INTER IIT TECH MEET 13.0

Mid Prep Dream11 Final Submission Report

Team 53

Next-Gen Team Builder With Predictive AI

CONTENTS

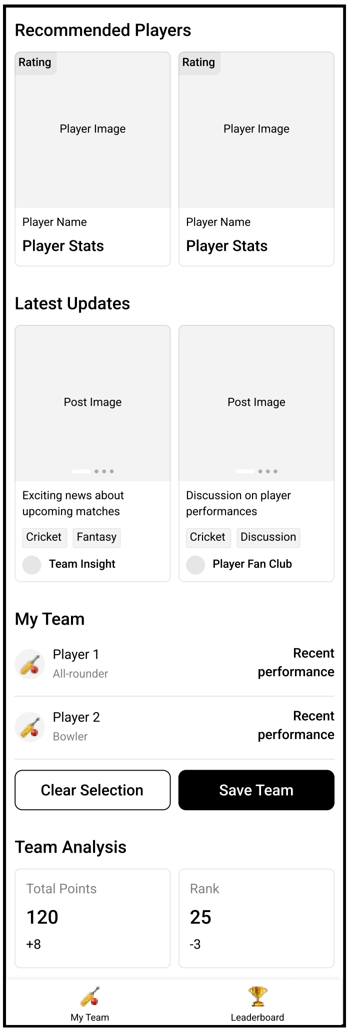
1.1 Problem Statement

Dream11 aims to enhance its fantasy cricket platform by improving the user experience and engagement in team creation. The challenge involves leveraging historical data and contextual factors to predict player performance effectively. Current functionalities, including access to player statistics and AI-powered recommendations, need to evolve to provide a more intuitive and strategic team-building experience for users, including real-time insights and guidance.

1.2 Objective

The task is to develop-

1. A robust ML model that predicts player performance using historical and contextual data while adhering to Dream11's constraints for fantasy team composition.
2. An interactive Python-based UI for:
    - **Users-** Offering personalized recommendations for selecting the best-performing team of 11 players, supported by audio/visual explanations.
    - **Evaluators-** Assessing the model's accuracy and effectiveness with detailed performance metrics.



First View Rough Wireframe

To work on

Model Development: Predictive Player Performance

- Build an ML model to predict player performance using historical data, match context, and relevant game variables.
- Adhere to strict constraints, including using only data before June 30, 2024, and following Dream11 team composition rules.
- Ensure model explainability by providing feature importance and clear rationales for player predictions.

User Interface (Product UI): Enhanced Team Selection Tool

- Develop an intuitive UI to recommend an optimal team of 11 players, aligned with Dream11 rules.
- Incorporate interactive features, such as audio/video descriptions, to assist users in understanding player insights.
- Ensure team recommendations are generated within 10 seconds for a seamless user experience.

Data-Driven Decision Support

- Integrate publicly available data, such as weather conditions and pitch reports, to enhance prediction accuracy.
- Combine contextual factors like stadium history with player statistics for more refined recommendations.
- Ensure all integrated data sources are accessible and well-documented.

2.1 Key Methodologies

Our approach combines advanced machine learning techniques and ensemble learning to predict the fantasy points of cricket players and form an optimized dream team of 11 players. The key steps are outlined below:

2.1.1 Modeling Approach:

We utilized two state-of-the-art regression models: LightGBM Regressor (LGBM) and XGBoost Regressor. Both models were trained separately to predict the fantasy points of individual players. The predictions from these models were then ensembled to enhance the overall accuracy and robustness of the final predictions.

2.1.2 Team Formation:

Based on the ensembled predictions, a dream team of 11 players was constructed from the pool of 22 players while adhering to given constraints (e.g., player roles).

- The player with the highest predicted fantasy points was designated as the captain.
- The player with the second-highest points was assigned the role of vice-captain.

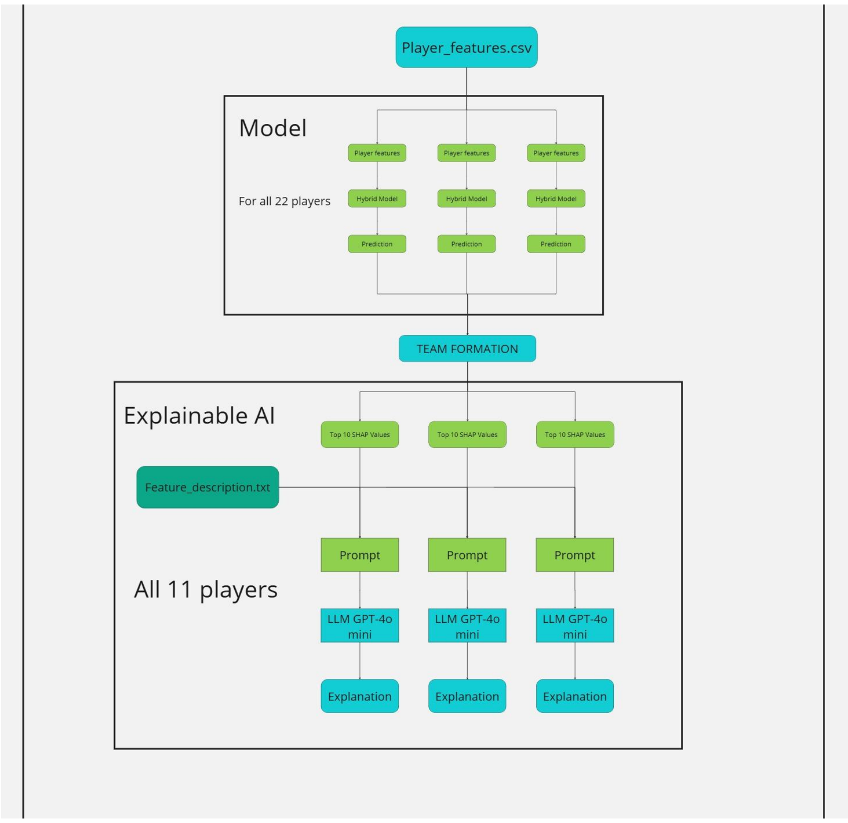
2.1.3 Model Explainability:

After forming the team, we focused on model explainability to provide transparency and insights into the decision-making process.

- A feature description file was created for all input features.
- SHAP (SHapley Additive exPlanations) values were computed to identify the top 10 features influencing the predictions for each player.
- These top features, along with their descriptions, were passed as input to the GPT-4o Mini API, prompting it to generate natural language explanations for why each player was selected in the final team.

This methodology ensures that both predictive accuracy and interpretability are prioritized, providing users with confidence in the results.

2.1.4 Pipeline:



2.2 Tools and Technologies

We employed a variety of tools and libraries for model training, data processing, and visualization:

- Machine Learning Frameworks:
  - LightGBM and XGBoost: For training regression models to predict player fantasy points.
  - SHAP: For calculating feature importance and enhancing model interpretability.
- Programming and Data Processing:
  - Python: The primary programming language for implementation.
  - Pandas and NumPy: For data preprocessing and manipulation.
  - scikit-learn: For auxiliary tasks like train-test splitting and evaluation metrics.
- Visualization:
  - Matplotlib: For exploratory data analysis and visual representation of feature importance.
- Integration with Generative AI:
  - OpenAI GPT-4o Mini API: For providing user-friendly explanations of model decisions, leveraging the power of generative AI for enhancing user understanding.

This combination of tools and methodologies ensured a seamless workflow, from data preprocessing to generating actionable insights for users.

3.1 Analysis of Fantasy Cricket and Dream11 Context

Dream11 is the world's largest fantasy sports platform, enabling users to create virtual teams composed of real-life players from various sports, such as cricket, football, and basketball. Users earn points based on the on-field performances of the players in their selected teams during live matches.

Rules of Dream11

Batsman	Bowler	All-Rounder	Wicket-Keeper
1-8	**1-8**	**1-8**	**1-4**

Players must include members from both competing teams.
Budget cap- 100 credits for the entire team.

Scoring System of Dream11

Batting	Bowling	Fielding	Bonus Points
Runs Scored	Wickets Taken	Catches	Runs Milestone
Strike Rate	Economy Rate	Run-Outs	Wickets Milestone
Boundaries	Dot Balls	Stumpings	Cap or Vice Cap

3.2 Challenges in Predicting

Unpredictability of Player Performance

- Real-time factors like injuries and form fluctuations.
- Limited historical data for emerging or returning players.

Balancing Model Explainability and Accuracy

- Making complex predictions understandable to users.
- Avoiding trade-offs that reduce prediction quality.

Handling Large and Diverse Datasets

- Integrating stats, venue, weather, and preferences .
- Ensuring completeness and consistency across datasets.

Optimizing Within Constraints

- Budget limits / role-based rules complicate optimization.
- Advanced techniques required to handle constraints.

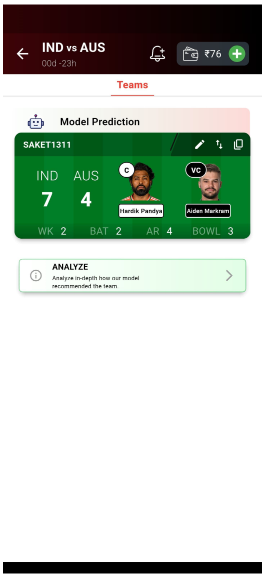
Real-Time Updates

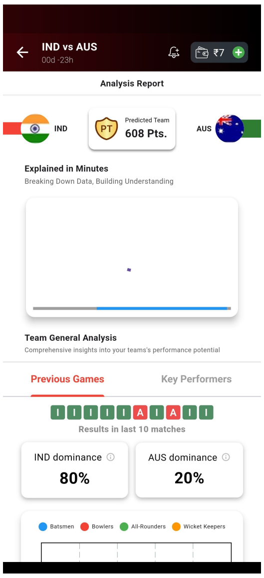
- Adapting to changes like toss results and lineup updates.
- Ensuring responsiveness during high-traffic periods.

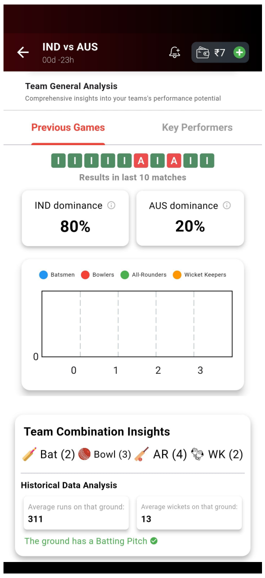
Creating Intuitive and Engaging UI/UX

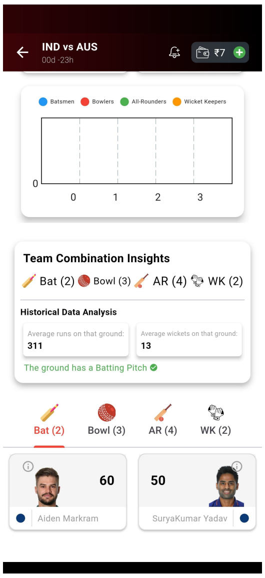
- Balancing detail for casual and advanced users.
- Incorporating interactive and explainable features .

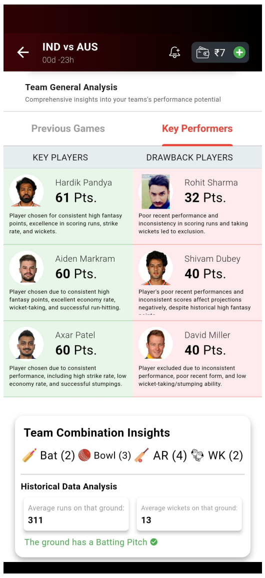
4.1 Product UI

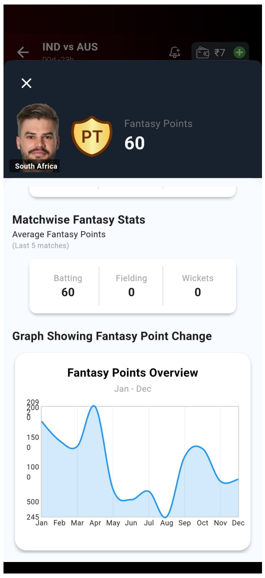


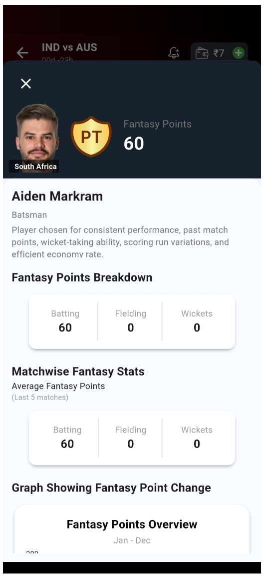


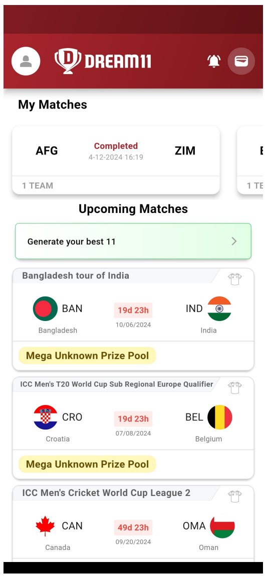












4.2 Model UI

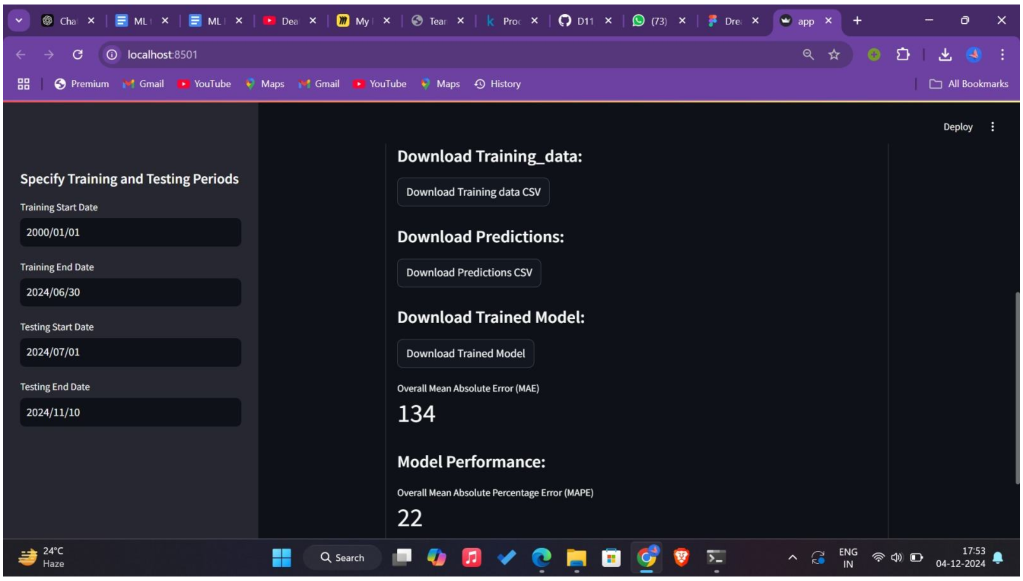
The ModelUI is a streamlined user interface developed using Streamlit to evaluate and validate machine learning model performance. It is hosted locally and can be accessed via the terminal using the ModelUI folder. The interface was designed to meet specific requirements for assessing models that predict fantasy cricket team performance.

4.2.1 Features of ModelUI

The ModelUI provides the following functionality, as displayed in the screenshot:

1. Training and Testing Period Definition:
   - Users can specify the training and testing periods by entering start and end dates.
   - Example:
     - Training Period: 2000/01/01 to 2024/06/30
     - Testing Period: 2024/07/01 to 2024/11/10
2. Data and Model Downloads:
   - Download Training Data: A button to download the processed training dataset in CSV format.
   - Download Predictions: A button to download the predictions as a CSV file containing the model's predicted outcomes.
   - Download Trained Model: Enables saving the retrained model file in .pkl format.
3. Performance Metrics:
   - Displays critical performance metrics:
     - Mean Absolute Error (MAE): Evaluates the error in total fantasy points for predicted versus actual dream teams.

Mean Absolute Percentage Error (MAPE): Quantifies the percentage error across predictions.



5.1 Data Preparation and Preprocessing

Used the cricksheet to prepare the data and further preprocess to make new features.
1. Match Context: Gather details like match date, teams, venue, and player roles.

- Fetch recent team compositions with did_player_play_last_match.

2. Player Statistics:
   - For each player, historical performance is fetched and features are generated using generate_features_for_single_date.
- Features include:
- Lag-based statistics (e.g., runs scored in the last 3 matches).
- Rolling statistics (e.g., rolling average, sum, variance).
- Cumulative statistics (e.g., cumulative mean, variance, and standard deviation).
- Contextual averages (e.g., performance at the venue, against opponents).

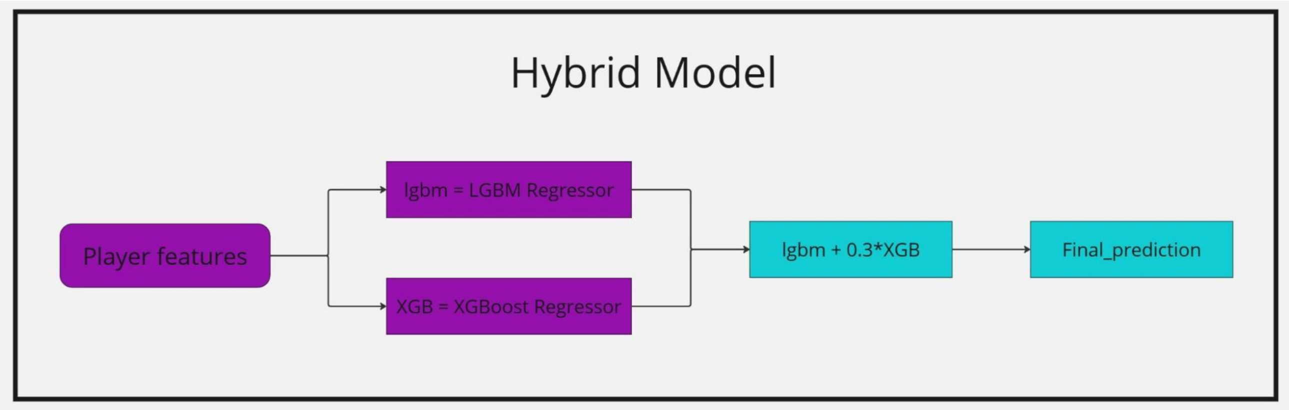
5.2 Training and Testing Methodology

In our training methods, we train the LGBM Regressor and XGBoost Regressor separately then ensemble their predictions to obtain the final prediction of fantasy points of a player. We have trained the model from 01-01-2000 to 30-06-2024

For testing and evaluating the model we used data from 01-07-2024 onwards.

The hyperparameters for the training the model are in the jupyter notebook(link).

Hybrid model or ensembled model of LGBM regressor and the XGBoost regressor:



5.3 Performance Metrics (MAE, MAPE)

The two used metrics for model evaluation are Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE):.

5.3.1 Mean Absolute Error (MAE)

$MAE = (1 / n) \Sigma |y_i - \hat{y}_i|$, where:

- n = number of matches,
- y_i = Dream Team points.
- $\hat{y}_i$ = Predicted Best Team Point.

5.3.2 Mean Absolute Percentage Error (MAPE)

$MAPE = (1/n)\Sigma|(yi-\hat{y}_i)/yi| \times 100$

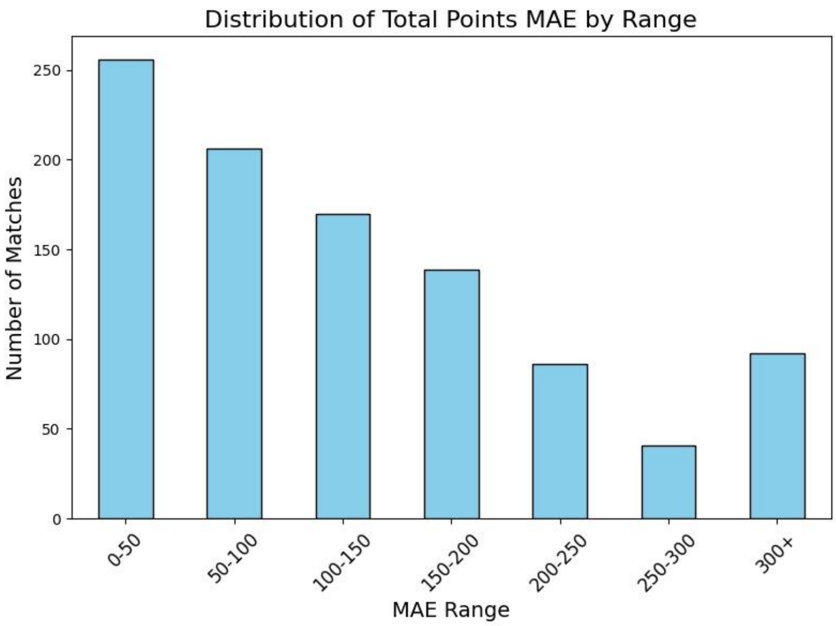
- n = number of matches,
- y_i = Dream Team points.
- $\hat{y}_i$ = Predicted Best Team Point.

6.1 Training Period Results (Data up to 30th June 2024)

Mean Absolute Error (MAE) = 38
Mean Absolute Percentage Error (MAPE) = 3

6.2 Testing Period Results (1st July 2024 – 10th November 2024)

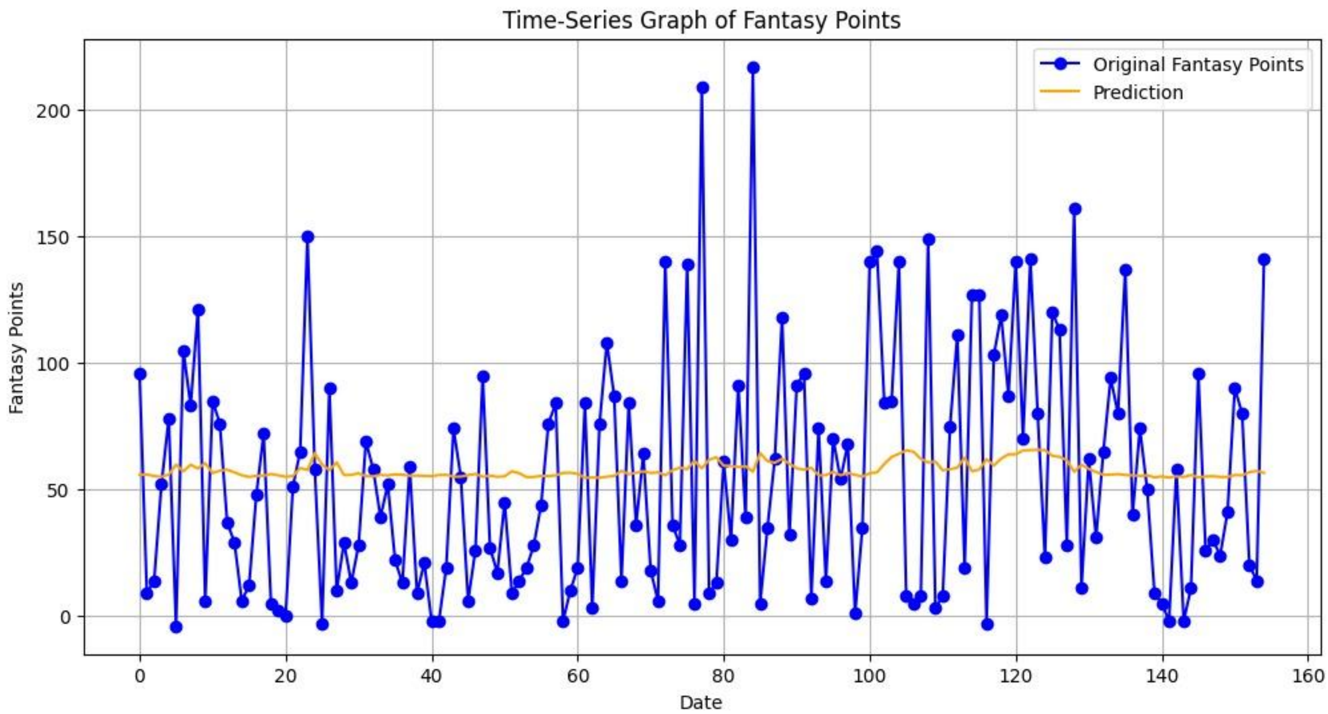
Mean Absolute Error (MAE) = 134
Mean Absolute Percentage Error (MAPE) = 22
MAE Distribution of testing period by Range:



7.1 Technical Challenges Faced

1. Data Extraction and Matching
   Integrating data from multiple sources like Cricsheet was challenging due to inconsistencies in formats, match dates, and player names.
2. Processing Large Datasets
   Handling massive datasets and creating new features, such as player trends, was computationally intensive and time-consuming.
3. Inconsistent Player Performance
   Cricket players' unpredictable performances made it difficult to forecast future outcomes reliably.
4. Unavailability of Player Roles
   Missing player roles complicated the application of Dream11 constraints for team formation, requiring additional heuristics.
5. High Variation in Fantasy Points
   Significant match-to-match variation in Dream Team fantasy points complicated model optimization and error analysis.

Despite these challenges, the system was designed to meet the required functionality effectively.



Inconsistency in player performance

7.2 Proposed Next Steps

1. Dynamic Match Context Predictions
    - Develop tools to adjust predictions based on live match conditions, tournament stages, and player role updates.
    - Provide venue-specific insights, such as expected runs and wicket trends.
2. Advanced Visualizations
    - Introduce radar charts for captain/vice-captain comparisons and dual-bar visualizations for form vs. potential analysis.
    - Add real-time graphical breakdowns of player fantasy points.
3. Global Market Expansion
    - Adapt the platform for international sports like soccer, rugby, and baseball with localized data and scoring rules.
    - Support region-specific leagues with custom fantasy scoring tools.
4. Advanced Modeling Approaches
    - Attention-Based Ranking Model: Use an encoder with multi-head attention to predict fantasy points and select top players.
    - Encoder-Decoder Transformer: Sequentially predict the best team while enforcing constraints, minimizing MAE between predicted and Dream Team points.
5. These steps aim to enhance predictions, expand reach, and improve user experience.

8.1 Summary of Findings

So we have trained and implemented different models for predicting the fantasy points of each players and here is the evaluation of each model.

Models	MAE per player
Hybrid model	12
LGBM Regressor	21
XGB Regressor	29
Encoder with embeddings from Neural Network	30
Fully connected Neural Network	31
xLSTM	40
LSTM	42