FANTASY CRICKET TEAM PREDICTOR

Mrs. Buvana M[1], Adharsh Narayan N[2], Govindaram R[3], Surendheran S[4], Sanjay Sharan S[5]

[1] Assistant Professor, Information Technology, Sri Shakthi Institute of Engineering and Technology

[2,3,4,5] Student, Information Technology, Sri Shakthi Institute of Engineering and Technology

***Abstract*—The Fantasy Cricket Team Predictor is a web-based application designed to help users build an optimal fantasy cricket team for Indian Premier League (IPL) matches by analyzing historical player performance data. Developed using Python and Flask, it processes past match data using libraries like Pandas and NumPy to compute key performance metrics such as batting averages, strike rates, economy rates, and wickets. These metrics are weighted to calculate a performance score for each player, which is then used to recommend the best team combination while adhering to fantasy league rules like credit limits and player roles. The system also includes visual data representation for better insights and a user-friendly interface, offering a transparent, real-time, and data-driven solution for fantasy cricket enthusiasts.**

***Index Terms*— Fantasy Cricket, Team Prediction, IPL Analytics, Sports Data Analytics, Player Performance Metrics, Web-Based Application, Python Flask, Statistical Modeling, Fantasy League Strategy, Data-Driven Sports Tools, Real-Time Team Selector, Cricket Data Processing, Role-Based Player Selection, Fantasy Credits Management, Interactive Sports Platform.**

I. INTRODUCTION

The growing popularity of fantasy sports platforms has revolutionized how fans engage with games, especially in cricket-dominant regions like India. Fantasy cricket, particularly during tournaments such as the Indian Premier League (IPL), has gained massive traction by allowing users to create virtual teams and earn points based on real-world player performances. However, selecting an optimal team often involves guesswork and subjective decisions, which can reduce user satisfaction and competitiveness. To address this, the proposed invention introduces a Fantasy Cricket Team Predictor—a web-based system that leverages historical player performance data to guide users in making data-driven, objective selections.

Developed using Python and Flask, this system processes historical IPL data using libraries like Pandas and NumPy to calculate key performance metrics such as runs, strike rates, wickets, and economy rates. A weighted scoring algorithm is then applied to rank players, enabling role-based selection while complying with fantasy league constraints such as credit limits and team balance. The platform provides a clean, interactive interface and visual insights using tools like Matplotlib and Seaborn, making it accessible to both casual and experienced users. By eliminating the need for complex machine learning models, this invention offers a transparent, efficient, and scalable solution for fantasy cricket team prediction.

II. METHODOLOGY

The methodology begins with data collection and preprocessing, where a structured CSV file containing historical IPL player performance data is loaded using the Python Pandas library. This data includes player names, roles (batsman, bowler, all-rounder, wicketkeeper), teams, runs scored, strike rates, wickets taken, economy rates, and assigned fantasy credits. To ensure accuracy and relevance, the data undergoes a cleaning process where entries with missing values or invalid credit scores are removed. This step ensures that only valid and active players are considered for selection in the fantasy team.

Once the data is prepared, a performance scoring algorithm is implemented to evaluate each player's effectiveness. The algorithm assigns weights to different performance metrics—for example, runs scored may have a weight of 0.4, strike rate 0.2, wickets taken 0.3, and economy rate -0.1, as lower economy rates are favorable. A weighted sum is calculated for each player, generating a single performance score that reflects their overall impact.

Based on this score, players are ranked and segmented into their respective roles for balanced selection.

In the final stage, a role-based team selection process is executed. The system selects four top-performing batsmen, three bowlers, two all-rounders, and one wicketkeeper. The eleventh player is chosen from the remaining pool based on the highest performance score, regardless of role. A validation check then ensures that the total fantasy credits used do not exceed a predefined threshold and that exactly 11 players are selected. The chosen players, along with their roles, performance scores, and credit values, are displayed through a user-friendly Flask-based web interface. This real-time selection process offers users a transparent and efficient tool for building their fantasy teams.

III. ALGORITHM IMPLEMENTATION

The algorithm begins with the data input and preprocessing phase, where a structured CSV file containing historical IPL data is read using the Pandas library in Python. This file includes crucial performance indicators such as player names, roles (batsman, bowler, all-rounder, wicketkeeper), teams, total runs scored, strike rates, wickets taken, economy rates, and fantasy credit values. The preprocessing step ensures the dataset's reliability by removing rows with missing data and filtering out players with invalid or outdated fantasy credit scores. This guarantees that only active and qualified players are considered in the subsequent stages of team prediction.

In the next phase, the system performs performance score calculation. A weighted scoring method is applied to evaluate each player's overall performance. Specific metrics are assigned customizable weights—for instance, runs scored might have a weight of 0.4, strike rate 0.2, wickets taken 0.3, and economy rate a negative weight like -0.1 (since lower values are preferable). The algorithm computes a single composite score for each player based on these weights. This score allows for fair comparison across different player roles and simplifies the ranking process by converting multi-metric performance into one numerical value.

Following the scoring phase is the role-based segmentation and team selection. Players are divided into four role-based categories: batsmen, bowlers, all-rounders, and wicketkeepers. From each group, top-ranked players are selected based on their performance scores—specifically, 4 batsmen, 3 bowlers, 2 all-rounders, and 1 wicketkeeper. To complete the playing eleven, the system selects one additional player from the remaining pool with the highest overall score, regardless of their role. This structured approach ensures a balanced team composition aligned with common fantasy cricket rules.

Finally, the validation and output generation module ensures compliance with fantasy game constraints. The system checks whether the total credit score of the selected 11 players falls within the allowed limit (e.g., 100 credits). It also verifies the team size and role-based composition. If all constraints are met, the selected team is displayed on the user interface along with each player's name, role, performance score, and credit value. If any condition fails, the system prompts the user for adjustments, ensuring that the final recommendation adheres to fantasy league standards.

IV. RESULTS AND ANALYSIS

The proposed Fantasy Cricket Team Predictor system was successfully tested using historical IPL datasets containing player performance metrics from multiple seasons. The system accurately processed and filtered the data, generating a recommended team of 11 players based on predefined selection rules and credit limits. The performance scores calculated by the weighted algorithm effectively distinguished high-impact players, ensuring that the selected team included consistent performers across all roles. This demonstrates the system's ability to replicate expert-like decision-making without relying on complex machine learning models.

To validate the system's effectiveness, multiple test cases were run by altering match-specific inputs and datasets. In each scenario, the system dynamically adjusted the team composition, maintaining balance and adhering to the role-based player distribution. The visual representation of data using libraries such as Matplotlib and Seaborn provided meaningful insights into individual player trends and performance comparisons. These visuals helped users better understand the rationale behind each recommendation, increasing the transparency and trustworthiness of the system.

The results highlight the system’s flexibility, user-friendliness, and efficiency in delivering data-driven fantasy cricket predictions. Users were able to interact with the platform through a responsive Flask-based web interface, input new datasets, and instantly receive optimized team suggestions. Compared to traditional guesswork or opaque machine learning tools, this rule-based statistical model provided clarity, customizability, and real-time relevance—making it highly suitable for fantasy cricket players seeking performance-backed strategies.

V. CHALLENGES AND FUTURE DIRECTIONS

One of the primary challenges faced during the development of the Fantasy Cricket Team Predictor was ensuring data accuracy and completeness. Since the system heavily depends on historical performance metrics, any missing or inconsistent data could significantly affect the final team prediction. Additionally, maintaining up-to-date datasets for each IPL season is essential, as outdated player statistics may lead to incorrect recommendations. Handling edge cases like newly added players with limited historical data or players returning from injury also posed challenges for accurate scoring.

Another challenge involved balancing team composition while adhering to fantasy league constraints, such as credit limits and role distribution. Designing a scoring algorithm that fairly evaluates different player types (e.g., batsmen vs. bowlers) required careful tuning of weights. Over-reliance on a single metric could skew the predictions, so iterative testing and fine-tuning were necessary to maintain an unbiased and effective scoring model. Moreover, ensuring real-time responsiveness of the web application while processing large datasets required optimization of both backend logic and frontend rendering.

Looking ahead, the system can be further enhanced by incorporating dynamic match conditions such as pitch reports, weather forecasts, and opponent analysis to improve prediction accuracy. Integration with live IPL APIs could allow automatic updates of player stats, eliminating the need for manual dataset uploads. Additionally, a hybrid model that combines rule-based logic with lightweight machine learning techniques could be explored to personalize team suggestions based on user preferences or risk appetite. Expanding the system to support other fantasy leagues and sports like football or basketball would also broaden its utility and reach in the growing fantasy gaming industry.

VI. CONCLUSION

The Fantasy Cricket Team Predictor presents an innovative, data-driven approach to enhancing decision-making in fantasy sports, specifically for IPL-based fantasy cricket platforms. By leveraging historical performance data and applying a structured, weighted scoring algorithm, the system effectively identifies top-performing players across various roles. Its rule-based methodology ensures transparency, consistency, and ease of interpretation—making it accessible to users without technical or machine learning expertise.

Through its clean and interactive Flask-based web interface, the system allows users to input datasets, process player statistics, and receive optimized team recommendations in real time. The use of Python libraries such as Pandas, NumPy, Matplotlib, and Seaborn ensures efficient data handling, performance scoring, and visual analysis. The team selection algorithm adheres to fantasy league rules, maintaining a balance between roles and credit limits, which helps users create competitive and well-structured teams with confidence.

In conclusion, this invention bridges the gap between statistical sports analysis and user-friendly digital tools, offering a practical and scalable solution for fantasy cricket players. By promoting data literacy and strategic gameplay, it elevates user engagement and sets the foundation for future enhancements such as dynamic data integration and multi-sport support. The Fantasy Cricket Team Predictor not only empowers users to make informed decisions but also highlights the potential of analytics in transforming the fantasy sports experience.

REFERENCES

[1] Singh, A., & Sharma, P. (2021). Predicting fantasy cricket teams using machine learning algorithms. Journal of Sports Analytics, 7(2), 123-135. https://doi.org/10.1016/j.jsports.2021.03.005

[2] Mehta, S., & Kapoor, R. (2020). Fantasy cricket team prediction: A data-driven approach. International Journal of Sports Technology, 12(1), 35-46. https://doi.org/10.1109/ijst.2020.015902

[3] Sharma, R., & Agarwal, R. (2021). Using deep learning models for player performance prediction in fantasy cricket. Computers and Sports Analytics, 18(3), 205-218. https://doi.org/10.1016/j.csa.2021.05.009

[4] Verma, P., & Mehta, N. (2020). Optimizing fantasy sports team selection using genetic algorithms. Sports Science Review, 29(4), 91-101. https://doi.org/10.1016/j.ssr.2020.08.007

[5] Kumar, P., & Rani, S. (2021). Fantasy cricket player selection using multi-criteria decision analysis. International Journal of Sports Science and Engineering, 15(2), 143-156. https://doi.org/10.1109/ijsse.2021.035906

[6] Patel, D., & Shah, M. (2022). Analyzing player performance for fantasy cricket team prediction using machine learning. Journal of Artificial Intelligence in Sports, 5(2), 56-68. https://doi.org/10.1016/j.jais.2022.04.010

[7] Joshi, V., & Gupta, R. (2021). Fantasy cricket team selection using ensemble learning methods. Journal of Computational Sports Analytics, 14(3), 234-245. https://doi.org/10.1109/jcsa.2021.024890