INTER IIT
TECH MEET 13.0

Team 27

DREAM11

NEXT-GEN TEAM BUILDER WITH PREDICTIVE AI

1. INTRODUCTION

In the dynamic realm of fantasy sports, predicting optimal team compositions and individual player performances has become increasingly significant, particularly in platforms like Dream11 for cricket enthusiasts. The integration of machine learning (ML) and statistical modelling offers promising avenues to enhance these predictions.

Current prevalent approaches often utilize traditional ML models, such as Random Forests and Decision Trees, to forecast match outcomes and player performances. While effective with structured data and historical records, these models often struggle to generalize across cricket formats (T20, ODI, Test), adapt to real-time variables, and account for nuanced contributions from different player roles like batsmen, bowlers, and all-rounders, limiting their predictive accuracy and utility in fantasy sports.

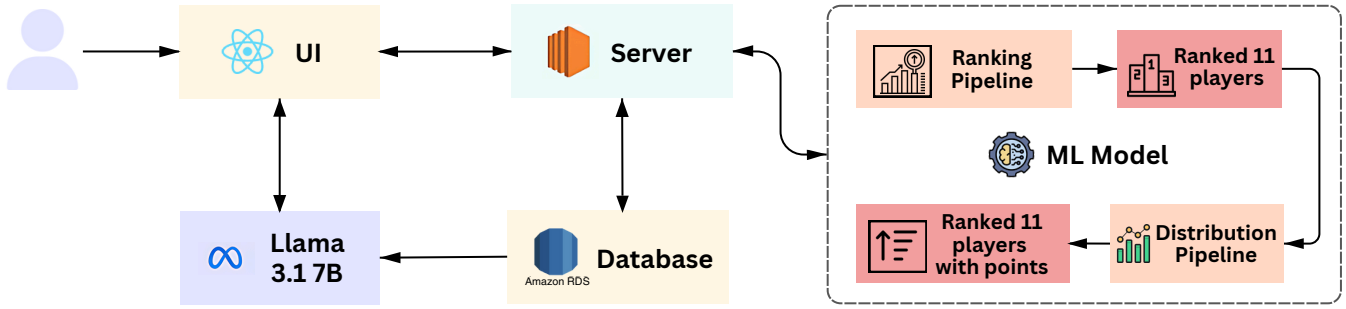


Figure 1: User Interaction Processing Framework

We propose a novel methodology for predicting fantasy points using a robust two-stage pipeline: a ranking pipeline and a distribution pipeline. The ranking pipeline employs role-specific models for batsmen, bowlers, and all-rounders, utilizing features such as runs, wickets, strike rate, venue-specific statistics and player-to-player relationships to accurately rank players. The distribution pipeline uses historical rank-based contribution ratios to allocate predicted fantasy points among the top 11 players, ensuring accurate predictions and maximizing the dream team's performance potential.

This robust pipeline significantly improves predictive accuracy by addressing challenges such as data sparsity, inter-player rank differences, and imbalanced performance metrics. By leveraging role-specific modelling and rank-based contribution allocation, it ensures reliable fantasy point predictions that align with the constraints and rules of fantasy platforms, offering practical and impactful solutions.

Dream11 Fantasy Points System

The Dream11 fantasy points system evaluates players' performances in batting, bowling, and fielding across various cricket formats. Batting points are awarded for runs, boundaries, milestones such as half-centuries and centuries, and strike rates, while dismissals for a duck incur penalties. Bowling contributions are scored through wickets, economy rates, and milestone bonuses for achievements like taking three or more wickets in a match, with maiden overs also rewarded. Fielding points cover catches, stumpings, and run-outs, with additional bonuses for multiple catches.

Role-based multipliers enhance scoring, with captains earning double points and vice-captains earning 1.5 times the points. Points are also awarded for inclusion in the announced playing lineup. This system provides a detailed and strategic framework for analyzing and ranking player performances.

In the following sections of this report, we delve into the details of our study. Section 2 provides a concise literature review to position our work within the broader context of existing research. Section 3 outlines our proposed solution, highlighting the data processing pipeline, feature engineering strategies, and model architecture. In Section 4, we present experimental results and analyze their implications. Section 5 discusses the challenges faced, limitations of our approach, and potential directions for future research. We will discuss the details of our product pipeline and the integration of Generative AI in the Appendix.

2. LITERATURE REVIEW

Advancements in machine learning and statistical modelling have significantly improved the prediction of optimal teams in fantasy cricket leagues like Dream11. Key areas of focus include feature engineering, modelling approaches, and rule-based strategies to enhance prediction accuracy and effectiveness.

Feature Engineering

Feature engineering is crucial for transforming raw data into meaningful inputs that capture patterns in player performance. Choudhari et al. **[3]** enriched their models by incorporating skill-based rankings and external factors like pitch conditions and weather, thereby improving accuracy. Mohith et al. **[1]** developed new batting and bowling metrics using weighted directed graphs, known as Player-to-Player (P2P) metrics, to capture interactions between players. This approach provided nuanced insights into individual matchups. Ramalingam et al. **[4]** emphasized the importance of the feature engineering process. They considered features like the number of runs scored, wickets taken, player experience, and match performances. Balpande et al. **[5]** included venue specific features, such as pitch and weather conditions, generating real-time performance indicators to enhance their predictive models.

Modelling Approaches

Various machine learning techniques have been employed to predict player performance and optimize team selection. Choudhari et al. **[3]** evaluated multiple classifiers and found that Random Forest outperformed others in predicting match winners, achieving 74% accuracy. Ramalingam et al. **[4]** also used batting and bowling statistics for player prediction. Balpande et al. **[5]** leveraged ensemble methods like XGBoost and CatBoost, incorporating past performances and match conditions to recommend fantasy teams aiming to maximize points. Ravichandran et al. **[6]** compared models and found Linear Regression most effective in forecasting player performance. Patil et al. **[7]** tried to select the best playing eleven based on past records, reducing bias in team selection. Building upon these methodologies, Mohith et al. **[1]** combined advanced P2P feature engineering with optimization techniques, effectively predicting a significant portion of Dream Team members and enhancing users' chances in fantasy leagues.

Rule-Based Approaches

Rule-based strategies involve applying predefined constraints essential for compliance with fantasy platforms like Dream11. Mohith et al. **[1]** employed optimization techniques to align team selection with constraints such as budget limits and team composition rules, using tools like Excel's Solver to maximize the total strength score while adhering to regulations. Choudhari et al. **[3]** ensured their recommended teams were both optimized for performance and compliant with game guidelines by integrating rule-based constraints. Balpande et al. **[5]** also incorporated rule-based logic to meet constraints like credit caps and player quotas, balancing predictive accuracy with platform requirements. Furthermore Ravichandran et al. **[6]** emphasized the importance of forming balanced teams with the appropriate mix of player roles, aligning with the regulations of fantasy platforms.

Challenges and Limitations

Understanding the inherent challenges in sports prediction is crucial for improving the reliability and effectiveness of predictive models in fantasy cricket. Patil et al. **[7]** acknowledged the limitations of their models in accounting for external factors like player fitness and psychological well-being, which are difficult to quantify but can significantly affect performance. Similarly, Choudhari et al. **[3]** highlighted that unforeseen events such as sudden weather changes, pitch conditions, and in-game injuries can drastically alter the course of a game, posing significant challenges for predictive accuracy. Choudhari et al. **[3]** discussed the difficulties posed in predictions of new or less-experienced players. Furthermore, Mohith et al. **[1]** addressed the issue of exceptional individual performances or match-winning plays, noting that while models can be effective in general scenarios, they struggle to account for the dynamic and unpredictable nature of sports, especially cricket.

3. METHODOLOGY AND IMPLEMENTATION

Data Extraction

The data was primarily extracted from Cricsheet, which contained a comprehensive collection of ball-by-ball details for different cricket formats. Specifically, data was gathered for 840 Test matches, 3,451 One-Day International (ODI) matches, and 13,015 T20 matches, encompassing both international and domestic competitions. These included 3,844 T20 Internationals (T20Is) and 9,171 domestic T20 league matches, such as the Indian Premier League (IPL), Big Bash League (BBL) and Caribbean Premier League (CPL).

The ball-by-ball data was aggregated to compute individual player statistics for each match. Fantasy points were calculated based on Dream11's predefined scoring system covering batting, bowling, and fielding contributions, enabling the evaluation of player performances on a per-match basis. It also served as a basis for deriving venue-based stats, enabling location-specific performance analysis.

In addition to the Cricsheet data, ESPN cricinfo was scraped to get the player-specific details, such as batting and bowling styles and mapped using the corresponding player IDs from Cricsheet. This facilitated the integration of player attributes and match context into the dataset, ensuring a more holistic representation of each player's role within a team.

Feature Engineering

Features were developed across four distinct domains to capture different aspects of player performance. These domains allowed for a comprehensive representation of the various factors that could influence player performance and, by extension, the prediction of fantasy points. The four domains are as follows:

Performance Trend Features

The performance trend domain focused on capturing recent player performance trends. Key metrics, such as cumulative runs, Exponential Time Weighted Performance Scores (ETWPS), rolling sum of strike rate, number of sixes, cumulative fantasy points, etc., were computed over varying numbers of matches played by each player. These statistics provided a dynamic view of a player's form and consistency, reflecting recent performance and helped in identifying players who were currently in strong form.

Venue-Based Features

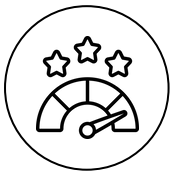
In this domain, player performance at specific venues was analyzed. This domain took into account factors such as runs scored, wickets taken, and sixes hit by players at particular venues in previous matches. By incorporating venue-specific data, this domain aimed to indirectly account for the impact of different playing conditions, such as pitch behavior and local weather, which can highlight the venue-specific trends and biases in player behavior.

Player-to-Player Features

The player-to-player domain created comparative metrics to assess a player's performance relative to others. This approach helped to better understand how they performed in relation to peers in similar roles. Features like player matchups, and head-to-head performance indicators allowed for the evaluation of how players might perform in specific match scenarios, enhancing the predictive accuracy of fantasy point calculations.

Performance Forecasting Features

Performance forecasting features focused on quantifying the expected average performance of bowlers and batsmen based on historical data. Features such as base bowlers and base batsmen were created, which represented the average expected performance for players in these roles under typical conditions. These features helped capture the inherent uncertainty and variability in cricket performance, as players can perform above or below their expected average due to a variety of factors (e.g., match conditions and opposition strength). Incorporating probabilistic features allowed for a more nuanced and dynamic representation of potential outcomes, accounting for performance variability.



Together, these four domains of feature engineering provided a well-rounded and comprehensive framework for evaluating player performance, incorporating both individual and contextual factors that influence outcomes in fantasy cricket.

It is important to note, features from the above four domains were exclusively used in our ranking pipeline.

Team Aggregated Features

The team aggregated features domain involves taking the average of the features from the four domains—Performance Trend Features, Venue-Based Features, Player-to-Player Features, and Performance Forecasting Features—across all 22 players participating in a particular match. This aggregation method ensures that the features reflect a comprehensive view of the match setting as a whole.

The above mentioned feature domain was exclusively used in our distribution pipeline and not in the ranking pipeline.

Performance Trend Features	Venue Based features	Player to Team & Player to Player Features	Performance Forecasting Features
Player Average Runs (Last 10 matches)	Wickets at a specific venue	Player Performance against a specific opponent	Batsman Fatigue Score
Cumulative Avg of Bowling/Batting Fantasy Points	Sixes conceded by the bowler at that venue	Historic Performance of a player against specific bowler	Batsman Base Performance
Bowler Average Wickets (Last 10 Matches)	Runs scored by the player at that venue	Player's performance probability against opposing bowlers	Cumulative Dream Team appearance of a player yearly

Table 1: Example of Player Performance Features across domains

Model Pipeline Overview

1. Ranking Pipeline: The ranking pipeline was designed to rank players based on predicted fantasy points. These points are used solely for ranking purposes and are **not** the final fantasy points. The pipeline consisted of several key stages, including data preprocessing, model training and role-based predictions, ultimately leading to the generation of ranked player lists.

2. Distribution Pipeline: The distribution pipeline was designed to allocate the predicted fantasy points among the top 11 players identified by the ranking pipeline. This component leverages statistical and machine learning methodologies to ensure a balanced and data-driven distribution of fantasy points, accurately reflecting each player's contribution within the context of the predicted team performance.

Ranking Pipeline

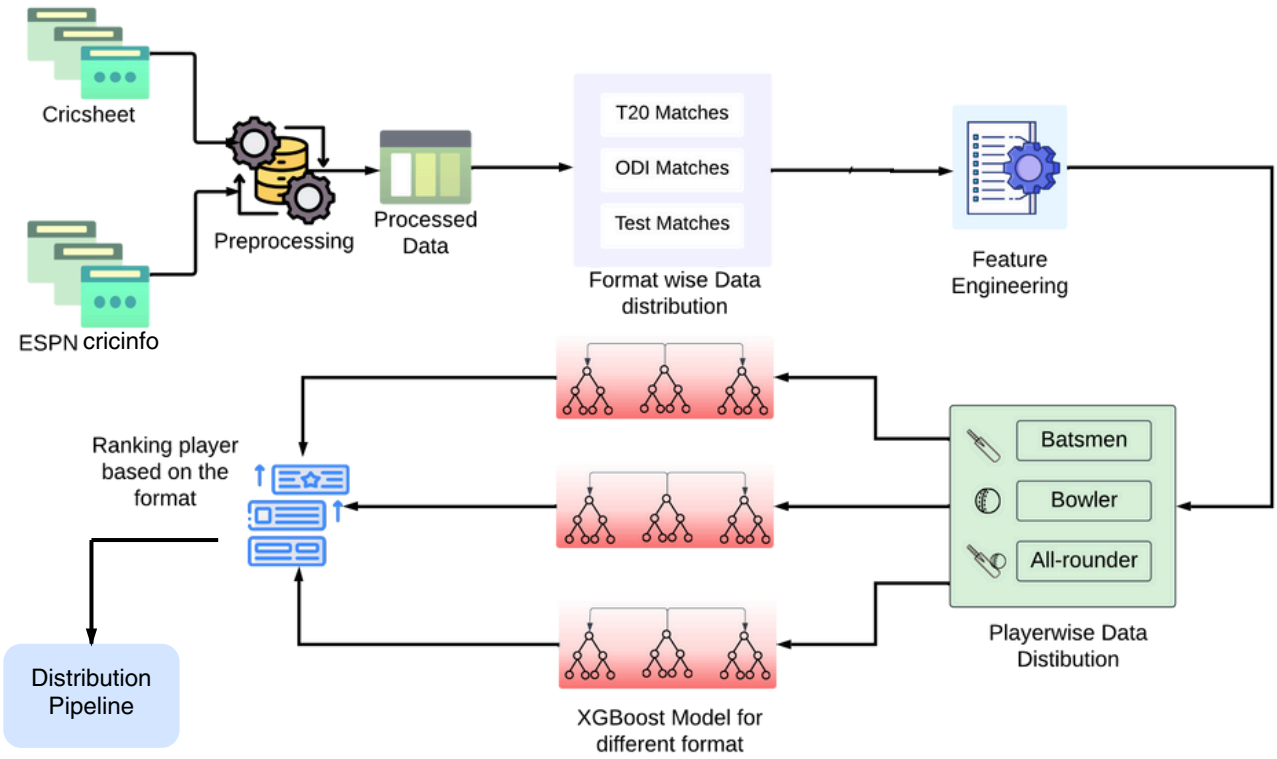


Figure 2: Flow of Ranking Pipeline

Data Segmentation and Preprocessing

The ranking pipeline began by segmenting the dataset based on different match formats (Tests, ODIs, and T20s). This segmentation was essential to account for the differences in player performance across match formats, such as the longer duration of Test matches versus the shorter nature of ODIs and T20s.

Subsequently, feature engineering was performed, specifically tailored to each format's unique style and length. This approach was essential, as each format requires distinct strategies from the player's perspective. The tailored feature engineering demonstrated a significant improvement in performance compared to employing a common set of features across all formats. This method not only optimized the analysis but also provided deeper insights into the nuances of each format, enabling more accurate predictions. We have also calculated the actual fantasy points for the match using the match statistics.

Role-Based Model Training

To account for the unique contributions of players in their specific roles—batsmen, bowlers, and all-rounders—the pipeline categorized players into these three groups based on custom-defined rules outlined below. Separate XGBoost models were then developed for each role, utilizing role-specific features to ensure accurate prediction of fantasy points.

Players are classified as batsmen if their average overs bowled is less than 1, indicating minimal involvement in bowling activities and a primary focus on batting.

Players with average overs bowled greater than 1 and a batting average of 10 or less signify their specialization in bowling.

Players are classified as all-rounders if their batting average exceeds 10 and their average overs bowled is greater than 1. This classification reflects their significant contributions to both batting and bowling.

The above mentioned player classification criteria is for ODIs and T20s. For Tests, refer to Appendix.

Total Fantasy Points = **Batting Points** + **Bowling Points** + **Fielding Points**

◆ Batsman Model

The batsman model focused exclusively on batting-specific features to predict **batting fantasy points** for batsmen. Features such as runs scored, strike rate, and the number of boundaries hit were used, as these directly represent the primary contributions of batsmen to a match's outcome.

◆ Bowler Model

The bowler model focused solely on bowling-specific features to predict **bowling fantasy points** for bowlers. Key features such as wickets taken, economy rate, and the number of maiden overs were included, capturing the essential aspects of a bowler's performance.

◆ All-Rounder Model

A hybrid approach was employed to capture the combined contributions of all-rounders as both batsmen and bowlers by predicting **sum** of their batting and bowling fantasy points. Thereby, reflecting the all-rounder's overall impact on the match.

This pipeline focuses exclusively on modelling batting and bowling fantasy points, intentionally excluding fielding fantasy points. This exclusion is due to the limited availability of detailed fielding data and the high unpredictability of fielding performance, making accurate predictions in this area challenging. While the pipeline uses predicted batting and bowling points to rank players, these predictions are **not** directly used in the final fantasy points calculation. The **final points** incorporate additional factors, such as fielding data and real-time match conditions, which are accounted for later in the distribution pipeline.

Prediction Aggregation

The predictions for each player category—batsmen, bowlers, and all-rounders—as outputs from their respective models were concatenated to form the total fantasy points of all the players. Now, these total fantasy points are used to rank the top 11 players.

◆ Explanation of results of the ranking pipeline:

Our model demonstrates strong predictive performance, accurately identifying an average of seven players matching the actual dream team. The accuracy for correctly predicting the first player stands at 13%, surpassing the rule-based method (9%) and the expected value (5%). Additionally, the model achieves 15.6% accuracy in correctly predicting the top 2 out of 3 players and 23.9% accuracy for the top 3 out of 5 players.

The rule-based method ranks players based on the average fantasy points from their past five matches, which has been explained in detail in the Appendix.

Distribution Pipeline

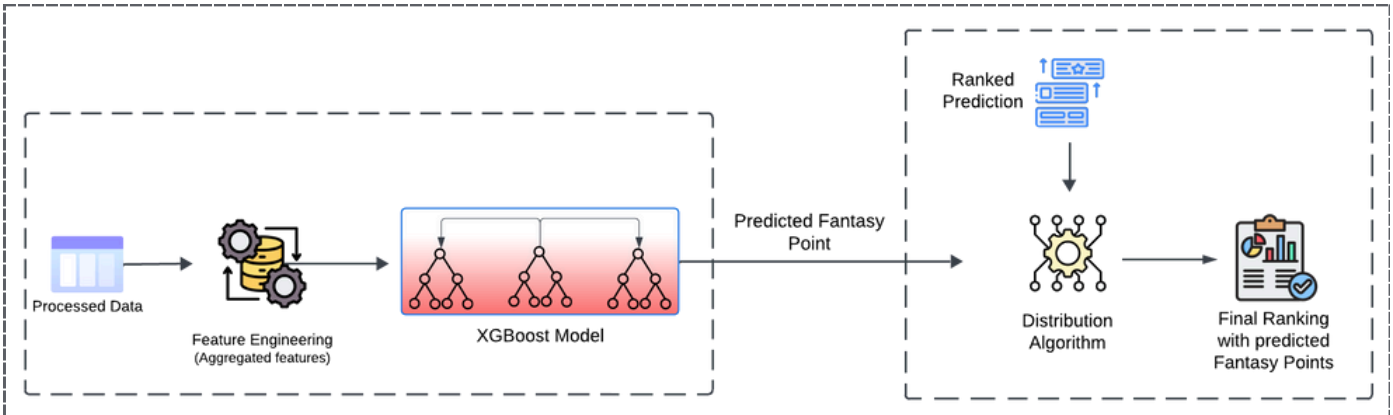


Figure 3: Flow of Distribution Pipeline

Prediction of Total Match Fantasy Points

The distribution pipeline begins by employing team aggregated features (refer Section 3) as input features for an XGBoost regression model. This machine learning model predicts the total fantasy points for a given match. By incorporating cumulative stats such as total runs, wickets, and boundaries of all participating players, the model generates quantitative predictions for the match's total expected fantasy points.

The Formation of the Dream Team

The formation of the dream team combines insights from historical player contributions and predictive analytics to allocate fantasy points effectively. The distribution algorithm begins by taking the top 11 ranked players as determined by the ranking pipeline (Section 3). It then allocates **each player's point share** from the **team's predicted total fantasy points**. This share is decided based on the statistics (refer Appendix) which are calculated by first grouping the training data by match and then sorting the match players by their actual total fantasy points. It then computes the statistical means for each player rank specific to their format to establish **Average Contribution Ratios** (ACR) (described below), accounting for exceptional performances and fielding points.

$$\text{ACR}_i = \frac{\sum_{m=1}^{M} \text{Fantasy points scored by the } i\text{-th position player in match } m}{\sum_{m=1}^{M} \left(\sum_{i=1}^{22} \text{Fantasy points scored by the } i\text{-th position player in match } m \right)}$$

Using these contribution ratios, the algorithm allocates the predicted total fantasy points (as described in the ranking pipeline) among the top 11 players identified by the ranking pipeline. This results in the final output: **The Dream Team**. By integrating machine learning predictions with statistically derived distributions, this approach ensures the dream team reflects both individual player potential and the overall dynamics of the match.

Product UI:

The dashboard is the central interface for the team building stage, offering users an interactive experience for fantasy team selection. The user starts off by viewing the recommended team, and then has access to a variety of features designed to enhance the decision-making stage of the user journey. They can start off by viewing the Recommended Team Report, which analyzes the performance of the suggested lineup. Additionally, the Player Details section provides in-depth player statistics, while the Fantasy Points System page outlines the criteria used to calculate fantasy points. An integrated a Chatbot offers real-time support for user queries. Users can use the edit team option to compare and swap players after viewing detailed statistical metrics. The Custom Team Report page allows users to review their selected team's performance, compare it to the recommended team, and access insights on key players, weather conditions, and pitch analysis. These features help users make more informed decisions with minimal friction when forming their fantasy teams.

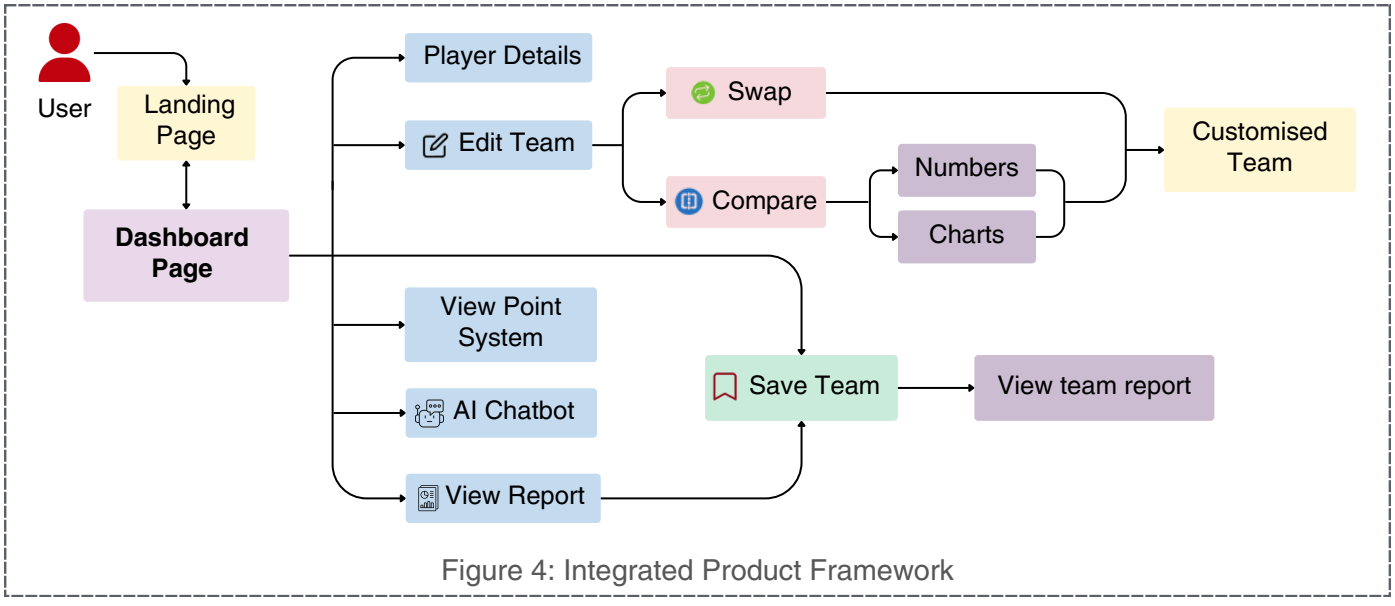


Figure 4: Integrated Product Framework

4. RESULTS AND ANALYSIS

Model Types Overview

The evaluation covered a diverse range of models applied across two pipelines: Ranking and Distribution. These methods included the rule-based approach, linear regression, Boosting (**XGBoost**), Bagging (**Random Forest**), Stacked Regression (Detailed in Appendix), and Deep Learning-based **TabNet**. The rule-based approach, specific to the ranking pipeline, offered a simple deterministic baseline. Linear Regression provided an interpretable benchmark, while TabNet introduced a modern, deep-learning perspective for tabular data. Ensemble methods, including Stacked Regression, demonstrated their well-known strength in generalizing across datasets and delivering robust, high-performance predictions.

Model Type	T20		ODI		Test		Combined	
	3 of 5*	ACP**	3 of 5	ACP	3 of 5	ACP	3 of 5	ACP
Rule based approach	17.4	6.39	12.4	6.12	13.5	6.01	14.5	6.12
XGBoost	**26.9**	**7.07**	**21.0**	**6.79**	**21.4**	**6.65**	**23.9**	**6.95**
TabNet	25.6	7.03	14.2	6.65	21.2	6.59	22.4	6.88
Linear Regression	22.0	7.00	15.6	6.68	20.1	6.59	19.7	6.52
Stacked Regression	25.9	7.03	19.8	6.69	21.3	6.63	23.1	6.89
Random Forest	22.5	6.87	17.4	6.32	18.0	6.30	18.9	6.51

Table 2: Evaluation of different models in the Ranking Pipeline

***3 of 5** : Percentage of 3 or more common players in top 5 of our prediction with the actual top 5.*
***Average Common Players (ACP)**: The average number of players that appear in both the predicted top 11 and the actual top 11.*

In the ranking pipeline, the rule-based approach provided a baseline for comparison but underperformed significantly. Its combined "3 of 5" score was 14.54, and the ACP score was 6.12, reflecting its limitations in handling the complexities of ranking tasks. TabNet and linear regression exhibited moderate performance, outperforming the rule-based system but remaining below the ensemble methods. Linear regression recorded a "3 of 5" score of 19.78 and a ACP score of 6.524, while TabNet achieved a "3 of 5" score of 22.48 and a ACP score of 6.875 in the combined dataset.

Among the ensemble models, XGBoost demonstrated the best performance, achieving a "3 of 5" score of 23.98 and a ACP score of 6.945 in the combined dataset, showcasing its precision in ranking tasks. LightGBM and Stacked Regression also performed competitively, with LightGBM achieving a "3 of 5" score of 23.29 and a ACP score of 6.845, closely matching XGBoost's performance.

MAE / Model Type	T20	ODI	Test	Combined
XGBoost	112.931	**152.014**	**201.788**	**136.457**
TabNet	113.589	158.568	219.465	142.256
Linear Regression	113.329	158.995	214.130	146.568
Stacked Regression	**111.522**	158.995	219.282	139.456
Random Forest	115.456	156.536	224.459	148.457

Table 3: Evaluation of different models in the Distribution Pipeline

In the distribution pipeline, ensemble methods proved dominant, with **XGBoost consistently outperforming** other models. It achieved the lowest Match Mean Absolute Error (MAE) and Each Player MAE across T20, One Day, and Combined datasets, including a T20 Match MAE of 112.931 and a Combined Match MAE of 136.457, demonstrating its adaptability across match types. Stacked Regression followed closely, with a Combined Match MAE of 139.456, making it a strong contender but **less consistent and more complex** compared to XGBoost, potentially **limiting scalability**.

Linear regression **performed well in specific metrics**, such as the Mean Absolute Error (MAE) for each player in T20 matches, thereby validating the initial claim by Ravichandran et al. [6], which highlighted linear regression as a close contender to other non-linear models. **TabNet**, a deep learning model, **demonstrated moderate performance**, occasionally surpassing linear regression but falling short of ensemble methods. Among the ensemble methods, **Random Forest exhibited the weakest performance**, consistently recording higher error rates. LightGBM and CatBoost produced competitive results, often outperforming TabNet, linear regression, and Random Forest. Nonetheless, they lacked the consistency and overall accuracy displayed by XGBoost.

Phase \ Model Type	MAE	MAPE
Training	**118.275**	**19.267**
Testing	**136.457**	**21.345**

Table 4: Performance metrics of the final tuned model

5. CHALLENGES AND NEXT STEPS

Challenges and Limitations

01 Difficulty in Predicting Top Performers

One of the primary challenges was accurately predicting the top-performing players in every match. Player performance is inherently variable, particularly for standout performers, making it difficult for the model to consistently identify the top two players and predict their fantasy points accurately. This variability reduced the precision of the predictions for critical players.

02 Limitations in Fielding Data

Modelling fielding points posed another significant challenge. Due to the cricsheet dataset's limitations, the ball-by-ball data specified which players took catches or contributed to run-outs but did not indicate whether a run-out was a direct hit. This lack of detailed information directly impacted the accuracy of the model's fielding-related predictions, as Dream11 allocates points differently based on these specifics.

03 Challenges with Debutant Players

The absence of data for debutant players further constrained the model's capabilities. As new players lack historical performance records in the cricsheet dataset and the problem statement restricted the use of any additional data sources for matches, the models struggled to predict their contributions accurately. This limitation was especially problematic in matches featuring multiple debutants.

Next Steps and Future Improvements

01 Incorporating Team Composition Constraints

Future work could revolve around enhancing the predictions by aligning with Dream11's rules on team composition. While the current pipeline does not restrict predictions by the number of batsmen, bowlers, or all-rounders according to the problem statement, adding such constraints would increase the model's applicability for creating realistic fantasy teams.

02 Introducing Innings-Specific Predictions

Currently, the model predicts the dream team before the start of the match, without accounting for changes influenced by the toss or match progress. An extension to include innings-specific predictions could significantly improve performance. By incorporating real-time match data, which could potentially provide insights into players' current form and conditions, the model could dynamically adjust predictions. This approach would mirror Dream11's feature allowing team adjustments after the first innings and could enhance the model's relevance for real-world applications.

6. REFERENCES

[1] Mohith, S., Guha, R., Khetarpaul, S., Saurabh, S. (2022). Team Selection Using Statistical and Graphical Approaches for Cricket Fantasy Leagues. In: Guizzardi, R., Ralyté, J., Franch, X. (eds) Research Challenges in Information Science. RCIS 2022. Lecture Notes in Business Information Processing, vol 446. Springer, Cham, doi: 10.1007/978-3-031-05760-1_48

[2] Chakraborty, Sanjay & Mondal, Arnab & Bhattacharjee, Aritra & Mallick, Ankush & Santra, Riju & Maity, Saikat & Dey, Lopamudra. (2023). Cricket data analytics: Forecasting T20 match winners through machine learning. International Journal of Knowledge-based and Intelligent Engineering Systems. 28. 1-20. 10.3233/KES-230060.

[3] 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.

[4] 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.

[5] M. Balpande, K. Mahajan, J. Bhandarkar, B. Kapadne and G. Borse, "Machine Learning Based IPL Fantasy Cricket Dream11 Best Team Prediction," 2024 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2024, pp. 1-6, doi: 10.1109/ESCI59607.2024.10497335.

[6] 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.

[7] Patil, Nilesh M. and Sequeira, Bevan H. and Gonsalves, Neil N. and Singh, Abhishek A., Cricket Team Prediction Using Machine Learning Techniques (April 10, 2020). Available at SSRN: https://ssrn.com/abstract=3572740 or http://dx.doi.org/10.2139/ssrn.3572740

APPENDIX

Alpha Concept

Initially, we mentioned that the ranking pipeline alone could not accurately determine the complete fantasy points, particularly due to the exclusion of fielding contributions and the inability to capture exceptional individual performances. To address this limitation, we introduced the alpha adjustment model. The need for this model arose from the observation that individual models were not capturing fielding points or exceptional performances accurately, leading to incomplete player predictions. This challenge was exacerbated by the difficulty in attributing fielding points to specific players and the lack of detailed fielding data in the Cricsheet dataset. Consequently, the model struggled to fully reflect player performance, particularly regarding fielding metrics.

This model adjusted the prediction by adding a constant value to player fantasy points, with the constant varying for each player. This adjustment helped to more accurately incorporate fielding contributions and reduced the net Mean Absolute Error (MAE) in the dream team predictions. However, due to data limitations and the inability to accurately capture the performance of exceptional players, this model was unable to perform as expected. As a result, we ultimately transitioned to a new modelling approach, the distribution pipeline, to improve accuracy and overcome the limitations faced by the alpha adjustment model.

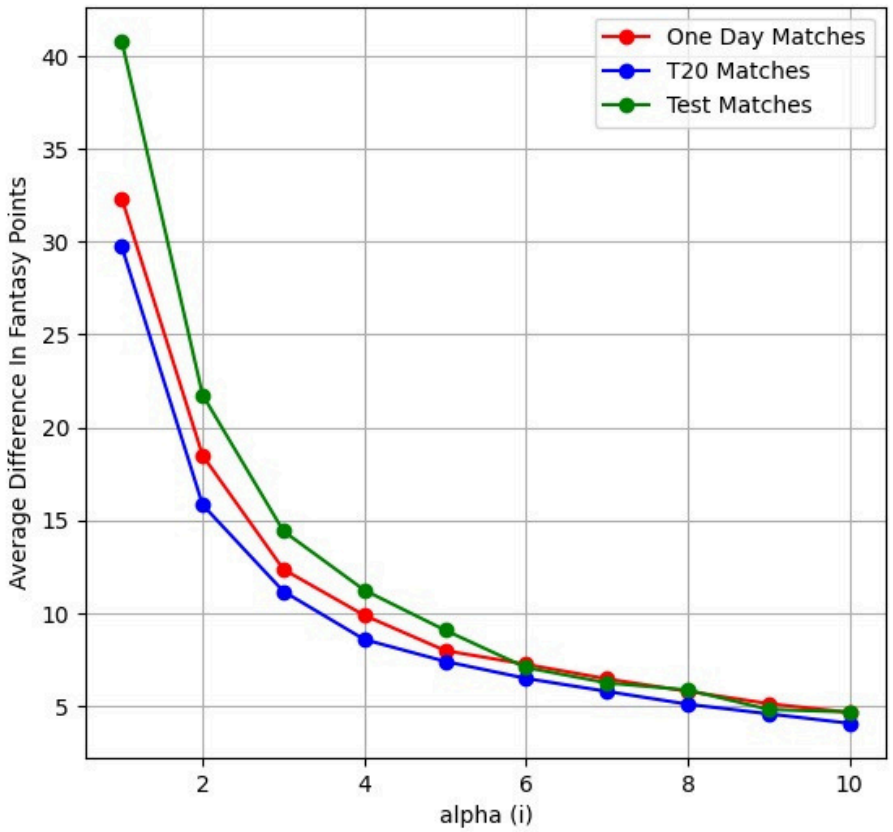


Figure 5: Alpha Values Distribution among Top Ranked Players across formats

Stacked Regression

In this model, we combine the predictive power of three different algorithms—XGBoost, CatBoost, and LightGBM—by using their outputs as input for a final Linear Regression model. This approach helps capture the strengths of each individual model, aiming to improve overall prediction accuracy.

IsolationForest Concept

The Isolation Forest model was employed to detect outliers in the dataset. However, instead of discarding these outliers, their associated outlier scores were retained and incorporated as an additional feature. This approach leveraged the Isolation Forest's strength in identifying anomalies, enriching the dataset with valuable insights about data points that deviated from standard patterns. By including the outlier scores, the model was better equipped to differentiate between typical and exceptional scenarios. This added layer of information enhanced the model's predictive capabilities, ultimately improving its accuracy.

Rule Based Method

In the rule-based method, players are ranked based on the average of their fantasy points scored in the last five matches, with those having the highest average points placed at the top. This approach assumes that recent performance is a strong indicator of a player's current form and potential. We have used this method as a baseline for the ranking pipeline to compare its effectiveness with more advanced models.

Performance Metrics

$$\text{3 out of 5 Common} = \frac{\sum_{i=1}^{n} \mathbf{1}\left(|\text{Top 5 Actual}_i \cap \text{Top 5 Predicted}_i| \geq 3\right)}{n}$$

$$\text{Average Common Players} = \frac{1}{N}\sum_{i=1}^{N} |A_i \cap P_i|$$

Percentage Distribution of Fantasy Points in Top 11 in Distribution Pipeline

Position	T20		ODI		Test	
	Mean	Std	Mean	Std	Mean	Std
1	14.42	3.71	15.06	3.65	14.98	3.98
2	11.30	2.25	11.74	2.88	11.62	2.29
3	9.46	1.52	10.56	2.67	9.81	1.60
4	8.25	1.17	8.93	1.84	8.51	1.24
5	7.30	1.00	7.31	1.59	7.51	1.05
6	6.53	0.92	6.27	1.08	6.65	0.94
7	5.83	0.84	5.29	1.01	5.91	0.88
8	5.22	0.82	4.74	1.24	5.25	0.86
9	4.68	0.78	4.08	1.00	4.66	0.85
10	4.21	0.75	3.66	0.87	4.14	0.84
11	3.77	0.74	3.17	0.75	3.65	0.82

Table 5: Statistical mean and standard deviation of player across different position in top 11

Features Table

Feature name	Explanation
Match Win Percentage in Last N	Percentage of matches won by the player/team in the last N matches.
Number of Sixes in Last N	Total number of sixes hit by the player in the last N matches.
Number of Fours in Last N	Total number of fours hit by the player in the last N matches.
Average Runs Scored in Last N	The average number of runs scored by the player in the last N matches.
Runs Conceded by Bowler in Last N	Total number of runs conceded by the bowler in the last N matches.
Economy of the Bowler in Last N	The average economy rate (runs per over) of the bowler in the last N matches.
Strike Rate of the Batsman in Last N	The average strike rate (runs per 100 balls faced) of the batsman in the last N matches.
Weighted Average of Runs	The weighted average of runs scored by the player, with more recent performances receiving higher weight.
Runs, Fours, Sixes, and Wickets at a Specific Venue	The total runs, fours, sixes, and wickets scored by the player at a specific venue.
Runs, Fours, Sixes, and Wickets Against a Specific Opponent	The total runs, fours, sixes, and wickets scored by the player against a specific opponent.
Batting Fatigue Score(Specifically for the test matches)	A composite metric to assess player fatigue based on various factors: (1) Batting Workload: Average batting points across all matches (weight: 0.05877676)(2) Running Between Wickets: Average non-boundary runs (weight: 0.04043447)(3) Match Frequency: Average days between matches for recovery (weight: -0.01322782)(4) Dot Balls: Average number of dot balls faced (weight: 0.62301173)
Bowler Performance Score (BPS)(Specifically for the test matches)	A composite metric used to evaluate a player's bowling performance in a match, considering factors such as wickets taken, economy rate, and impact on the game.
Batting/Bowling_point_exp_avg	Exponential Weighted Moving Average (EWMA) for batting/bowling points against each opponent.
Anomaly Score	Using Isolation Forest outlier score as a feature
Batting/Bowling Form	Batting avg/Bowling avg of a player across formats in last 10 matches

Winning_exponentially_decayed_bowling_points	Exponentially decayed average of batting points for each player in winning cause
Sixes_scored_till_date_on_this_venue_by_this_player	Sixes scored by a player at a particular venue
Wickets_taken_till_date_on_this_venue_by_this_player	Wickets taken by a player at a particular venue
Runs_scored_till_date_on_this_venue_by_this_player	Runs scored by a player at a particular venue
Cumulative_avg_runs_given	Runs given by a player against a particular opponent
Days_since_last_match	Days since last match of a player
Cumulative_average_bowling_points	Cumulative average of bowling fantasy points of a player till last match
Cumulative_average_batting_points	Cumulative average of batting fantasy points of a player

Table 6: Subset of features used for modelling

We used the above set of features in the ranking pipeline to assess player performance. For the distribution pipeline, we utilized the cumulative versions of these features.

Explanation: Classification Model

The player classification assigns players to one of three roles—Batsman, All-Rounder, or Bowler—based on their performance metrics. For ODI and T20 formats, a player is classified as a Batsman if their average wickets taken is below 0.5 or average overs bowled is less than 1. An All-Rounder has over 0.5 wickets taken, a batting average above 10, and a bowling average greater than 1. Any other player is categorized as a Bowler.

For Test format, a player is classified as a Batsman if their average wickets taken is below 1 or average overs bowled is less than 1. An All-Rounder has over 1.5 wickets taken, a batting average above 25, and a bowling average greater than 1. Players who don't meet these criteria but excel in bowling are classified as Bowlers. We only use the training data for player classification, ensuring no information from the test data is used in this process.

Mixture of Experts

To forecast fantasy points, we used a Mixture of Experts (MoE) model. We employed a variety of expert models, such as XGBoost, CatBoost, Decision Tree Regressor, and Linear Regression, each selected for their unique capabilities in processing various facets of the data. A neural network with a softmax output layer, the gating network dynamically allocates weights to each expert's prediction in response to input. In essence, this network was trained to determine which expert should have a greater impact on the final forecast for each input by optimising the weighting process. In contrast to employing a single model, the MoE technique enables a diverse set of predictions by integrating various models and using the gating network to weight their predictions. A weighted average of the expert forecasts, informed by the learnt weights of the gating network, is the end result.

Since alternative techniques produced better results and the typical neural network-like gating network, significantly reduces explainability, this modelling approach was not adopted for the final solution.

Product UI

Introduction

Dream11 aims to deliver an engaging user experience for a diverse use base, balancing simplicity for newcomers with advanced analytical features for experienced users. The Team Builder User Interface aims to address this need during the fantasy cricket team building stage of the user during the user journey, using technology to predict player performance, show advanced statistics and guide users through the interface in an engaging way. Ensuring accessibility and transparency is essential to drive engagement and long term retention.

Target User Persona

1. New Users: Seek guided experience and education with minimal complexity and intuitive features.
2. Casual Users: Understand the sport and tactics used, need intuitive guidance and access to key insights
3. Experienced Users: Desire more control over team selection and friction-less access to detailed statistics, player comparisons and customisation options.

Key Features

Recommended Team

The Recommended Team Feature provides users with an algorithmically curated team of 11 players based on advanced Machine Learning (ML) models. This feature simplifies team creation by giving a starting team to the users, and predicting player points with the help of historical player performance, match conditions, and contextual variables.

Player Details Feature

The Player Details feature provides users an in-depth view of each player's performance statistics in an easy to understand way. This feature also includes a Gen-AI based description and video which gives insight into player and their recent form.

Compare and Swap Players

The compare and swapping features allow users to compare two selected players side-by-side using various statistical parameters and charts in a user friendly way. After comparison, the user can choose to swap the players into their team. The feature can be accessed through multiple entry points, which improves the user experience and aids in swift team formation.

Recommended and Custom Team Report

The Recommended and Custom team report provides users with a comprehensive analysis of their selected fantasy team, offering detailed insights into the selected team's performance and how their team compares to the algorithmically recommended team. The Match overview section in the reports provide a glance at key features of top players, weather and pitch conditions of the upcoming match.

Fantasy Points System explanation

The Fantasy Points System explanation page feature provides users with a comprehensive breakdown of Dream11's fantasy game rules and criteria of the scoring system, including details for edge cases.

AI Chatbot

The GenAI Chatbot for Dream11 offers an interactive and personalized experience by providing real-time insights into player statistics, match context, and scoring rules. Leveraging Gen AI, the chatbot responds to user queries with natural language explanations, delivering tailored recommendations.

Measuring Success

The success of our product UI is measured by user engagement, ease of navigation and retention, ensuring enhanced user experience, and drives consistent interaction with the platform. Some KPIs considered were:

- User Retention Rate
- Daily Active Users (DAU)
- Chatbot Engagement Rate
- Average Session Duration