1 x \tag{6}$$

We can solve for the coefficients by estimating the best-fitting straight line as the one which minimizes error between estimate of the line & actual data. Let *D* be a training set. It will consist of predictor variable values, *x*, for some population and corresponding values of response variable,*y*. The training set contains |D| data points of the form $\{(x_1, y_1), (x_2, y_2), ..., (x_{|D|}, y_{|D|})\}$. The regression coefficients are estimated using:

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{|D|}(x_i - \bar{x})} \tag{7}$$

$$w_0 = \bar{y} - w_1 \bar{x} \tag{8}$$

where $\bar{x}$ and $\bar{y}$ are averages of *x* and *y* respectively. w_0 and w_1 offer good estimation of complex regression equations.

Multiple Linear Regression is an extension of straight-line regression. Here, we involve multiple predictor variables. Thus we can model response variable y as a linear function of n predictor variables $\{A_1, A_2, ..., A_n\}$, which describe a tuple, X i.e X = $\{x_1, x_2, ..., x_n\}$. The training data set, D. consists of data of the form $\{(X_1, y_1), (X_2, y_2), ..., (X_{|D|}, y_{|D|})\}$ where X_i are n - dimension tuples. Example of two predictor attributes, A_1 and A_2, is:

$$y = w_0 + w_1 x_1 + w_2 x_2 \tag{9}$$

where x_1 and x_2 attribute values A_1 and A_2, respectively in **X**.

4 Case Study

4.1 MVP Calculation and It's Dynamic Characteristics

We introduce a parameter of comparison among players which may help the team management to purchase the best set of players during an auction, called MVP value. It takes into account attributes relating to player statistics(PBT, PBW & PEX) and team management requirements (BARP, BORP & ERP). The method of calculation of MVP value is given in Algorithm 1. For fair comparison, MVP value is calculated differently for Batsman, Bowlers and All-Rounders. It is done in such a way that for a batsman, batting points(PBT) is given more preference than bowling points(PBW). Similarly MVP value calculation is done for Bowlers and All-Rounders. Also there may be different requirement of skills in different teams, so the method consider these requirement in form of points(BARP, BORP & ERP). The method compares players taking account of the these points. Hence, every player has different MVP values for different teams.

The method adapts to changing scenario in the auction. For instance, when Aaron Finch is purchased by Mumbai Indians, the method deducts the performance attributes points (PBT, PBW & PEX) of Aaron Finch from the require-

Table 1 MVP Points of Players Before Auction

Team Name				CSK	MI	SRH	DD	RCB	RR	KKR	KXIP
Fund				50,000,000	**100,000,000**	208,500,000	400,000,000	210,000,000	130,000,000	130,000,000	118,000,000
Batting Requirement				13	**22**	46	72	24	17	14	9
Bowling Requirement				10	**19**	42	69	34	23	21	20
Experience Requirement				9	**16**	43	59	21	20	13	15
Name	**PBT**	**PBW**	**PEX**	**MVP**	**MVP**	**MVP**	**MVP**	**MVP**	**MVP**	**MVP**	**MVP**
Yuvraj Singh	6.22	6.20	7.06	2.2067	**2.2080**	2.1088	2.1783	2.2398	2.0987	2.2295	2.0835
Dinesh Kartik	5.90	0.00	8.91	2.1667	**2.0705**	1.9486	1.9608	1.6697	1.6333	1.6170	1.2685
Kevin Pieterson	6.74	5.83	2.69	2.0888	**2.0986**	1.9814	2.0697	2.1732	1.9950	2.1643	2.0062
Hashim Amla	6.03	0.00	6.22	2.1344	**2.0363**	1.8979	1.9199	1.6307	1.5739	1.5753	1.1986
Mike Hussey	6.52	0.00	4.62	2.2493	**2.1432**	1.9836	2.0144	1.7077	1.6322	1.6467	1.2246
Aaron Finch	5.86	0.00	3.03	1.9886	**1.8934**	1.7446	1.7760	1.5039	1.4284	1.4485	1.0615
Chris Morris	6.57	6.53	6.22	2.2933	**2.2950**	2.1845	2.2620	2.3311	2.1738	2.3196	2.1574
Kane Williamson	6.50	5.99	6.22	2.0569	**2.0543**	1.9580	2.0226	2.0681	1.9364	2.0564	1.9082
Irfan Pathan	5.55	5.61	8.24	2.0388	**2.0392**	1.9586	2.0145	2.0618	1.9481	2.0530	1.9332

Table 2 Dynamic MVP Value of Players After Finch's Selection

Team Name				CSK	MI	SRH	DD	RCB	RR	KKR	KXIP
Fund				50,000,000	**100,000,000**	208,500,000	400,000,000	210,000,000	130,000,000	130,000,000	118,000,000
Batting Requirement				13	**22**	46	72	24	17	14	9
Bowling Requirement				10	**19**	42	69	34	23	21	20
Experience Requirement				9	**16**	43	59	21	20	13	15
Name	**PBT**	**PBW**	**PEX**	**MVP**	**MVP**	**MVP**	**MVP**	**MVP**	**MVP**	**MVP**	**MVP**
Yuvraj Singh	6.22	6.20	7.06	2.2067	**2.2317**	2.1088	2.1783	2.2398	2.0987	2.2295	2.0835
Dinesh Kartik	5.90	0.00	8.91	2.1667	**1.8225**	1.9486	1.9608	1.6697	1.6333	1.6170	1.2685
Kevin Pieterson	6.74	5.83	2.69	2.0888	**2.1499**	1.9814	2.0697	2.1732	1.9950	2.1643	2.0062
Hashim Amla	6.03	0.00	6.22	2.1344	**1.7858**	1.8979	1.9199	1.6307	1.5739	1.5753	1.1986
Mike Hussey	6.52	0.00	4.62	2.2493	**1.8747**	1.9836	2.0144	1.7077	1.6322	1.6467	1.2246
Chris Morris	6.57	6.53	6.22	2.2933	**2.3218**	2.1845	2.2620	2.3311	2.1738	2.3196	2.1574
Kane Williamson	6.50	5.99	6.22	2.0569	**2.0665**	1.9580	2.0226	2.0681	1.9364	2.0564	1.9082
Irfan Pathan	5.55	5.61	8.24	2.0388	**2.0564**	1.9586	2.0145	2.0618	1.9481	2.0530	1.9332

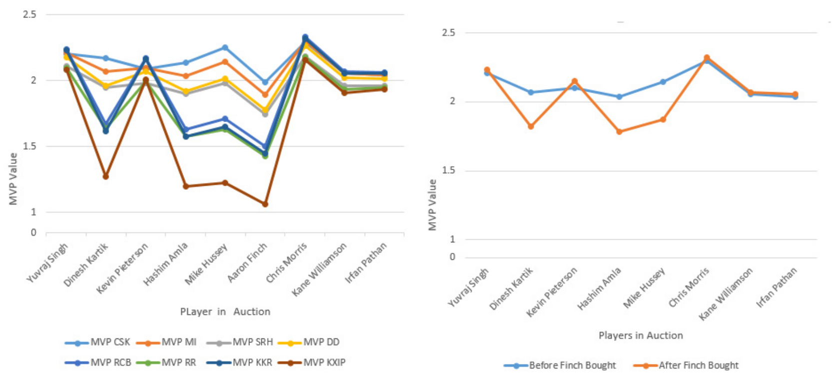


Fig. 2 Different Perspectives of MVP

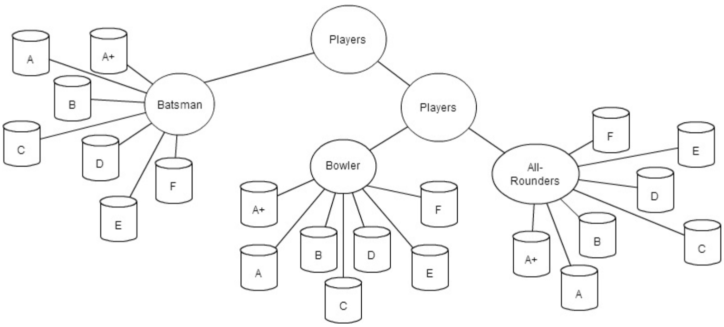
ment points of Mumbai Indians. This deduction triggers the re-computation of MVP values as requirement points has changed. Since Finch was Batsman as he has PBT=5.86, PBW=0 and PEX=3.03, so deduction is more from the Batting Requirement Points(BARP). After re-computation, MVP value of Batsman such as Mike Hussey will decrease whereas Bowlers like Irfan Pathan will increase. The comparison before re-computation and after re-computation is shown in Figure 2.

4.2 Classification of Players Using Decision Tree

Every Player can be classified by the help of Decision Tree and parameters like batting points(PBT), bowling points(PBW) and MVP value. Our method classifies into batting, bowling and all-rounder based on PBT and PBW.

Table 3 Calculation of TCP and Classification of Players Using Decision Tree

Name	Mat	Runs	HS	Avg	SR	100	50	Balls	Runs	Wkts	Avg	Eco	SR	4W	5W	PBT	PBW	PEX	TCP	Type	Grade
Rohit Sharma	112	2903	109*	32.25	129.59	1	21	332	440	15	29.33	7.95	22.13	1	0	6.67	5.71	9.41	6.51	ALL ROUNDER	A+
Lasith Mailinga	83	75	17	4.68	87.2	0	0	1929	2102	119	17.66	6.53	16.21	3	1	3.63	7.57	6.97	6.72	BOWLER	A+
Kieron Pollard	77	1332	78	27.18	144.31	0	6	1076	1539	53	29.03	8.58	20.3	1	0	6.71	5.53	6.47	6.15	ALL ROUNDER	A
Ambati Rayudu	81	1710	81*	26.71	125.18	0	10	18	22	0	0	7.33	0	0	0	6.01	0.00	6.81	5.49	BATSMAN	C
Harbhajan Singh	96	574	49*	15.51	147.93	0	0	2037	2281	92	24.79	6.71	22.14	1	1	6.38	6.60	8.07	6.65	ALL ROUNDER	A+
Corey Anderson	12	265	95*	29.44	146.4	0	1	108	184	4	46	10.22	27	0	0	6.76	4.33	1.01	5.09	ALL ROUNDER	C
Aditya Tare	27	299	59*	17.58	137.15	0	1	0	0	0	0	0	0	0	0	6.03	0.00	22.27	5.05	BATSMAN	C
Jaspreet Bumbrah	13	1	1*	1	33.33	0	0	280	371	7	46.37	7.95	35	0	0	1.36	4.99	1.09	3.88	BOWLER	F
Josh Hazlewood	27	11	6*	5.5	47.82	0	0	623	787	36	21.86	7.57	17.5	1	0	2.08	6.50	2.27	5.19	BOWLER	C
Merchant de Lange	30	47	9*	11.75	134.28	0	0	638	920	38	24.21	8.65	16.7	1	0	5.72	5.91	2.52	5.49	ALL ROUNDER	C
Pawan Suyal	9	2	0	1	10	0	0	174	240	8	30	8.27	21.7	0	0	0.43	5.55	0.76	4.05	BOWLER	E
Shreyas Gopal	6	35	24	17.5	159.09	0	0	96	142	6	23.66	8.87	16	0	0	6.89	0.00	0.50	5.56	BATSMAN	B
Lendl Simmons	34	761	77	25.36	113.92	0	4	36	55	6	9.16	9.16	6	1	0	5.40	0.00	2.86	4.60	BATSMAN	D
Aaron Finch	36	888	88*	26.11	123.84	0	6	45	67	1	67	9.34	43	0	0	5.86	0.00	3.03	4.99	BATSMAN	D
Pragyan Ojha	91	17	4	1.41	37.77	0	0	1887	2309	89	25.94	7.34	21.2	0	0	1.55	6.19	7.65	5.41	BOWLER	C
McClenaghan	46	53	13*	13.25	128.25	0	0	967	1262	56	22.53	7.83	17.2	0	1	5.33	6.34	3.87	5.64	ALL ROUNDER	B
Unmukt Chand	38	731	125	20.88	114.04	2	1	0	0	0	0	0	0	0	0	5.41	0.00	3.19	4.65	BATSMAN	D
Vinay Kumar	86	291	26*	11.19	113.22	0	0	1765	2438	91	26.79	8.28	19.39	1	0	4.86	5.78	7.23	5.51	ALL ROUNDER	B
Parthiv Patel	79	1411	61	20.75	109.37	0	5	0	0	0	0	0	0	0	0	5.10	0.00	6.64	4.74	BATSMAN	D
Aiden Blizzard	81	1724	89	25.35	135.74	0	8	0	0	0	0	0	0	0	0	6.35	0.00	6.81	5.76	BATSMAN	B



Grade	Range
A+	>= 6.5
A	6.0 - 6.5
B	5.5 - 6.0
C	5.0 - 5.5
D	4.5 - 5.0
E	4.0 - 4.5
F	< 4.0

Fig. 3 Graphical Representation of Decision Tree

Lower level of classification is assigning grades to each type of player in the dataset based on the MVP values calculated based on past performance.

4.3 Bowlers and All-Rounders Using BiPartite Cover

Out of the total number of players who can bowl, we use the concept of bipartite cover to find out the best combination of 4 bowlers + 2 all-rounders who can collectively bowl all 20 overs, as demonstrated below:

Consider Fig. 4 with two sets (Players : A to L & Overs : 1 - 20)

{Players} = { SG, LM, KP, CA, HS, JB, VK, JH, MMC, MARCHANT, PS, OJHA }
{Overs} = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 }

We have to determine a subset {Players'} such that it covers the set {Overs}

Now, let each of the players contain the following values:

Shreyas Gopal = {1,3,12,19}	Lasith Malinga= {1,3,10,12,15,18,20}	Kieron Pollard = {6,7,8,14,19}
Cory Anderson = {2,4,5,18}	Harbhajan Singh = {6,9,11,13 }	Jasprit Bumrah = {7,11,14,16}
Vinay Kumar = {3,11,17}	Josh Hazlewood = {7,8,9,13}	Mitchell McClenaghan = {11,16}
Marchant de Lange = {5,13,16}	Pawan Suyal = {14,19}	Pragyan Ojha = {8,9,13,15}

We thus obtain the bipartite graph as follows:

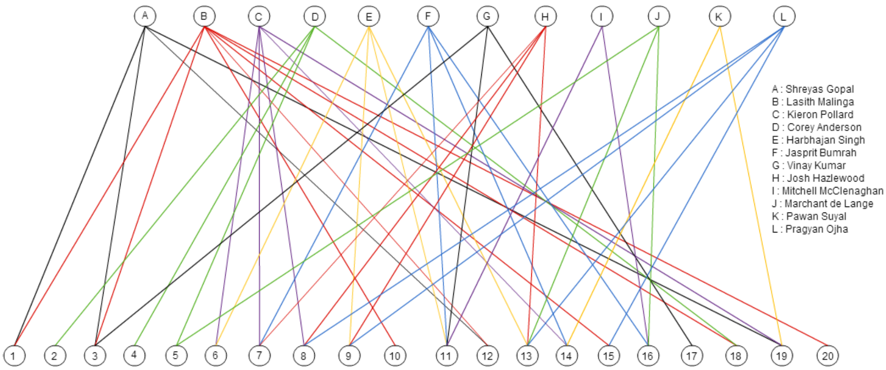


Fig. 4 Bipartite Graph Showing the Mapping of Bowlers & Overs

Initially our selected bowlers set, S = {φ} In the each iteration, we select the player covering the maximum uncovered overs. Initially, all overs are uncovered. In the first iteration, we select LM with a count of 7 uncovered overs and thus S = {1, 3, 10, 12, 15, 18, 20}. We mark {1, 3, 10, 12, 15, 18, 20} as covered overs. In the second iteration, we select KP with a maximum count of 5 uncovered overs and thus S = { 1, 3, 6, 7, 8, 10, 12, 14, 15, 18, 19, 20 }. We mark { 6, 7, 8, 14, 19 } as covered overs. On further calculation, we obtain the set of { LM, KP, CA, HS, JB, VK } as the most optimal solution for bowling selection which ensures possible combinations across all 20 overs. S = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20 }.

In Fig. 4, we have shown the relationship of players along with their best bowling performance in corresponding overs. Table 4 depicts each step during the selection and construction of solution from bipartite graph. A selected player, is marked with SELECT and the un-selected players are mentioned as the number of uncovered overs which the player can bowl.

Table 4 Stepwise Selection Process of Bowlers in Each Iteration

Player	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Initially Uncovered	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6
SG	Y		Y									Y							Y		4	1	0	0	0	0	0
LM	Y		Y							Y		Y			Y			Y		Y	7	SELECT	SELECT	SELECT	SELECT	SELECT	SELECT
KP						Y	Y	Y						Y					Y		5	5	SELECT	SELECT	SELECT	SELECT	SELECT
CA		Y		Y	Y													Y			4	3	3	SELECT	SELECT	SELECT	SELECT
HS						Y			Y		Y		Y								4	4	3	3	SELECT	SELECT	SELECT
JB							Y				Y			Y		Y					4	4	2	2	1	SELECT	SELECT
VK			Y								Y						Y				3	2	2	2	1	1	SELECT
JH							Y	Y	Y				Y								4	4	2	2	0	0	0
MMC										Y				Y							2	2	2	2	1	0	0
MDL					Y								Y			Y					3	3	3	2	1	0	0
PS														Y					Y		2	2	0	0	0	0	0
OJHA								Y	Y				Y		Y						4	3	2	2	0	0	0

Thus, Lasith Malina, Kieron Pollard, Corey Anderson, Harbhajan Singh, Jaspreet Bumrah, Vinay Kumar are selected.

4.4 Variance Analysis

After selecting best suitable bowlers, we are analyzing the similarity and dissimilarity among batsmen by using co-variance analysis. Here we are finding the similarity between Ambati Rayudu and Rohit Sharma. The individual scores for the last fifteen matches of IPL - 2014 are considered.

Table 5 Similarity measure among Rohit Sharma & Ambati Rayudu

Innings	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
A(Rayudu)	48	95	1	14	35	16	9	59	68	33	17	2	30	2
B(Rohit)	27	2	50	4	1	39	59	9	14	51	18	30	16	20
A*B	1296	190	50	56	35	624	531	531	952	1683	306	60	480	40

Average(A) = 30.6428 Average(B) = 24.2857 Sum of **A*B** = 6834
Average of A * Average of B = 744.1837 Sum/14 = 488.1429

Since **(Sum/14) - (Average of A * Average of B) = -256.0408** which is less than 0, Hence Negatively Correlated

Next, we find the similarity between Aditya Tare and Parthiv Patel.

Table 6 Similarity measure among Aditya Tare & Parthiv Patel

Innings	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A(Taare)	24	17	23	8	7	16	9	2	14	6
B(Parthiv)	37	57	21	2	3	26	13	29	10	4
A*B	888	969	483	16	21	416	117	58	140	24

Average(A) = 26.2 Average(B) = 20.2 Sum of **A*B** = 3132
Average of A * Average of B = 254.52 Sum/14 = 313.2

Since **(Sum/14) - (Average of A * Average of B) = 58.68** which is greater than 0, Hence Positively Correlated

Similarly, we obtain a positive correlation among Lendl Simmons & Aaron Finch, and also between Aidn Blizzard & Unmukt Chand. Since Rayudu and Rohit are negatively correlated, hence both players are advised to be in the team. Similarly, Tare & Parthiv, Simmons & Finch and Blizzard & Unmukt are positively correlated, hence either one of the player's among above pairs are advised to be inducted into the team. Since, both players in a pair are similar, therefore there is no drastic effect on the team's performance.

4.5 Selection of Batsmen Using Genetic Algorithm

The selection of batsmen using Genetic Algorithm can be done with the following strategy. We have devised an encoding scheme to achieve selection of batsmen. The encoding bits are allotted in the following manner:

- First and second bits are allotted to the opening batsmen selection.
- Fourth and fifth bits are allotted to the middle order batsmen selection.
- Seventh and eighth bits are allotted to the wicketkeeper selection.
- Tenth and eleventh bits are allotted to the sloggers batsmen selection.

Table 7 Initial Encoding Scheme for Batsman Selection

0	**1**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**
Opener			Middle Order			Wicketkeeper			Slogger		

Type of Batsman	Name of Batsman	Genetic Code
Opener	Simmons	10
Opener	Finch	01
Middle-Order	Rohit	11
Middle-Order	Rayudu	11
Wicketkeeper	Tare	01
Wicketkeepre	Parthiv	10
Slogger	Blizzard	11
Slogger	Unmukt	10

Now, we select twelve random chromosomes (12-bit) for the selection of batsmen as per the encoding scheme. For each 12-bit chromosome, we count the number of *1's* in the chromosome and the count is called **Fitness of the Chromosome**. After calculating the fitness for each chromosome, we determine the average fitness i.e =7 (in our case).

Table 8 Initial Chromosome Selection & Fitness Evaluation for Batsman

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Bit 9	Bit 10	Bit 11	Fitness	Selected (Y/N)
0	1	0	1	1	1	0	1	0	1	1	1	8	Y
1	1	0	1	0	1	1	1	1	0	0	1	8	Y
0	0	0	1	0	1	1	0	0	0	1	1	5	N
1	0	1	1	0	1	1	0	1	1	1	0	8	Y
0	0	0	1	1	1	0	1	1	0	1	0	6	N
0	1	1	0	1	1	0	1	0	0	1	0	6	N
1	1	1	0	0	1	1	1	1	1	0	0	8	Y
0	1	0	1	1	1	0	1	1	1	0	1	8	Y
1	1	1	1	0	0	0	0	1	1	0	1	7	Y
0	1	1	0	1	1	0	1	1	0	0	0	6	N
1	1	0	1	1	0	0	1	1	0	1	0	7	Y
1	0	1	1	0	1	1	1	0	0	0	1	7	Y

$$\text{Average Fitness} = 84/12 = 7$$

We find those chromosomes which have a fitness value $\geq$ average fitness. So, out of the twelve chromosomes, eight are selected. Now, we take two chromosomes in a pair and randomly decide the cross-over site. we apply cross-over on the chromosome pairs. After the cross-over result, we randomly decide the mutation site for individual resultant chromosomes. After mutation, we obtain the desired sets of offsprings. We select the valid off-spring among the resultant off-spring obtain the batsmen for the team.

Table 9 Intermediate Steps of Crossover and Mutation For Batsman Selection

S.No	Selected Chromosomes	Cross-Over Sites	Result of Cross-Over	Mutation Sites	Result
1	010101101011	7	010101111001	8	010101101001
2	110101111001	7	110101101011	9	110101100011
3	101101101110	6	101101111100	2	111101111100
4	111001111100	6	111001101110	3	110001101110
5	**010111011101**	**8**	**010111011101**	**11**	**010111011111**
6	111100001101	8	111100001101	4	111000001101
7	110110011010	5	110111110001	6	110110110001
8	101101110001	5	101100011010	10	101100011110

All combinations obtained from the genetic algorithm fails to give the desired selection of players (at least one opener batsman, two middle-order batsman, one wicketkeeper and one slogger) except the fifth result, which is 010111011111. This result gives:

Table 10 Implication of Genetic Algorithm

01	11	01	11
Finch	Rohit & Rayudu	Tare	Blizzard

After the selection of batsmen, the team needs to decide the batting order for the day's play. For this, the encoding scheme is given in Table 11, the chromosome fitness selection is done in Table 12 and cross-over & mutation of the chromosome to the resultant offspring is done in Table 13. Using these genetic operations, we get the batting order shown in Table 14.

Table 11 Encoding Scheme for Selection of Batting Order

0 1	0 1	1 1	0 1
Opener	Middle Order	Slogger	Tail Enders

Type of Batsman	Name of Batsman	Genetic Code
Opener	Finch - Tare	0
Opener	Tare - Finch	1
Middle-Order	Anderson - Rohit - Rayudu	00
Middle-Order	Rohit - Anderson - Rayudu	01
Middle-Order	Rohit - Rayudu - Anderson	10
Middle-Order	Rayudu - Rohit - Anderson	11
Slogger	Pollard - Blizzard	0
Slogger	Blizzard - Polllard	1
Tail-Enders	Harbhajan - Malinga - Bumrah - Vinay	0
Tail-Enders	Harbhajan - Malinga - Vinay - Bumrah	1

Table 12 Chromosome Selection and Fitness Evaluation

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Fitness	Selected (Y/N)
0	1	0	1	1	1	0	1	5	Y
1	1	0	1	0	1	1	1	6	Y
0	0	0	1	0	1	1	1	4	N
1	0	1	1	0	1	1	0	5	Y
0	0	0	1	1	1	0	1	4	N
0	1	1	0	1	1	0	1	5	Y
1	1	1	0	0	1	1	1	6	Y
0	1	0	1	1	1	0	1	5	Y

Table 13 Intermediate Steps of Crossover and Mutation of Batting Order

S.No	Selected Chromosomes	Cross-over Sites [0 - 8]	Result of Crossover	Mutation Sites	Result
1	**01011101**	**6**	**01011111**	**5**	**01010111**
2	11010111	6	11010101	4	11000101
3	10110110	4	10111101	4	10101101
4	01101101	4	01100110	2	00100110
5	11100111	3	11111101	3	11011101
6	01011101	3	01000111	6	01000011

Table 14 Selection of Batting Order

Batting Number	1	2	3	4	5	6	7	8	9	10	11
Player	Finch	Tare	Rohit	Anderson	Rayudu	Blizzard	Pollard	Harbhajan	Malinga	V Kumar	J Bumrah

4.6 Linear Regression

We have taken a dataset comprising of runs & strike-rate of *Rohit Sharma* in the IPL-2014 matches and the corresponding total score of *Mumbai Indians*. Refer to Table 15.

Table 15 Multiple Linear Regression Analysis for Rohit Sharma

Match	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
(X) StrikeRate (S/R)	**135**	**40**	**121.95**	**80**	**20**	**114.7**	**168.57**	**100**	**233.33**	**113.33**	**210.52**	**90**	**142.85**	**145.45**	**125**
(Y) Runs	**27**	**2**	**50**	**4**	**1**	**39**	**59**	**19**	**14**	**51**	**40**	**18**	**30**	**16**	**20**
(Z) Total Score	**122**	**115**	**141**	**125**	**157**	**170**	**187**	**157**	**160**	**141**	**178**	**159**	**173**	**195**	**173**
(R) S/R - S/R Avg	12.29	-82.71	-0.76	-42.71	-102.71	-8.01	45.86	-22.71	110.62	-9.38	87.81	-32.71	20.14	22.74	2.29
(S) Runs - Runs Avg	1	-24	24	-22	-25	13	33	-7	-12	25	14	-8	4	-10	-6
(T) Total - Total Avg	-34.87	-41.87	-15.87	-31.87	0.13	13.13	30.13	0.13	3.13	-15.87	21.13	2.13	16.13	38.13	16.13
$(S/R - Avg(S/R))^2$	151.04	6840.94	0.58	1824.14	10549.34	64.16	2103.14	515.74	12236.78	87.98	7710.59	1069.94	405.62	517.11	5.24
$(Runs - Avg(Runs))^2$	1	576	576	484	625	169	1089	49	144	625	196	64	16	100	36
R*T	-428.39	3462.93	12.11	1361	-13.69	-105.24	1381.81	-3.02	346.56	148.88	1855.65	-69.79	324.87	867.03	36.89
S*T	-34.87	1004.78	-380.78	701.05	-3.35	170.74	994.42	-0.94	-37.61	-396.65	295.88	-17.07	64.54	-381.34	-96.80

Average(X) = 122.71　　Average(Y) = 26　　Average(Z) = 156.86

$$\Sigma(R*T) = 9177.76 \qquad \Sigma(S*T) = 1882 \qquad \Sigma(S/R - Avg(S/R))^2 = 44082.38$$

$$\Sigma\ (Runs - Avg(Runs))^2 = 4750$$

$$w_1 = \frac{\Sigma(R*T)}{\Sigma(S/R - Avg(S/R))^2} = 0.208196 \quad \text{and}$$

$$w_2 = \frac{\Sigma(S*T)}{\Sigma(Runs - Avg(Runs))^2} = 0.39621$$

$$w_0 = \text{Average(Z)} - (w_1 * \text{Average(X)}) - (w_2 * \text{Average(Y)}) = 121.01254$$

After the analysis of the above mentioned dataset, we obtain the equation which predicts the total team score based on Rohit Sharma's runs & individual strike-rate in the match. The equation is as follows:

$$TotalScore = w_0 + (w_1 * S/R) + (w_2 * Runs) \tag{10}$$

Suppose Rohit Sharma scores 45 runs at a strike-rate of 153. The predicted score is 172.

5 Conclusion

In the paper, we demonstrate the dynamic changing requirement of a player in the duration of an auction, it results to selection of best balanced team for the team management. When a team have a full set of players, method classify them based on their role and performance using decision tree. This classification helps in choosing the playing XI. Using Bipartite Cover, method tried to find certain set of bowlers and all rounders who can bowl well in all possible overs in the match. This will give more option for the captain to utilize his bowling strength during the match. Also method utilizes the differences and similarity in performance of batsman using variance to constrain the possible accept cases for the output of genetic algorithm. After that method again applies genetic algorithm to decide the batting order of the whole team after considering the constraint from the performance of players at various batting positions. The random selection of batsmen and batting order may give a better results in the ever unpredictable game. The total score prediction's accuracy using regression shows the dependency of a team's batting on an individual player.

References

1. Singh, S., Gupta, S., Gupta, V.: Dynamic bidding strategy for players auction in IPL. International Journal of Sports Science and Engineering **5**(01), 3–16 (2011)
2. Singh, S.: Measuring the performance of teams in the Indian Premier League. American Journal of Operations Research **1**, 180–184 (2011)
3. Kalgotra, P., Sharda, R., Chakraborty, G.: Predictive Modeling in sports leagues: an application in Indian Premier League. In: SAS Global Forum 2013, pp. 019–2013 (2013)
4. Ahmed, F., Jindal, A., Deb, K.: Cricket team selection using evolutionary multi-objective optimization. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) SEMCCO 2011, Part II. LNCS, vol. 7077, pp. 71–78. Springer, Heidelberg (2011)
5. Sonali, B., Shubhasheesh, B.: Auction of players in Indian premier league: the strategic perspective. International Journal of Multidisciplinary Research. ISSN 2231 5780
6. Saikia, H., Bhattacharjee, D.: On classification of all-rounders of the Indian premier league (IPL): A Bayesian approach. Vikalpa **36**(4), 25–40