Machine Learning Based IPL Fantasy Cricket Dream11 Best Team Prediction

Mangesh Balpande
Information Technology
SVKM's Institute of Technology
Dhule, Maharashtra, India
mangesh.balpande111@gmail.com

Kalparatna Mahajan
Information Technology
SVKM's Institute of Technology
Dhule, Maharashtra, India
kalparatna223@gmail.com

Jayesh Bhandarkar
Information Technology
SVKM's Institute of Technology
Dhule, Maharashtra, India
bhandarkarjayesh721@gmail.com

Bhushan Kapadne
Information Technology
SVKM's Institute of Technology
Dhule, Maharashtra, India
bhushan.kapadnee@gmail.com

Gaurav Borse
Information Technology
SVKM's Institute of Technology
Dhule, Maharashtra, India
gauravborse612@gmail.com

2024 International Conference on Emerging Smart Computing and Informatics (ESCI) | 979-8-3503-0661-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/ESCI59607.2024.10497335

***Abstract* — As it's widely recognized, the popularity of the Dream11 platform is steadily increasing, particularly during the Indian Premier League (IPL) season. This platform serves as an exciting avenue for cricket enthusiasts to engage in predictive team selection, offering opportunities to earn rewards. However, many users, including those with limited platform knowledge, participate in creating virtual teams, often resulting in losses, with occasional wins attributed to luck. Fortunately, a solution rooted in machine learning aims to address this challenge by creating a model capable of predicting Dream11 fantasy cricket teams for IPL matches. This prediction relies on a multitude of influential factors affecting match outcomes. In the development of this solution, machine learning algorithms such as XGBoost, CatBoost, and Random Forest, all belonging to the category of supervised gradient boosting algorithms, played a pivotal role. Furthermore, we incorporated considerations of historical records and live updates, with ESPN CricInfo serving as the primary source for real time match information.**

Keywords — Dream 11, IPL, Machine Learning, XGBoost, CatBoost, Random Forest, ESPN CricInfo, team statistics, historical data, sports analytics.

I. INTRODUCTION

In this 21st Century along with Football, Cricket is also being one of the popular game in the world. It is second most watched game on television. Hence the fantasy cricket platform came into existence. This is online gaming platform which allows sports enthusiasts to create a virtual team from players of both team on the real-life basis for the particular match which is going to happen, but yet to start. And for this most popular platform in India is Dream 11.

The Indian Premier League (IPL) stands as the worlds most renowned cricket franchise league, commencing in 2008 under the administration of the Indian cricket governing body, commonly referred to as the Board of Control for Cricket in India. Every year, from March to May, the IPL season unfolds, continuously drawing in a growing number of enthusiasts. This surge in interest has led to an increasing fondness for Fantasy Cricket platforms like Dream11 and others. Dream11, for instance, enables users to craft a virtual cricket team comprising 11 players selected from both participating teams. The platform enforces specific rules and conditions, necessitating the selection of a total of 11 players while adhering to a credit cap of 100 credits. A balanced team composition mandates the inclusion of at least one player from each of the four designated categories: Batsman, Bowler, All-rounder, and Wicket Keeper. Moreover, we can take a maximum of 10 players from a single team.

For Creating a best optimised team for dream 11 for an ipl, it needs lots of study and analysis like players past records, Batsman's Performance, Bowler's Performance, both Teams past results, Pitch Conditions, Weather Impact, Batting Bowling averages etc. And it is not possible or easy for us to study and analyse these all things and create proper team for every match. In this paper we proposed the system which predict the team based on players current form, past records against team, pitch conditions, whether updates, playing 11 of both teams, injuries updates and also considering rules for creating and all other factors.

II. LITERATURE SURVEY

In Cricket Selection of Squad for Series is import, because outcomes of match depend upon teams overall performance and strength. In [1] Research paper used Multiple Random Forest Regression and taken such inputs like venues of the match, opponent teams for selecting cricket team for ODI format in different tours based on the attributes of Batsman and Bowler. Algorithms like Decision Tree, Random Forest and Support Vector Regression.

In [2] created a Machine Learning model to predict the efficient players for an IPL Tournament who can be good performers for the team. Here Random Forest algorithm is used in which uses majority average for regression and votes for classification to create decision tree from different samples. This algorithm is able to handle data which contains categorical and continuous variables. This effort aims to reduce manual work which can be tedious. Random Forest algorithm records data from selected data set and decision tree is created at each stage which contains result at the end of each tree. This algorithm gives accurate and more understandable result as compared to other algorithms.

In [3] it applied different technique for selection of IPL player. Fuzzy logic is used which gives 5 best players for each position by analyzing and verifying the data. This reduces the work of analyzing player's performance by watching videos, studying data and all other techniques. Fuzzy logic approach is mainly based on "Degree of Truth" except usually "True or False (1 or 0)" in which modern computers are depends. In fuzzy logic numbers data input and result are taken in consideration. Fuzzy logic is used for artificial intelligence, natural language processing, image processing etc.

Selecting Optimal playing 11 team for match is also can be challenging sometimes for such situations. In research paper [4] proposed a ML model which helps us and gives proper playing 11 based on player's performance, either pitch conditions suited him or not, are the conditions are ripe according to his batting or bowling. Some of popular algorithms came into act like XG Boost, CatBoost, Random Forest, Linear Regression. Factors taken in consideration like total runs conceded by bowler, Runs scored by Batsman. Prepared different ML model for batsman, bowler and for all-rounder.

In [5] a method for predicting Dream 11 Fantasy Cricket Team for an IPL match. Team predicted with player's performance, whether updates. Team predicted by considering rules for creating team on dream 11 like choosing maximum 7 players from one team also to choose minimum 1 wicket keeper. Some of calculations where required for prediction like to calculate average run need to divide total runs by how many innings played by player. This ML model is proposed in COVID conditions were all matches shifted to UAE.

In [6] author has contributed to build an algorithm aims to analyze batsman's performance and their participation in IPL by collecting player's all type of data like avg. total runs, strike rate etc, 50s, 100s, 4s, 6s etc. Regression model is used here which is statistical test allows to identify independent and dependent variables considered.

Whether the team is going to win or lose the match can be predicted with ML model with the help of Machine Learning algorithms like Gaussian Naive Bayes, Support Vector Machine, K- Nearest Neighbour as well as Random Forest. Paper [7] mentioned the ML model and analysed result of predicted result which was average 80%. In order to achieve the most optimized match outcome prediction, a thorough analysis of Ball-by-Ball data is conducted, ensuring increased accuracy across various scenarios.

III. Proposed Methodology

As we all know predicting an accurate team for fantasy cricket platform need a lot of study and also live updates about the match such as weather updates, pitch conditions etc. We proposed Model to predict Fantasy team for cricket by considering all possible features that can affect on the match result, player's performance for particular match, pitch conditions, weather updates like how much moisture the air contains, is there a chance of rain, dew factor etc., in this paper we have taken Dream 11 Fantasy Platform into consideration as it is most popular in India.

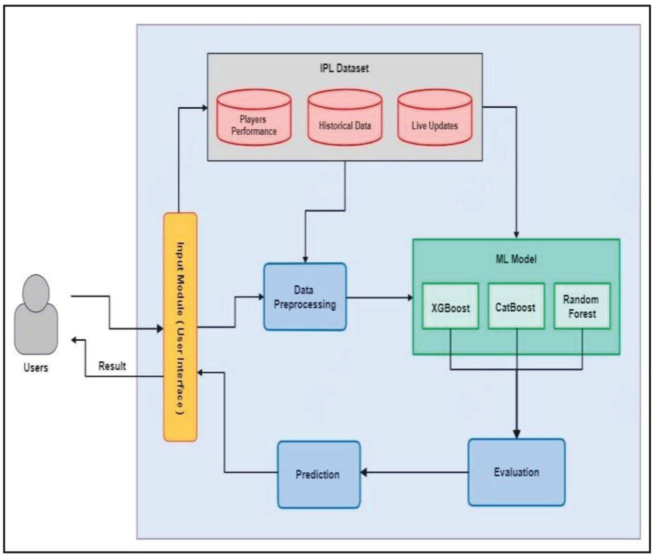

Fig. 1. System Architecture

The project mainly contains python as backend programming language and jupyter for data visualisation. For creating a model user friendly Flask Framework is used.

Project consist of following modules for its implementation:

A. *Input Module*

Fig. 2. Input Module

This module deal with user interference which is created with the help of flask interference and jinja template engine. It is web templates for python programming language & it's similar to Django templates engine just a little bit difference is that it is a text-based template language and therefore can be used for generating and marking up also. Here user will simply enter some details about the match which is going to happen on particular day.

B. Database Module

df

	match_no	Batsman	team	Run	Ball	4s	6s	out_by
0	1	Devon Conway	Chennai Super Kings	1	6	0	0	Mohammed Shami
1	1	Ruturaj Gaikwad	Chennai Super Kings	92	50	4	9	Alzarri Joseph
2	1	Moeen Ali	Chennai Super Kings	23	17	4	1	Rashid Khan
3	1	Ben Stokes	Chennai Super Kings	7	6	1	0	Rashid Khan
4	1	Ambati Rayudu	Chennai Super Kings	12	12	0	1	Joshua Little
...	...	...	...	...	...	...	...	...
1109	70	Shubman Gill	Gujarat Titans	104	52	5	8	Not Out
1110	70	Vijay Shankar	Gujarat Titans	53	35	7	2	Vijaykumar Vyshak
1111	70	Dasun Shanaka	Gujarat Titans	0	3	0	0	Harshal Patel
1112	70	David Miller	Gujarat Titans	6	7	1	0	Mohammed Siraj
1113	70	Rahul Tewatia	Gujarat Titans	4	5	0	0	Not Out

1114 rows × 8 columns

Fig. 3. Past Result Dataset

The dataset used for this research is "IPL ball-by-ball 2008-2023.csv" which contains previous year data of Player Statistics and the "IPL Matches 2008-2023.csv" which contains all the data about the team performance also the live match data used for this research was collected from ESPN CricInfo which provides match data, for the Indian Premier League (IPL). We accessed ball by ball commentary, match statistics and player profiles from 2008 to 2023 IPL seasons. Several factors were taken into account when training machine learning models using the IPL ballby-ball 2008-2023.csv "IPL Matches 2008-2023.csv" and ESPN CricInfo dataset.

- Player statistics: batting average, bowling economy, strike rate, number of fours and sixes hit, number of centuries and centuries scored, etc.
- Team performance: match results between teams and win/loss ratios.
- Pitch conditions: type of pitch and whether it favors batsmen or bowlers.
- Weather conditions on match day.
- Recent form of players in their five matches.

To ensure accuracy with last minute player changes we also scraped up to date playing elevens and live commentary during matches from ESPN CricInfo. For training purpose, we divided the data into two sets.

- Training data: Match data from IPL seasons 2008 to 2022 (80%) was utilized to train our models.
- Testing data: Data from the 2023 IPL (20%) season was kept aside as a testing set.

We used this testing set to simulate how well the models would perform on data that they haven't seen before. The final models were chosen based on how they performed on this testing set.

C. Machine Learning Module

The Machine Learning module predict the points based on players past performance in the match. Those top two players who has most points to be our captain and vice-captain. We store this result into excel file with csv. We also need to clean the data i.e., to correct inaccurate details like player previous name and current name, new added teams, data about new players, retired players, who didn't involve in batting or bowling may have to null value in dataset and these needed to be handled, also some of details data set not contains like batsman strike rate, bowling economy, batsman average, stumping, run concede to bowler, no of 6s ,4s, 50s, 100s. To calculate these above factors, we used following calculations.

- Average runs scored by the Batsman = Total Runs / No. of Innings played by batsman.
- Bowling Economy = Total Runs concede / Total overs bowled
- Strike Rate = (Runs scored / Ball faced) * 100

In Similar way other factors are calculated. And by using this new dataset is generated. After collecting and preparing the data we moved towards training dataset. Here we used XGBoost, CatBoost, Random Forest algorithms.

XGBoost: In [5], this is supervised for learning problems where we used training data with having multiple features say input X_i for prediction of target variable say y_i. This model has mathematical structure, by which prediction is made. Here Predicted Result may have different interpretations depends on task. We can express various task with y_i like Rankings, Classification, Regression. A simple example of y_i is given as $\hat{y}_i = \sum_i \theta_j X_{ij}$. (1)

This is an ensemble method which is a combination of various types of decision trees to create stronger model.

- Loss function: This function quantifies that how well actual target is being matched with model's prediction. For problems like regression problem for classification, cross entropy is often used.
  For regression: $Loss = \sum(y_i - \hat{y}_i)^2$ (2)
  For binary classification:
  $Loss = \sum [y_i * \log(1 + \exp(-\hat{y}_i)) + (1 - y_i) * \log(1 + \exp(\hat{y}_i))]$ (3)

- Decision Tress: In XGBoost each tree makes prediction for input Value.
  Decision Tree Prediction: $F\_t(x) = w\_q$ if $x \in R\_q$
  were,
  $F\_t(x)$ = prediction of the t-th tree for input x,
  q = specific leaf in tree
  w_q = weight associated with that leaf.

- Gradient Boosting: XGBoost iteratively minimizes loss function with addition of decision trees. Every new tree is fitted in negative gradient of the loss with respect to prediction that made from the current ensemble.
  Negative Gradient: $\frac{\partial Loss}{\partial F(x)}$

- Update step: For prevention of overfitting each new decision tree is with learning rate.
  update: $F\_t(x) = F\_(t-1)(x) + \eta * F\_t(x)$ (4)
  were, η = learning rate
  $F\_t(x)$ = prediction of new tree
  $F\_(t-1)(x)$ = prediction of the ensemble up to the (t-1)-th tree.

- Regularization: For prevention overfitting this regularization term is used
  Regularization Term = $\gamma * |q| + 0.5 * \lambda * w\_q^2$ (5)
  were,
  γ and λ = regularization hyperparameters
  $|q|$ = number of data points in leaf q
  w_q = weight of leaf q

- Final Prediction: The Final prediction will be sum of all decision trees in ensemble.
  Final Prediction: $F(x) = \sum F\_t(x)$ (6)

CatBoost: In [5] it is high performance open-source library which is based on Gradient-Boosted decision tree. It is mainly used for problems like regression and classification which contain number of independent features. It can handle categorial and numerical features.

- Objective Function: To find out best model CatBoost minimizes specific objective function
  Objective = $Loss(y_i, \hat{y}_i) + \Omega(f)$ (7)

y_i = true label of the i[th] instance
$\bar{y}_i$ = predicted label by the current ensemble
f = ensemble of trees
$\Omega(f)$ = regularization term
CatBoost combines these all steps and strategies to refine ensemble of trees.

Random Forest: In [1], The Random Forest algorithm was implemented using the Scikit Learn python library with the following parameters:

- Number of trees (n_estimators): 100
- tree depth: No limit (the trees expand until all leaves are pure)
- Minimum samples for splitting: 2
- Minimum samples for a leaf node: 1
- Maximum number of leaf nodes: No limit
- Bootstrap sampling: Enabled

A forest of 100 decision trees was built by training each tree on a bootstrapped subset of the data, with uncapped maximum depth and regularization through a minimum sample split threshold of 2 and a minimum leaf sample threshold of 1. Predictions were aggregated via majority voting for classification and averaging for regression. Bootstrap sampling enhances predictive performance by training on diverse data samples, and uncapped tree depth reduces bias. Random Forest's key advantages include robustness to noise, the ability to model complex relationships, and avoidance of overfitting through ensemble learning. This information can be incorporated into the revised manuscript for improved implementation clarity.

D. Model Selection

Once the model provided us with the projected points earned by each player in the match, need to proceed with necessary Rules for team selection. To achieve this, we need linear programming with pulp (Used to solve linear programming problems) library. And the team selection happened based on dream 11 credit points for each player. Major problem here is that Continuous credit change of credit points for player. Also, selection of Captain (Gives 2x points) and Vice-Captain (1.5x Credit Points). The solution of this was Top 2 player with most Credit points kept as the Captain and Vice-Captain. Python pulp library had main role here. First, we categorised players according to their role such as Batsman, Bowler, All-Rounder, Wicket-Keeper and binary variables for defining them are bt_i, bl_i, all_i and wkt_i respectively. Player were indexed from 0 and ith player performance was denoted by p_i. If the value for particular variable is one then it is true, mean if bl_i is 1 then it is bowler. And if it is bowler the sp_i variable has importance otherwise it has no meaning and he is fast bowler. Dream 11 also put limit while selecting players so we defined variables for that. For Batsman it was bt_{lb} (lower bound) bt_{ub} (upper bound). Also, for other bl_{lb}, bl_{up}, all_{lb}, all_{ub}, wkt_{lb}, wkt_{ub}. T_i was for player choosen from which team. If T_i is 0, then player is from team 2, otherwise from Team 1.

IV. Result Analysis

After receiving the projected points earned by each player for a match from the models, we had to apply for mentioned Rules and Constraints to determine the best Dream11 team. For this purpose, PULP library required to us. The selection of the 11 players was primarily based on two factors: the Dream11 points expected for each player and the credit cost associated with each player. The credit cost was crucial, especially in high-profile matches such as MI vs. CSK. For instance, we might want to include expensive players like Rohit Sharma, Ruturaj Gaikwad, D Conway, Jasprit Bumrah, and MS Dhoni, all of whom cost more than 10 credit points. However, it was essential to ensure that this didn't deplete your credit points budget, leaving you with insufficient credits to assemble a competitive remaining team. Out of the 11 players chosen, our system would designate the top 2 players as the Captain and Vice-Captain, awarding them 2x and 1.5x points, respectively, for the match.

To determine the performing batsman we can utilize a machine learning model that analyzes their performances and predicts the number of runs they are likely to score. By training a machine learning model to estimate the teams score based on batsman's runs we can even forecast match outcomes. Machine learning algorithms enable us to uncover patterns, in the data, such as which batsman tend to complement each other or which bowlers pose challenges, for specific batsman. Armed with this information strategic decisions can be made during matches.

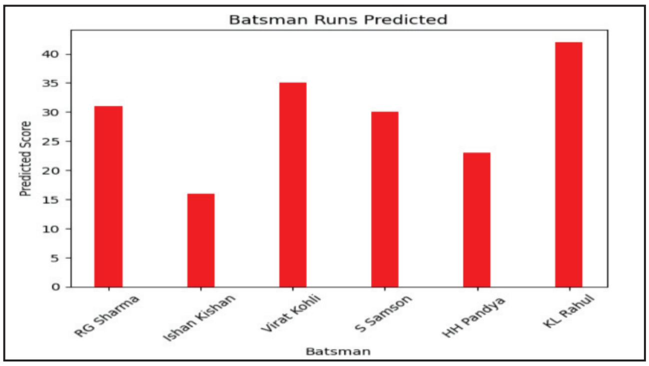

Fig. 4. Batsman Performance Prediction

To determine the performing bowler one can, employ machine learning techniques same as used for batsman performance.

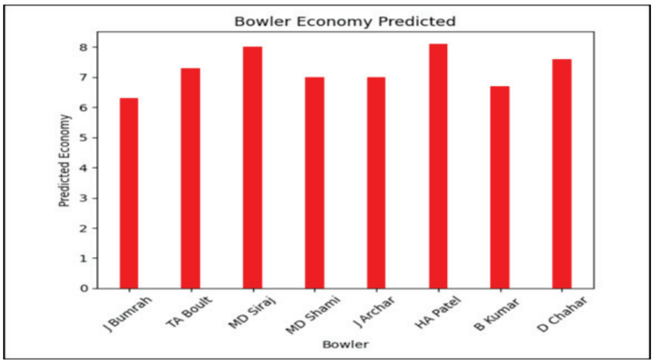

Fig. 5. Bowler Economy Prediction

Additionally, machine learning can unveil patterns within the data highlighting which bowlers excel against batsman or in game scenarios. This valuable insight can then be utilized by coaches and teams to make decisions during matches.

Predicted Team

Name	Role	Team	Cost
R_Sharma	Batsman	Mumbai Indians	9.5
S_Yadab	Batsman	Mumbai Indians	10
R_Gaikwad	Batsman	Chennai Super Kings	9
Ishan Kishan	Wkt-Kepper	Mumbai Indians	8.5
M_ali	All-Rounder	Chennai Super Kings	8.5
J_Bumrah	Bowler	Mumbai Indians	9
M_Theekshna	Bowler	Chennai Super Kings	8
D_Chahar	Bowler	Chennai Super Kings	8.5
D_Conway	Batsman	Chennai Super Kings	8.5
T_David	All-Rounder	Mumbai Indians	8
C_Green	All-Rounder	Mumbai Indians	9.5

Fig. 6. Predicted Result

A. Performance Evaluation

By applying all the constraints, we made our selection of the top 11 players based on their actual Dream11 points earned, evaluating them against the 11 players chosen using the points predicted by our ML models and considering the assigned credits for each player. We assessed the team chosen by the model's performance in two different ways:

1) Reward Estimation : The earned points display a heavily right-skewed distribution, with a significant gap between the top two teams compared to teams lower in rank. This disparity is particularly pronounced in contests with higher entry fees. Our model leveraged these insights to determine rewards for the 11 selected players, focusing on the substantial differences in rewards between top-ranked and second-ranked teams.

2) Error Rate : We assessed the variance in points achieved by our chosen team in comparison to the highest-scoring team while adhering to the imposed constraints. For example, if the team with the maximum Dream11 points earned in a match scored 650 points, and our system-selected team scored 450 points, the error rate would be calculated as (650–450)/650 = 0.307. In this research, the XGBoost, CatBoost, and Random Forest machine learning models were employed to predict players' performance points based on their match data.The training dataset encompassed match records from 2008 to 2023, while the testing dataset exclusively contained data from the year 2023. The assessment of these models was conducted by examining their error rates, which are defined as follows:

$$Error\ Rate = \frac{Actual\ Points - Predicted\ Points}{Actual\_points}$$

MODEL ERROR RATE

Model	XGBoost	CatBoost	Random Forest
Error Rate	17	22	25

Predicted Rewards Earned Over the Entire IPL Season. Conducted a comprehensive evaluation of this system throughout an entire season, employing a training dataset that encompassed IPL seasons spanning from 2008 to 2023. For testing purposes, we utilized the IPL23 dataset. Among the various models considered, XGBoost consistently demonstrated the most promise and reliability across all IPL seasons. XGBoost was selected as Model 1 for team prediction, and an ensemble model, Model 2 was crafted by blending 1/2 XGBoost with 1/6 Moving Average, 1/6 CatBoost, and 1/6 Random Forest to optimize the prediction of top-performing teams.

To operationalize our system, we executed a script to scrape the playing XI teams for each match 15 minutes before the match's start using a cronjob. Then we applied our models to these players to predict the points each player was expected to earn and selected the best 11 players for the Dream11 team for that particular match. This pipeline generated a CSV file that could be used to select the team for the match between MI and CSK in IPL 23. At the end we compared Our Predicted Team with actual Best 11 team for dream 11.

TABLE I. TEAM COMPARISON

Best 11	**Predicted 11**
R_Sharma	R_Sharma
S_Yadav(C)	S_Yadav(VC)
R_Gaikwad	R_Gaikwad
Ishan_Kishan	M_Dhoni
M_Ali(VC)	M_Ali
J_Bumrah	J_Bumrah(C)
M_Theekshna	M_Theekshna
D_Chahar	D_Chahar
D_Conway	D_Conway
T_David	AT_Rayadu
C_Green	C_Green

Based on the results the model has an accuracy rate of around 83% and the error rate of around 17%. Looking at the table we can see that two players deviate from what the model predicted including the captain and vice captain. Out of all the algorithms used for this model XGBoost demonstrated the level of accuracy.

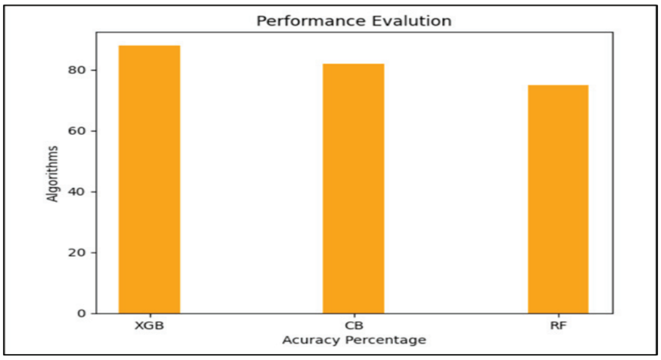

Fig. 7. Algorithm Performance Evaluation

Through the entire process of research, the XGBoost model had the lowest error rate and the maximum consistency in its predictions among the tested models. In comparison to CatBoost and Random Forest, XGBoost is a more precise algorithm. XGBoost's accuracy rate is above 80%, CatBoost's average accuracy rate is above 75%, and Random Forest's accuracy rate is above 70%. The accuracy of each algorithm is comparatively high. As a result, the selection procedure made use of the player points predicted by this model. On other datasets, the algorithms might perform differently. While CatBoost and Random Forest might be more accurate on different datasets, it's possible that XGBoost is more accurate on the one that was used to construct a project.

V. CONCLUSION

In conclusion, the project topic of predicting Dream11 teams for IPL using XGBoost, CatBoost, and Random Forest algorithms based on various factors like player performance, live updates, pitch conditions, past results, and credit points holds great potential in enhancing the fantasy sports experience for users. By leveraging advanced machine learning techniques, this project aims to provide more accurate predictions for assembling winning teams, thereby increasing users engagement and success rates in fantasy sports leagues. The integration of XGBoost, CatBoost, and Random Forest algorithms allows for a comprehensive analysis of player performance and other relevant factors. XGBoost's gradient boosting framework can capture complex relationships between features and target variables, while CatBoost's categorical variable handling and intrinsic feature scaling provide an edge in terms of performance.

REFERENCES

[1] N. Rodrigues, N. Sequeira, S. Rodrigues and V. Shrivastava, "Cricket Squad Analysis Using Multiple Random Forest Regression," 2019 1st International Conference on Advances in Information Technology (ICAIT), Chikmagalur, India, 2019, pp. 104-108, doi: 10.1109/ICAIT47043.2019.8987367.

[2] M. Ramalingam, S. Gokul, L. S. Mythravarshini and K. S. Harine, "Efficient Player Prediction and Suggestion using Machine Learning for IPL Tournament," 2022 International Mobile and Embedded Technology Conference (MECON), Noida, India, 2022, pp. 162-167, doi: 10.1109/MECON53876.2022.9752414.

[3] R. Dutt, T. A. Kusupati, A. Srivastava and D. Hore, "IPL Player Selection using Fuzzy Logic," 2022 IEEE Industrial Electronics and Applications Conference (IEACon), Kuala Lumpur, Malaysia, 2022, pp. 180-184, doi: 10.1109/IEACon55029.2022.9951755.

[4] N. Ravichandran, N. C. P. A and P. Rao Rebala, "Optimal IPL Playing 11 Team Selection," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-6, doi: 10.1109/I2CT57861.2023.10126227.

[5] S. Choudhari, N. Wagholikar, A. Swaminathan and S. Kurhade, "Dream11 IPL Team Recommendation using Machine Learning and Skill-Based Ranking of Players," 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-6, doi: 10.1109/ICONAT53423.2022.9725819.

[6] A. Santra, A. Sinha, P. Saha and A. K. Das, "A Novel Regression based Technique for Batsman Evaluation in the Indian Premier League," 2020 IEEE 1st International Conference for Convergence in Engineering (ICCE), Kolkata, India, 2020, pp. 379-384, doi: 10.1109/ICCE50343.2020.9290569.

[7] D. Thenmozhi, P. Mirunalini, S. M. Jaisakthi, S. Vasudevan, V. Veeramani Kannan and S. Sagubar Sadiq, "MoneyBall - Data Mining on Cricket Dataset," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-5, doi: 10.1109/ICCIDS.2019.8862065.

[8] M. Shetty, S. Rane, C. Pandita and S. Salvi, "Machine learning-based Selection of Optimal sports Team based on the Players Performance," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 1267-1272, doi: 10.1109/ICCES48766.2020.9137891.

[9] E. Mundhe, I. Jain and S. Shah, "Live Cricket Score Prediction Web Application using Machine Learning," 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Pune, India, 2021, pp. 1-6, doi: 10.1109/SMARTGENCON51891.2021.9645855.

[10] S. Priya, A. K. Gupta, A. Dwivedi and A. Prabhakar, "Analysis and Winning Prediction in T20 Cricket using Machine Learning," 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2022, pp. 1-4, doi: 10.1109/ICAECT54875.2022.9807929.

[11] A. Peddii and R. Jain, "Random Forest-Based Fantasy Football Team Selection," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 1822-1826, doi: 10.1109/ICACCS57279.2023.10113019.

[12] G. Kaur and G. Jagdev, "Analyzing and Exploring the Impact of Big Data Analytics in Sports Science," 2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), Rajpura, India, 2020, pp. 218-224, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181320.

[13] G. Bhatambarekar and S. Rai, "Batsmen Performance Prediction Using Regression Models," 2022 International Conference on Artificial Intelligence and Data Engineering (AIDE), Karkala, India, 2022, pp. 80-85, doi: 10.1109/AIDE57180.2022.