Fantasy League Team Recommendation System For Cricket

Mayank Chamarthi *IMT2021073*
Meher Ashish Nori *IMT2021085*
Rishi Nelapati *IMT2021076*
Prasad Jore *IMT2021104*

***Abstract*—This report presents the development and evaluation of a fantasy cricket recommendation system leveraging machine learning models. Using IPL match data, models including Random Forests, Linear Regression, Polynomial Regression, XGBoost, CatBoost, and LSTM were employed to predict player points. Finally we converted it to an optimisation problem to find the optimal eleven.**

1. Introduction

Fantasy sports have gained immense popularity in recent years, offering enthusiasts an immersive experience by allowing them to engage with their favorite sports at a more participatory level. Among these, cricket, with its global appeal and fervent fan base, stands out as a prime contender in the fantasy sports arena.In this context, our project endeavors to enhance the fantasy cricket experience through the development of a recommendation system tailored specifically for cricket and IPL enthusiasts.

The fundamental goal of our system is to empower users with the optimal team composition from the players participating in a cricket match. With the abundance of statistical data available, our system aims to sift through this wealth of information to curate a team that not only maximizes the potential for scoring points but also aligns with the user's strategic preferences and constraints.
Our recommendation system operates on the premise of leveraging advanced data analytics and machine learning techniques to analyze player statistics, historical performances, and other relevant factors. By harnessing the power of data-driven insights, our system endeavors to provide users with actionable recommendations that can serve as a cornerstone for their fantasy league strategies.

By streamlining the team selection process and optimizing player positions, our system aims to enhance user engagement, satisfaction, and ultimately, their success in fantasy cricket platforms such as Dream11, My11Circle, and others.
In the subsequent sections of this report, we delve deeper into the methodology employed in the development of our recommendation system, the datasets utilized for training and evaluation, the results obtained through empirical testing, and finally, draw conclusions regarding the efficacy and implications of our approach in the context of fantasy cricket.

2. Dataset Used

In our project focused on developing a recommendation system for fantasy cricket, we employed a rich and expansive dataset sourced primarily from Kaggle. The dataset in question is the "IPL complete dataset 2008-2020," a comprehensive repository of match data spanning over a decade of Indian Premier League (IPL) cricket.
It encompasses both ball-to-ball data and match-level statistics, providing a holistic view of player performances, match outcomes, and contextual factors influencing gameplay.

In addition to the Kaggle dataset, we explored supplementary avenues for augmenting our dataset with more recent match data. This involved leveraging various Application Programming Interfaces (APIs) and employing web scraping techniques to extract pertinent information from reliable sources.
Kaggle dataset

3. Methodology

3.1. Preprocessing the dataset

In the initial phase of our methodology, we focused on transforming the raw ball-by-ball dataset into a structured scorecard that encapsulated the key performance metrics of players in a cricket match. This scorecard served as the foundation for subsequent analysis and model development.For batsmen, the scorecard included the total runs scored, balls faced during their innings. Conversely, for bowlers, the scorecard documented the number of wickets taken during their bowling spell and runs conceded. Additionally, fielding contributions were accounted for by recording the number of catches taken during the match,

In the development of our fantasy cricket recommendation system, an integral aspect was the alignment of our scoring mechanism with the point-assigning system utilized by popular platforms such as Dream11. To compute the total points accrued by each player in a match, we adopted the scoring mechanism that accounts for the diverse contributions made by batsmen, bowlers, and all-rounders by Dream 11. So we created new features like for batsmen, key metrics such as runs scored, balls faced, boundaries hit, and strike rate were computed to gauge batting proficiency and impact. Similarly, for bowlers, statistics such as wickets taken, runs conceded, economy rate, and bowling variations were captured to assess bowling prowess and effectiveness.

Additionally, fielding contributions such as catches taken, run-outs executed, and stumpings were recorded to account for players' fielding capabilities.

3.2. Approach

Our approach to addressing the challenge of recommending the optimal fantasy cricket team revolves around a two-stage methodology aimed at leveraging predictive modeling and optimization techniques to maximize points accumulation while adhering to constraints imposed by platforms like Dream11.

- **Predictive Modeling for Points Estimation**: In the initial stage of our approach, we focus on predicting the points that each player is likely to score based on their performance in previous k matches, where k is a hyperparameter of the model. To accomplish this, we employ supervised learning techniques, utilizing historical match data to train predictive models. These models are trained to analyze various factors such as batting averages, bowling economy rates, recent form, match conditions, and player-specific attributes to estimate the expected points for each player in upcoming matches. Any supervised learning model, such as regression or ensemble methods, can be employed for this prediction task.
- **Optimization for Team Selection**: Once we have the predicted points for batting, bowling, and fielding performances of all players in consideration, we proceed to the team selection stage. Here, our objective is to assemble a team of 11 players that collectively maximize the total points earned while adhering to the constraints imposed by platforms like Dream11. These constraints typically include limitations on the number of players from each team, player positions (batsman, bowler, all-rounder), and overall budget. This task can be formulated as an optimization problem, where the objective is to maximize the total predicted points subject to the aforementioned constraints. Various optimization techniques, such as integer linear programming or genetic algorithms, can be employed to efficiently solve this problem and identify the optimal team composition.

By adopting this two-stage approach, we aim to provide users with a systematic and data-driven methodology for selecting their fantasy cricket teams. The combination of predictive modeling and optimization enables us to harness the power of historical data while adhering to practical constraints, ultimately empowering users to make informed decisions and maximize their success in fantasy cricket leagues.

3.3. Models Used

Here are the models used:

- **Random Forests**: Random Forests are well-suited for our project due to their ability to handle large datasets with numerous features. They excel in capturing complex relationships between player performance metrics and expected points. Their ensemble nature also mitigates overfitting and provides robust predictions.
- **Linear Regression**: Linear Regression offers simplicity and interpretability, making it suitable for initial exploratory analysis. While less complex compared to other models, it provides a baseline for predicting points based on linear relationships between input features and output points.
- **XGBoost**: XGBoost, an ensemble learning method, is highly effective for our project due to its scalability, speed, and performance. It excels in handling structured data and can capture complex interactions between features, making it ideal for predicting points based on player performances in cricket matches.
- **CatBoost**: CatBoost, another gradient boosting algorithm, is particularly advantageous for handling categorical features present in cricket match data. Its ability to handle categorical variables without preprocessing makes it a valuable addition to our model ensemble, enhancing prediction accuracy.
- **Long Short Term Memory (LSTM)**: LSTM, a type of recurrent neural network (RNN), is well-suited for sequential data such as match-by-match statistics. Its ability to capture temporal dependencies and long-range dependencies makes it promising for predicting player performance.

3.4. Optimisation Problem

Now that that we have the predicted points of each player as a batsman/bowler/all-rounder, we want to select the team with the highest points while keeping in mind the constraints on player selection this boils down to a linear optimization problem.

Variables:

- $x_i^{batsman}$: 1 if player i is selected as a batsman, else 0.
- x_i^{bowler} : 1 if player i is selected as a bowler, else 0.
- $x_i^{allrounder}$: 1 if player i is selected as an all-rounder, else 0.

Maximize:

- $x_i^{batsman}(BattlingPoints_i + FieldingPoints_i) + x_i^{bowler}(BowlingPoints_i + FieldingPoints_i) + x_i^{allrounder}(BattlingPoints_i + BowlingPoints_i + FieldingPoints_i)$

Constraints:

- Each player can be only one of batsman/bowler/all-rounder:

$$x_i^{batsman} + x_i^{bowler} + x_i^{allrounder} \leq 1, \forall i$$

- Total cost cannot exceed the budget:

$$\sum_i (x_i^{batsman} + x_i^{bowler} + x_i^{allrounder}) \cdot cost_i \leq Budget$$

- You can only choose 3-6 batsmen:

$$3 \leq \sum_i x_i^{batsman} \leq 6$$

- You can only choose 3-6 bowlers:

$$3 \leq \sum_i x_i^{bowler} \leq 6$$

- You can only choose 1-4 all-rounders:

$$1 \leq \sum_i x_i^{allrounder} \leq 4$$

- You have to choose 11 players:

$$\sum_i (x_i^{batsman} + x_i^{bowler} + x_i^{allrounder}) = 11$$

4. Results

4.1. Evaluation of Models

Here is the MSE error of all the models used for predicting the points earned by each player. It is evident that LSTM emerged as the top-performing model, demonstrating superior predictive capabilities with respect to Mean Squared Error (MSE) metrics. Following closely behind LSTM, Linear Regression emerged as a formidable contender, delivering competitive results in terms of MSE. Despite its simplicity compared to more complex models, Linear Regression proved to be adept at capturing linear relationships between input features and output points.
CatBoost surpassed XGBoost in predictive accuracy, despite both being gradient boosting algorithms. CatBoost's distinctive ability to handle categorical features without preprocessing likely contributed to its superior performance, highlighting the importance of leveraging specialized algorithms tailored to the nuances of the dataset.

We can see that LSTM performed the best as a model considering the MSE errors. Linear Regression is a close second and CatBoost performed better than XGBoost

Model	Batsman Model MSE Error	Bowler Model MSE Error	Fielder Model MSE Error
Linear Regression	854	764.7	22.64
Random Forests	875.38	796.622	26.92
XGBoost	997.49	886	27.7
CatBoost	889	796	24.83
LSTM	852	772	22.14

Figure 1. The MSE Errors of each model

4.2. Predicted Scores

Here is the predicted scores by all the models on the ten most recent matches in the dataset compared to the best team which got the maximum points.

match_id	actual_best	lstm	lr	catboost
1237181	625	300	310	358
1237180	755	548	504	524
1237178	567	428	413	405
1237177	786	641	605	656
1216495	629	524	402	527
1216505	594	415	378	424
1216506	565	364	299	204
1216530	764	592	595	553
1216535	497	333	377	386
1216502	495	366	413	352

Figure 2. The predicted points earned by the 11 predicted by each model

5. Conclusion

The analysis of the results shows that most of the times our system recommends the team with close to 80% of the best possible score. From our observations in the rankings of platforms like Dream11, this score is sufficient to guarantee a decent rank.

6. Future Work

On analysis of the features we found the features(previous game performances) are weakly correlated to the target, this explains the poor performance of the models. We can use more detailed datasets that would allow us to use other factors like pitch conditions, record against opponent team, etc. to make our predictions.

References

[1] Mirko Stojiljković. *Linear Programming in Python*. Retrieved from https://realpython.com/linear-programming-python/

[2] Dream11. *Dream11 Point System*. Retrieved from https://www.dream11.com/games/point-system