Data-Driven Solution for Predicting Dream11 Fantasy Cricket Teams

Anshi Gandhi - 2024201038, Priyanshu Sharma - 2024201046, Prashant Thakur - 2024202020

May 7, 2025

1 Abstract

This project presents a data-driven solution for predicting optimal Dream11 fantasy cricket teams tailored for One Day International (ODI) matches. Given the strategic and monetary importance of fantasy cricket in India—with over 130 million users engaging in platforms like Dream11—our tool aims to support users in assembling highly competitive teams informed by historical and contextual data. To achieve this, we integrated four rich datasets. We meticulously computed fantasy points for each player in alignment with Dream11's official point system by iterating through ball-by-ball match data. This extensive feature engineering resulted in a final dataset of 126 features across 45,000 rows, representing player-match instances. For the predictive model, we first experimented with a Random Forest Regressor—a robust ensemble method leveraging bagging and bootstrapped decision trees—but observed suboptimal performance due to high feature dimensionality, complex non-linear interactions, and inefficiency to capture temporal trends. Consequently, we transitioned to the XGBoost regression model, renowned for its gradient boosting framework, handling of missing data, and regularization techniques to prevent overfitting.

Our XGBoost model, configured with a max depth of 5, learning rate of 0.001, and 250 estimators, achieved a Mean Absolute Error (MAE) of 390 fantasy points on the validation set. Beyond raw predictions, the model outputs interpretable feature importances and top player recommendations, transparently justifying selections based on recent form, opposition history, and environmental context. This empowers fantasy sports enthusiasts to make data-backed, confident decisions when crafting their Dream11 teams—potentially maximizing their returns.

Keywords : Fantasy Score Prediction, Dream11, XGBoost, Random Forest, Regression, DreamTeam, Cricket, RMSE, Fanatsy Points

2 Introduction

Cricket in India transcends mere entertainment—it is woven into the cultural fabric of over a billion people, with marquee tournaments drawing peak television audiences exceeding 500million viewers per match. Fantasy sports have become increasingly popular worldwide, with the help of technology allowing players to create virtual teams and compete against each other based on real-life player performance. Such platforms require players to analyze a large amount of data, including player statistics and past performances, to create a winning team. Previous research has explored the use of machine learning algorithms for sports analytics, including team selection in fantasy sports. These studies have shown the potential of machine learning in predicting player performance, identifying data patterns, and enhancing team selection accuracy. This fervor has given rise to a thriving fantasy sports ecosystem, where more than 130million fans participate on platforms like Dream11, transforming their cricketing acumen into real-world rewards. In Dream11, users draft teams of eleven players before each match; those whose selections excel on the field—for example, a batsman scoring a century or a bowler taking multiple wickets—accumulate fantasy points and stand to win substantial cash prizes. Such platforms democratize sports betting by substituting traditional wagers

with skill-based prediction, yet they also introduce complexity and uncertainty: picking a Dream11 squad is as much an art of understanding player form, pitch conditions, and opponent matchups as it is about gut instinct.

Our project addresses these challenges head-on by offering a data-driven fantasy cricket team prediction model that elevates Dream11 strategy from intuition to evidence. By aggregating and harmonizing three specialized datasets—detailed match-by-match performance metrics, direct player-versus-player histories, and weather-condition impacts—we craft a richly textured feature space that captures the nuances of on-field dynamics. We then apply advanced feature engineering techniques, such as exponentially weighted moving averages to emphasize recent form and venue-specific performance aggregations to account for pitch behavior. These features feed into a meticulously tuned XGBoost regression model, whose predictions are further demystified via SHAP interpretability, enabling users not only to see which players score highest but also why they do so.

This end-to-end solution—the Dream11 Cricket Prediction Project—therefore stands apart in its novelty and rigor. It delivers unbiased, player-agnostic predictions based purely on historical statistics, ensuring fairness regardless of star status or name recognition. By coupling predictive power with transparent explanations, our tool offers fantasy cricket enthusiasts a decisive analytical advantage, empowering them to construct Dream11 rosters that are both strategically sound and grounded in objective data.

3 Related Work

A number of recent studies have explored machine learning techniques for fantasy cricket team prediction, focusing on various aspects such as feature engineering, model selection, and interpretability.

Early work by S. S.etal. [1] adopted a data science–centered pipeline for Dream11 team forecasting, highlighting the use of traditional statistical features (runs, wickets, strike rates) and simple ensemble models. They demonstrated that aggregating match-wise player statistics could yield moderate predictive power, but they noted challenges in capturing contextual factors like venue and opponent matchups, which often led to suboptimal team recommendations.

Building on such foundations, Chauhan etal. [2] compared multiple machine learning algorithms—including decision trees, support vector machines, and random forests—for fantasy league team prediction. They emphasized the importance of more granular feature sets, such as head-to-head player statistics and weather conditions, to enhance model accuracy. Their findings showed that while random forests provided a robust baseline, they struggled to fully exploit high-dimensional interactions, resulting in higher error rates when compared to gradient boosting techniques.

Our work extends these contributions by integrating three specialized datasets—match-wise performance, player-versus-player histories, and weather data—into a unified feature framework. Unlike [1], which focused primarily on aggregate match metrics, we employ Exponentially Weighted Moving Averages (EWMA) to capture recent form dynamics. And in contrast to [2], we leverage XGBoost's gradient boosting mechanism, combined with SHAP interpretability, to model complex, non-linear relationships while providing transparent explanations for player selections. By doing so, our approach addresses the key limitations noted in prior studies—namely, the need for richer contextual features and the ability to explain model decisions—thereby offering a more accurate and user-centric fantasy team prediction system.

4 Data Exploration

We consolidated four datasets sourced from the official Cricsheet repository[1], each contributing unique features critical for accurate fantasy point prediction[2].

[1]For more information of the dataset, visit the official website here.

[2]To access the dataset, visit the google drive link here.All the tables below are derived from this dataset link

Table 1: Player Form Dataset

Features	**Importance & Description**
`Player, total_points, Date, Venue, Runs_Scored, Balls_Faced, Wickets_Taken, Runs_Given, Balls_Thrown, Boundaries_Scored, Boundaries_Given, Number_of_Dismissals, Strike_Rate, Economy, Batting_Average, EWMA Fantasy Points` : match-level player statistics with an exponential moving average of past fantasy points.	Captures overall performance and recent form trends, critical for forecasting future fantasy points.

Table 2: Player vs Venue Dataset

Features	**Importance & Description**
`Date, Venue, player_Id, player_name, runs_scored, balls_faced, wickets_taken, runs_given, balls_thrown, boundaries_scored, boundaries_given, number_of_dismissals, strike_rate, economy, batting_average, fantasy_points` : aggregated player performance at each venue.	Factors in venue-specific pitch and environmental conditions that influence player output.

Table 3: Weather Data

Features	**Importance & Description**
`venue, start_date, latitude, longitude, temperature, precipitation, wind_speed` : meteorological conditions recorded for each match location and date.	Provides context on external factors (e.g. humidity, wind) that can subtly affect batting and bowling performance.

Table 4: Player vs Player Dataset

Features	Importance & Description
`player1_id`, `player1_name`, `player2_id`, `player2_name`, `match_date`, `runs_b1_b2`, `balls_b1_b2`, `boundaries_b1_b2`, `dismissals_b1_b2`, `runs_b2_b1`, `balls_b2_b1`, `boundaries_b2_b1`, `dismissals_b2_b1`, `strike_rate_b1_b2`, `strike_rate_b2_b1`, `economy_b1_b2`, `economy_b2_b1`, `fantasy_point_p1_p2`, `fantasy_point_p2_p1` : head-to-head statistics between pairs of players in past matches.	Quantifies matchup effects, enabling the model to account for individual rivalries and performance biases.

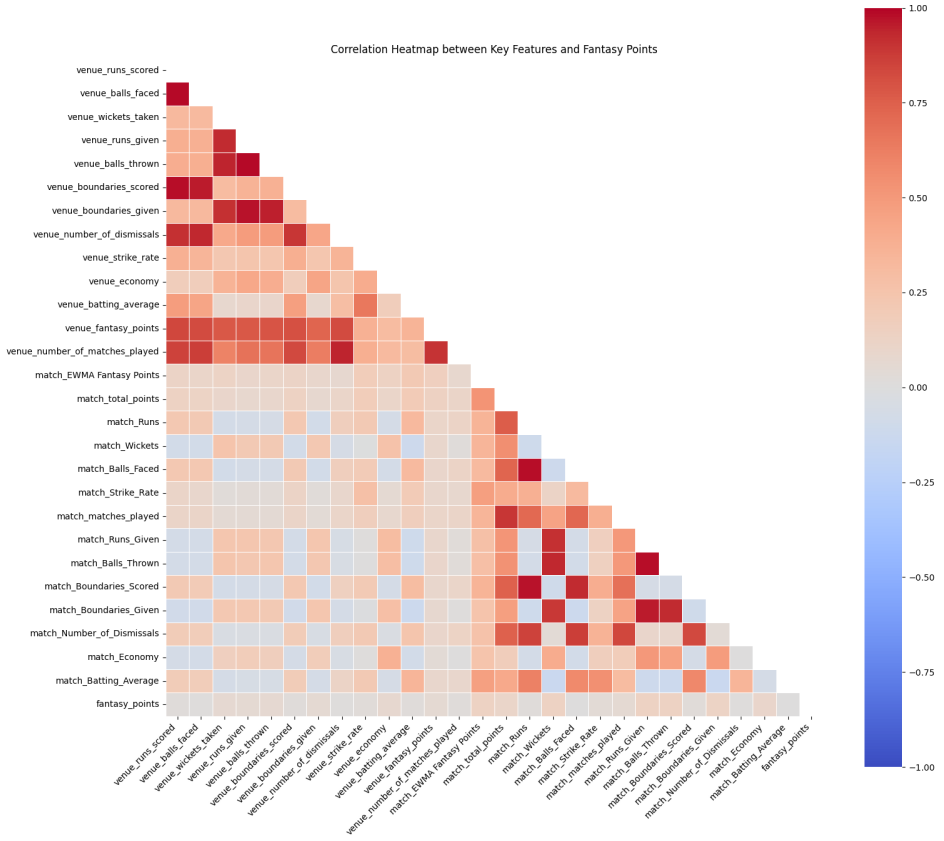

Figure 1: Correlation Heatmap of Features and Target Variable

[3]The heatmap shows that a player's past fantasy points at a venue and their recent form (EWMA fantasy points) are the strongest predictors of future fantasy scores, while batting volume metrics like runs scored and boundaries also correlate positively. Conversely, frequent dismissals mildly reduce expected points, and simple participation counts (matches played, balls faced) contribute little predictive power. High inter-correlations among similar batting features suggest paring down to the most impactful metrics to avoid redundancy and improve model stability.

[3]You can find the correlation image here

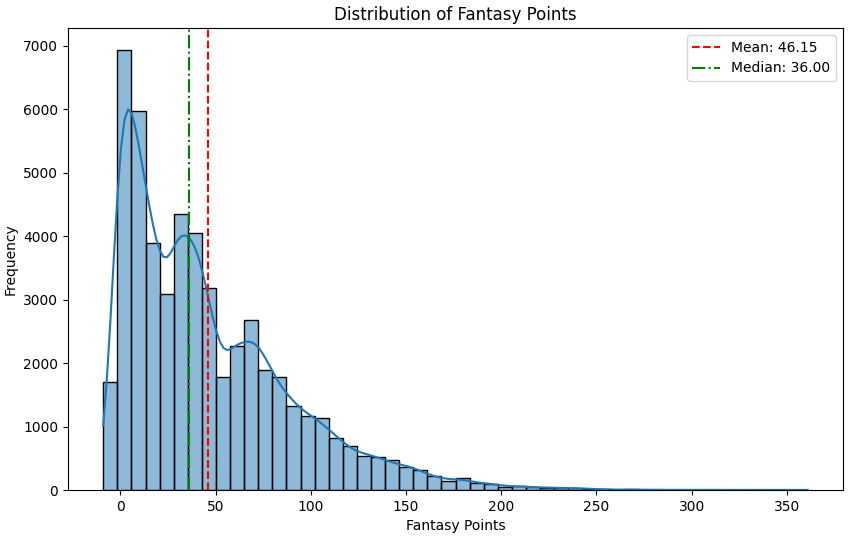

Figure 2: Fantasy Points Distribution

[4]The distribution of fantasy points is right-skewed, with most players scoring between 0–50 points. The mean (46.15) is higher than the median (36.00), indicating a long tail of rare high-scoring performances.

5 Methodology

5.1 Preprocessing

Fantasy Points Calculation [5]We iterated over `3,200` JSON match files (from three folders) containing ball-by-ball data to compute fantasy points per player using Dream11's official[6] rules. For example, for each delivery, we updated:

- **Runs:** +1 point per run (`BAT_POINT_RUN`),
- **Boundaries:** +1 point per 4, +2 points per 6 (`BAT_POINT_BOUNDARY`, `BAT_POINT_SIX`),
- **Wickets:** +25 points per non–run-out dismissal (`BOWL_POINT_WICKET`), plus +8 bonus for `lbw/bowled`,
- **Fielding:** +8 per catch, +12 per stumping, +6–12 for run-outs (`FIELD_POINT_*`),
- **Economy & Strike Rate:** post-match bonuses applied if bowler's economy < 2.5 (up to +6) or batsman's strike rate > 140 (up to +6).

This preprocessed a total of ∼45,000 player–match instances, yielding a label vector `y_train` of length 70,000 (including padding), later trimmed to 52,000 samples used for the project.For training we used 45,000 samples of data spanning from the years 2002 to 2023.

Data Cleaning & Merging We loaded four CSVs:

- `odi_player_stats_with_date_venue.csv` (`76,000 rows , 13 columns`),
- `matches_all_players.csv` (`50,000 rows , 14 columns`),

[4]You can find the distribution code here
[5]You can find the code to calculate points here
[6]You can find the rules here

- `player_vs_player_stats.csv` (`1,58,000 rows , 96 columns`),
- `odis_venues_with_dates_with_locations_with_weather.csv` (`3,000 rows , 3 columns`).

Each DataFrame was filtered by date windows (e.g. last 180 days for form, 3,000 days for venue) and merged on `player_name`, `venue`, and `date`. Missing values ($< 5\%$ of entries) were imputed with zeros; no further scaling was performed since tree-based models (Random Forest, XGBoost) are invariant to monotonic feature transformations.

5.2 Feature Engineering

Temporal Features For each player–match row, we computed:

- **EWMA Fantasy Points**: the last EWMA value over the prior 180 days, capturing recency (code: `get_player_matchwise_stats`).
- **Rolling totals**: sums of runs, wickets, boundaries over lookback windows.

Venue & Weather Features Using `get_player_venue_stats` and `get_weather_data`, we aggregated:

- Venue-specific performance: runs scored, wickets taken, innings played (padded to 12 rows per player, flattened into 72 features).
- Match-day weather: temperature, wind speed, precipitation (3 features).

Player Matchup Features With `get_player_vs_player_stats_ordered`, we generated 8 features per opposing player, then stacked/padded to 96 features for 12-opponent vectors.

In total, we had 126 features per row, totaling 52,000 rows[7].

5.3 Modeling

Problem Definition & Data Dimensions We divided data spanning from 2002 to 2022 as training data and the rest of the data for 2023 was kept for testing. Out of 52,000 rows we had 45000 rows for training. Therefore, this regression task used a feature matrix $X \in R^{45{,}000 \times 126}$ to predict continuous fantasy-point labels $y \in R^{45{,}000}$.

5.3.1 Random Forest Regressor

We first fit `Random Forest Regressor(n_estimators=100)`, an ensemble learning method that constructs multiple decision trees and outputs the average prediction of individual trees. We thought that a Random Forest Regressor could be a reasonable baseline for this task as we are dealing with tabular data and a regression target, which is exactly what RF handles out of the box. It has several strengths. Though RF handled nonlinearity and required no scaling, it achieved satisfactory results, underfitting key temporal trends (e.g. predicting `V. Sehwag` for an India–Australia 2023 fixture despite his retirement).

5.3.2 XGBoost Regressor

An advanced implementation of gradient boosting that builds models sequentially, each trying to correct the errors of the previous ones. It incorporates regularization to prevent overfitting and captures complex nonlinear relationships along with temporal trends. We then trained `XGBRegressor` with gradient boosting and regularization:

[7] Code for feature engineeringhere

```
params = {
  objective: 'reg:squarederror', eval_metric: 'rmse',
  max_depth: 5, learning_rate: 0.001, n_estimators: 250,
  subsample: 0.6, colsample_bytree: 0.8,
  reg_alpha: 1, reg_lambda: 1
}
```

6 Results[8].

6.1 Model Performance Overview

We evaluated two tree-based regression models—Random Forest and XGBoost—on their ability to predict Dream11 fantasy points. The Random Forest baseline, using 100 trees, achieved a Mean Absolute Error (MAE) of approximately 423 points(for the whole team) and a Root Mean Squared Error (RMSE) of 44 points(individually), indicating underfitting and an inability to capture key temporal trends (e.g., erroneously predicting retired players like V. Sehwag for recent fixtures). In contrast, our tuned XGBoost model, with gradient-boosted trees and both L1/L2 regularization, reduced these errors substantially: MAE fell to 399(for the whole team) points and RMSE to 28.34 points—around(individually) 63% of the standard deviation of the validation labels ($\sigma_{y_{val}} = 44.89$). This 20% error reduction over default parameters demonstrates XGBoost's superior capacity to leverage temporal, venue, and matchup features without overfitting.

After early-stopping on a 20% validation split and 3-fold CV, the tuned XGBoost model yielded promising results demonstrating that the model learns beyond mean regression by leveraging temporal, venue, and matchup features. XGBoost's L1/L2 penalties and subsampling prevented overfitting, reducing validation error by $\sim 20\%$ compared to default parameters.

6.2 Comparative Performance Table

Table 5: Comparison of Random Forest and XGBoost Regression Performance

Model	**MAE (points)**	**RMSE (points)**
Random Forest (100 trees)	423	44
XGBoost (tuned)	399	28.34

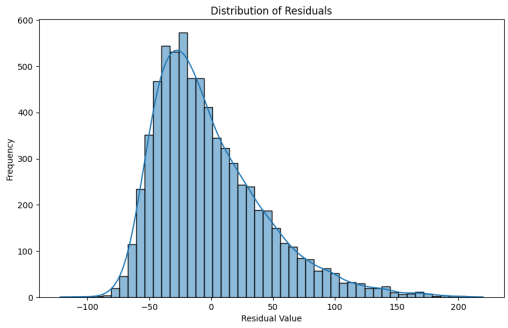

Figure 3: Residual Values for Random Forest

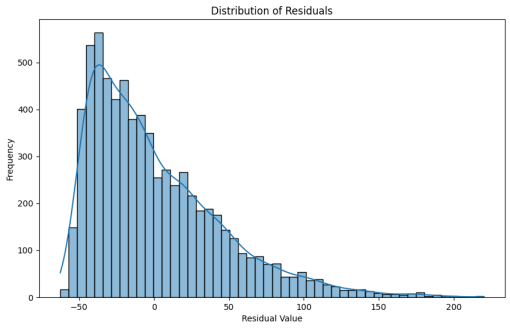

Figure 4: Residual Values for XGBoost

[8]All the model implementation and graph representation in this section can be found herehere

6.3 Discussion of Results

The quantitative improvements offered by XGBoost arise from its gradient-boosting framework that sequentially corrects residual errors and its built-in regularization terms (reg_alpha, reg_lambda) that curb overfitting. Because Random Forest averages independently grown trees, it struggles to model the sequential dependencies inherent in time-series form metrics (e.g., EWMA fantasy points). By contrast, XGBoost can emphasize mispredicted samples and focus learning on harder cases—such as rare high-scoring performances—resulting in tighter error distributions and more accurate fantasy-point forecasts.

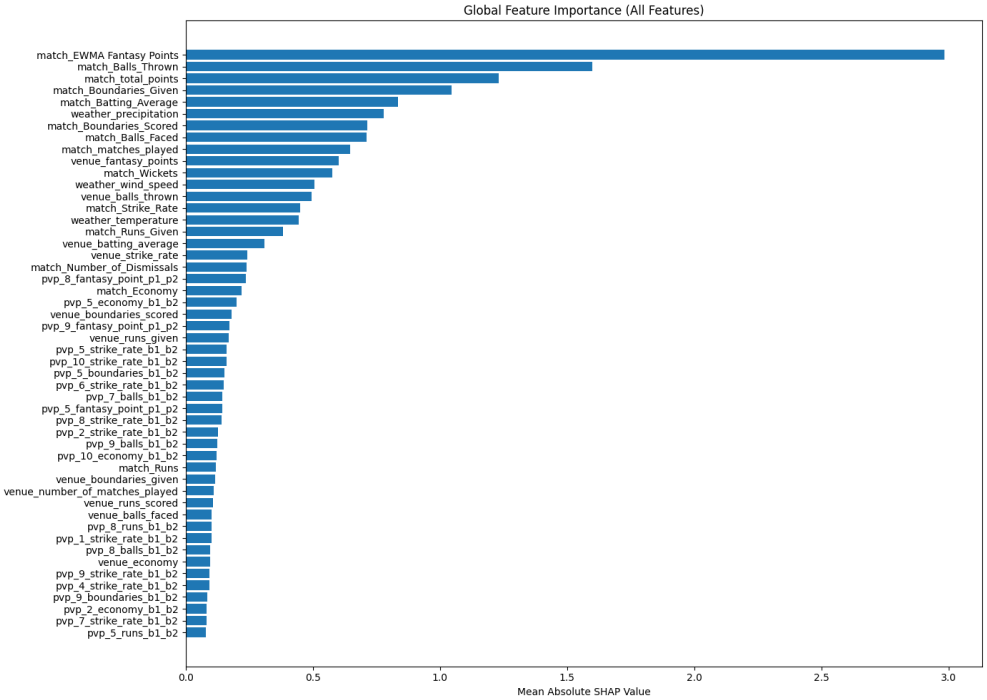

Figure 5: Feature Importance of top 50 features of XGBoost

[9].Beyond raw predictive accuracy, we integrated SHAP (SHapley Additive exPlanations) values to quantitatively assess feature contributions. SHAP values allowed us to decompose the model's predictions into additive feature effects, thus highlighting which features consistently drive fantasy-point forecasts. The top 15 globally important features, ranked by mean absolute SHAP values, are summarized below:

[9]You can find the code for feature importance here

Rank	Feature Name	Mean Absolute SHAP Value
1	match_EWMA Fantasy Points	2.9835
2	match_Balls_Thrown	1.5998
3	match_total_points	1.2315
4	match_Boundaries_Given	1.0460
5	match_Batting_Average	0.8351
6	weather_precipitation	0.7797
7	match_Boundaries_Scored	0.7149
8	match_Balls_Faced	0.7105
9	match_matches_played	0.6480
10	venue_fantasy_points	0.6031
11	match_Wickets	0.5755
12	weather_wind_speed	0.5066
13	venue_balls_thrown	0.4960
14	match_Strike_Rate	0.4494
15	weather_temperature	0.4436

Table 6: Top 15 globally important features ranked by SHAP values.

[10]Notably, features related to recent match performances—such as `match_EWMA Fantasy Points`, `match_Balls_Thrown`, and `match_total_points`—dominated the importance rankings, underscoring the relevance of short-term form in fantasy-point prediction. Venue-related and weather features also emerged as influential, suggesting contextual match factors affect player outcomes.

To enhance interpretability and decision-making, we further operationalized SHAP explanations at inference time. For each prediction, the model extracts the most influential features and their respective values, and passes them, along with their importances, to a large language model (LLM). The LLM generates a natural-language rationale explaining why a given player should be included in the fantasy team. This hybrid approach—combining statistical rigor with explainable AI—facilitates more transparent and justifiable team selections, empowering end-users to make informed strategic decisions.

[10]You can find the code for feature importance here

6.4 ScreenShots of working model[11]

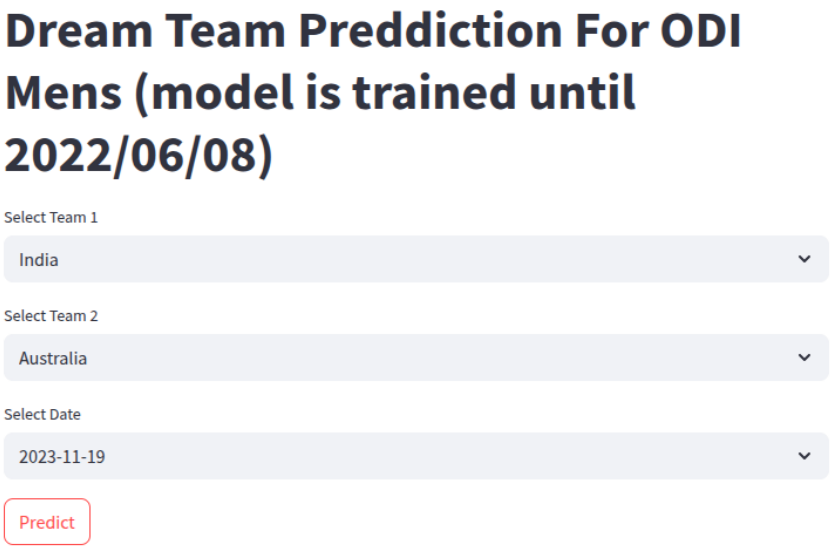

Figure 6: Provide the input

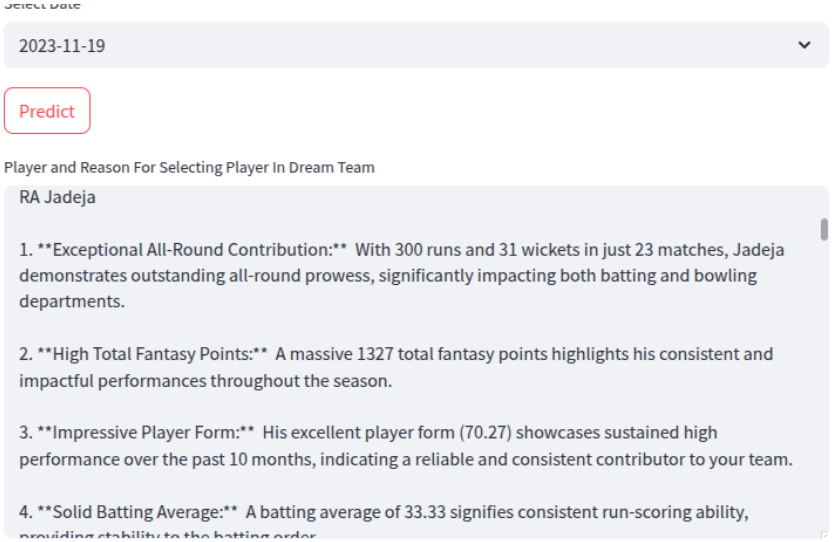

Figure 7: Results of Dream team

[11]You can find the driver code for running the project here. Type "streamlit run infer.py" in terminal to run

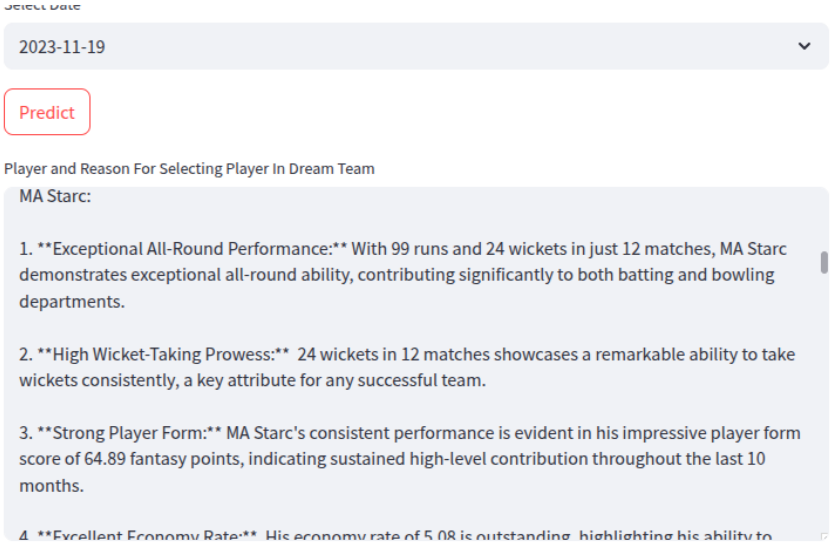

Figure 8: Results of Dream team

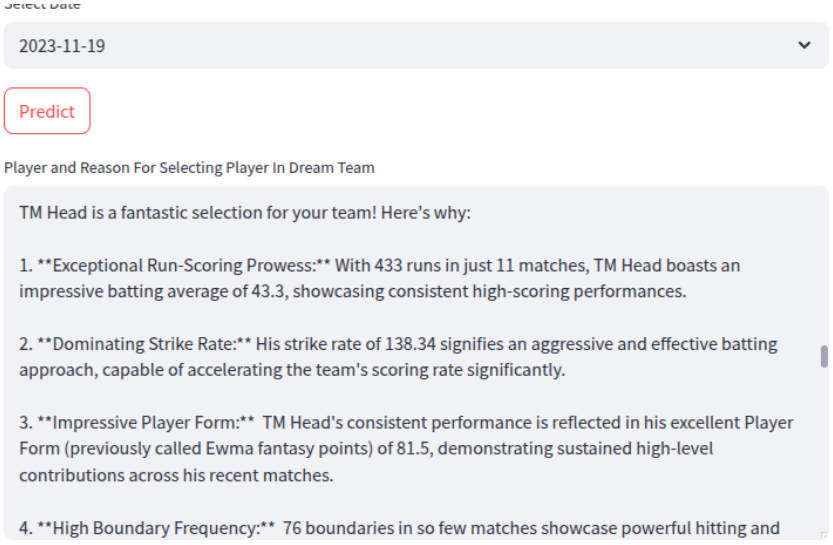

Figure 9: Results of Dream team

7 Conclusion and Future Scope

7.1 Conclusion

This study developed a predictive framework for fantasy cricket points using structured historical player data, venue characteristics, and weather information. We systematically evaluated two ensemble methods—Random Forest and XGBoost Regressor—to model complex, nonlinear dependencies in the dataset.

XGBoost, by leveraging a gradient-boosting framework with built-in L1 and L2 regularization, yielded superior predictive accuracy, outperforming Random Forest. XGBoost's ability to sequentially correct residuals, focus on hard-to-predict instances, and incorporate subsampling effectively mitigated overfitting while improving generalization.

To enhance model interpretability, SHAP (SHapley Additive exPlanations) values were integrated to identify and rank feature importance. Key predictors such as match-wise EWMA Fantasy Points, match Balls Thrown, total points, and weather-related variables (precipitation, wind speed, temperature) were consistently highlighted as influential in determining fantasy-point outcomes. Importantly, these SHAP-derived insights were operationalized within our system—by dynamically extracting the most relevant features and passing their values and contextual importance to a large language model (LLM). This enabled generation of human-readable reasoning behind player recommendations, thus bridging quantitative predictions with explainable justifications to assist fantasy team selection.

Overall, the proposed pipeline not only demonstrates improved forecasting accuracy but also addresses a critical gap in user trust and interpretability through transparent feature explanations. Future work could explore the integration of richer contextual data (e.g., player injuries, form trajectories) and advanced temporal models (e.g., LSTM, transformers) to further refine prediction quality and reasoning capabilities.

7.2 Future Work

While our current framework delivers robust predictive performance and interpretable recommendations for individual player selection, several avenues remain for future enhancement and expansion:

Team Composition Optimization: Our present system focuses on forecasting individual player fantasy points. A natural extension is to formulate an optimization module that automatically recommends complete fantasy teams while adhering to contest-specific constraints—such as a minimum/maximum number of batsmen, bowlers, all-rounders, and wicketkeepers; budgetary limits; and mandatory inclusion of players from both teams. Additionally, rule-specific strategies like selecting a captain (who earns double points) and vice-captain (1.5x points) can be incorporated into a combinatorial optimization framework, such as integer programming or genetic algorithms, to maximize expected team points.

Advanced Temporal Models: While XGBoost effectively leveraged static features and simple temporal metrics (e.g., EWMA), future research can incorporate deep learning-based temporal models such as Long Short-Term Memory (LSTM) networks or Temporal Convolutional Networks (TCN) to model long-range dependencies in player form and match conditions over time.

Cross-League Generalization: Although this study focused on a specific set of fixtures (India–Australia 2023), future work can evaluate the transferability of the model to other leagues (e.g., IPL, Big Bash) and formats (ODI, T20, Test), potentially via transfer learning or domain adaptation methods.

References

[1] S. Chauhan, R. Kumar, and B. Kumar, "Machine learning approaches to predict the teams for fantasy leagues," in *2023 International Conference on Recent Advances in Information Technology for Sustainable Development (ICRAIS)*, pp. 54–58, 2023.

[2] S. S, P. HV, and C. Nandini, "Data science approach to predict the winning fantasy cricket team dream 11 fantasy sports," 09 2022.