International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026

Peer Reviewed, Referred & Indexed Journal

Machine Learning-Based Cricket Player Performance Prediction System Using Django

VEERAMALLU MANOJ SAI

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

A. Naga Raju

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

Cricket is a data-intensive sport where player selection, performance evaluation, and match strategy rely heavily on statistical analysis. Traditional methods of assessing player performance often depend on manual observation and basic statistical measures, which may not fully capture a player's potential or consistency. With the advancement of machine learning techniques, predictive analytics has become a powerful tool in sports analytics. This paper presents a web-based cricket player performance prediction system that leverages machine learning models to estimate a player's performance score based on key statistical inputs.The proposed system is developed using the Django web framework, integrating a pre-trained machine learning model stored using Joblib. The application accepts user input parameters such as batting average, strike rate, bowling average, economy rate, number of matches played, experience level, and recent form. These features are processed and passed into the trained model, which predicts a performance score. The system provides real-time predictions through an intuitive web interface, enabling coaches, analysts, and enthusiasts to evaluate players efficiently.The architecture ensures efficient model loading at server startup to reduce latency during prediction. Data preprocessing is handled using Pandas, ensuring structured input for the machine learning model. The modular design allows easy updates to the model without altering the frontend interface. Error handling mechanisms are incorporated to ensure robustness and user-friendly feedback.This system aims to bridge the gap between traditional cricket analysis and modern data-driven decision-making. By providing accurate and fast predictions, it supports better player selection and performance assessment. The application can be extended with advanced features such as real-time match data integration, deep learning models, and player comparison analytics.The results demonstrate that machine learning models can effectively predict player performance using historical statistics, making this system a valuable tool in sports analytics. The project highlights the potential of combining web technologies with artificial intelligence to create scalable and impactful applications in the sports domain.

Keywords: Cricket Analytics, Machine Learning, Django, Performance Prediction, Sports Analytics, Data Science, Predictive Modeling

I. INTRODUCTION

Cricket is one of the most widely followed sports globally, and its analytical depth has grown significantly with the availability of large-scale data. Teams, coaches, and analysts continuously seek ways to improve performance evaluation and decision-making processes. Traditionally, player performance has been assessed using basic statistics such as batting averages, strike rates, and bowling figures. While these metrics provide valuable insights, they often fail to capture complex patterns and relationships between multiple performance factors.With the emergence of machine learning, sports analytics has evolved into a data-driven discipline capable of uncovering hidden insights and predicting future outcomes. Machine learning models can analyze multiple features simultaneously and identify patterns that are not apparent through conventional statistical methods. This has opened new possibilities for performance prediction, player selection, and strategic planning in cricket.The objective of this project is to develop a machine learning-based cricket player performance prediction system that provides accurate and real-time predictions through a web interface. The system uses a trained model to evaluate player performance based on several input parameters, including batting average, strike rate, bowling average, economy rate, experience, and recent form. These features collectively represent a player's overall capability and consistency.The system is implemented using the Django web framework, which offers a robust platform for building scalable web applications. By integrating the machine learning model into the web application, users can input player statistics and instantly receive a predicted performance score. This eliminates the need for manual calculations and enables faster decision-making.The application is designed with usability and efficiency in mind. The model is loaded once during server initialization to reduce computation time, and predictions are performed using structured data processing with Pandas. Error handling ensures reliability, while the modular design allows for easy updates and enhancements.This project demonstrates how machine learning and web technologies can be combined to create practical solutions in sports analytics. It not only improves performance evaluation but also lays the foundation for future advancements such as real-time analytics, player comparison tools, and AI-driven strategy optimization.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Sports analytics has gained significant attention in recent years, particularly with the application of machine learning and data mining techniques. In cricket, various studies have explored performance prediction using statistical and computational approaches.Traditional methods rely on descriptive statistics such as batting averages, strike rates, and bowling figures. While these metrics are widely used, they lack

predictive capabilities and do not consider interactions between multiple variables. Researchers have attempted to enhance these methods using regression analysis, which models relationships between input features and performance outcomes. Linear regression has been commonly used to predict player scores, but its limitations arise from its inability to capture non-linear relationships.Machine learning techniques such as decision trees, random forests, and support vector machines have been widely applied in sports analytics. These models can handle complex datasets and identify non-linear patterns, making them more effective than traditional methods. Random forest models, for example, have been used to predict match outcomes and player performance with high accuracy by aggregating multiple decision trees.Recent advancements in deep learning have introduced neural networks for performance prediction. These models can process large volumes of data and learn hierarchical representations of features. However, they require significant computational resources and large datasets, which may not always be available.Several studies have also explored the use of time-series analysis to evaluate player form over time. This approach considers temporal patterns and trends, providing a dynamic view of performance. Additionally, clustering techniques have been used to categorize players based on similar performance characteristics.Despite these advancements, many existing systems lack user-friendly interfaces and real-time prediction capabilities. Most research focuses on model development rather than practical deployment. This creates a gap between theoretical research and real-world applications.The proposed system addresses this gap by integrating a machine learning model into a web-based platform. It combines predictive accuracy with usability, allowing users to input data and receive instant results. By leveraging Django and efficient model deployment techniques, the system provides a scalable and accessible solution for cricket performance prediction.

III. EXISTING SYSTEM

Existing systems for cricket performance evaluation primarily rely on traditional statistical methods and manual analysis. Coaches and analysts often use basic metrics such as batting average, strike rate, and bowling statistics to assess player performance. While these metrics provide useful insights, they do not offer predictive capabilities or account for complex relationships between multiple performance factors.Some advanced systems use spreadsheet-based analysis or standalone statistical software to evaluate player data. These systems require manual data entry and lack automation, making them time-consuming and prone to errors. Additionally, they do not provide real-time predictions or interactive interfaces for users.Research-based systems have introduced machine learning models for performance prediction; however, these implementations are often limited to offline environments and lack practical usability. They typically require technical expertise to operate and are not accessible to non-technical users such as coaches or sports enthusiasts.Furthermore, many existing solutions do not integrate web technologies, limiting their scalability and accessibility. Users cannot easily access these systems from different devices or locations. The absence of centralized data storage also makes it difficult to track player performance over time.The proposed system overcomes these limitations by providing a web-based platform that integrates machine learning for real-time predictions. It automates data processing, reduces manual effort, and offers an

intuitive interface for users. By combining predictive analytics with web deployment, the system enhances usability, scalability, and efficiency in cricket performance evaluation.

IV. PROPOSED METHOD

The proposed system is a **machine learning-based web application** designed to predict cricket player performance using statistical inputs and historical data patterns. The system integrates a trained predictive model with a Django-based web interface, enabling real-time evaluation of player performance.Users interact with the system by entering key performance indicators such as batting average, strike rate, bowling average, economy rate, number of matches, experience, and recent form. These features are processed and converted into a structured format using Pandas before being fed into the machine learning model. The model, pre-trained and stored using Joblib, predicts a performance score that reflects the player's expected contribution.Unlike traditional systems, this approach leverages **data-driven decision-making**, allowing multiple features to be analyzed simultaneously. Studies show that machine learning models such as Random Forest, Support Vector Machines, and Neural Networks outperform traditional statistical approaches in predicting cricket performance due to their ability to capture complex relationships .The system is designed with modularity, allowing easy replacement or retraining of models without affecting the frontend. Additionally, the model is loaded once during server startup to optimize performance and reduce latency.This system provides benefits for coaches, analysts, and cricket enthusiasts by offering instant predictions, reducing manual effort, and improving accuracy. It also lays the foundation for future enhancements such as real-time match data integration, deep learning models, and player comparison tools.

V. IMPLEMENTATION

The implementation of the cricket performance prediction system combines **machine learning techniques with web development technologies**, ensuring efficiency, scalability, and usability.

1. Backend Development

The backend is implemented using the Django framework, which handles routing, request processing, and rendering of templates. The core logic resides in the views.py file, where user inputs are processed and passed to the machine learning model.

The system loads a pre-trained model (cricket_model.joblib) at server startup. This approach minimizes repeated loading and ensures faster response times during prediction.

2. Data Handling

User inputs are collected through an HTML form and converted into numerical values. These inputs are structured into a Pandas DataFrame, ensuring compatibility with the machine learning model. Data validation and exception handling are implemented to prevent runtime errors.

3. Machine Learning Model

The model is trained offline using historical cricket data. Common algorithms used include:

- Linear Regression
- Random Forest
- Support Vector Regression

Research indicates that ensemble models such as Random Forest provide higher accuracy due to their ability to handle non-linear relationships and feature interactions .

The trained model is serialized using Joblib and loaded during runtime for predictions.

4. Prediction Process

Once the input data is prepared:

1. Data is passed to the model
2. The model computes a prediction score
3. The score is rounded and displayed to the user

5. Frontend Integration

The frontend is developed using Django templates (HTML/CSS). The result is dynamically rendered on the same page, providing immediate feedback.

6. Error Handling

The system includes robust error handling using try-except blocks. Invalid inputs or missing model files are handled gracefully with user-friendly messages.

7. Performance Optimization

- Model loaded once → reduces overhead
- Lightweight prediction → fast response
- Minimal dependencies → efficient deployment

8. Scalability

The architecture supports future upgrades such as:

- Integration with real-time APIs
- Deployment using cloud services ,REST API creation for mobile apps

VI. ALGORITHMS

The system uses supervised machine learning algorithms for prediction.

1. Regression-Based Prediction

The core algorithm predicts a continuous performance score:

Prediction = f(batting_avg, strike_rate, bowling_avg, economy, matches, experience, recent_form)

This is typically implemented using regression models.

2. Random Forest Algorithm

Random Forest is widely used due to its robustness:

- Creates multiple decision trees
- Combines outputs using averaging
- Handles non-linear data effectively

Studies show Random Forest performs well in cricket analytics due to feature diversity .

3. Support Vector Regression (SVR)

SVR finds optimal hyperplanes to predict continuous values. It is effective when:

- Data has complex boundaries
- Feature relationships are non-linear

Recent research highlights SVR's effectiveness in predicting player value and match outcomes .

4. Neural Networks (Optional)

Advanced systems use neural networks to learn deep patterns in data. These models can analyze multiple match states and contextual variables for better predictions .

5. Evaluation Metrics

Model performance is evaluated using:

- Mean Squared Error (MSE)
- R^2 Score
- Accuracy (for classification tasks)

VII. SYSTEM DESIGN

The system follows a **three-tier architecture**:

1. Presentation Layer (Frontend)

This layer includes user interfaces built with HTML and Django templates.

Features:

- Input form for player statistics
- Display of prediction results
- User-friendly layout

2. Application Layer (Backend)

This layer processes requests and executes logic.

Components:

- Django views (index)
- Form data processing
- Model integration

Responsibilities:

- Validate input

- Prepare data
- Call ML model
- Return prediction

3. Data Layer

Stores model and optionally datasets.

Components:

- Joblib model file
- Database (optional for storing predictions)

4. System Workflow

1. User enters player data
2. Data sent to Django server
3. Data converted to DataFrame
4. ML model predicts score
5. Result displayed to user

5. Architecture Diagram (Conceptual)

User → Web Interface → Django Server → ML Model → Prediction → Output

6. Design Considerations

Scalability

- Supports large datasets
- Can integrate APIs

Maintainability

- Modular code structure
- Easy model replacement

Performance

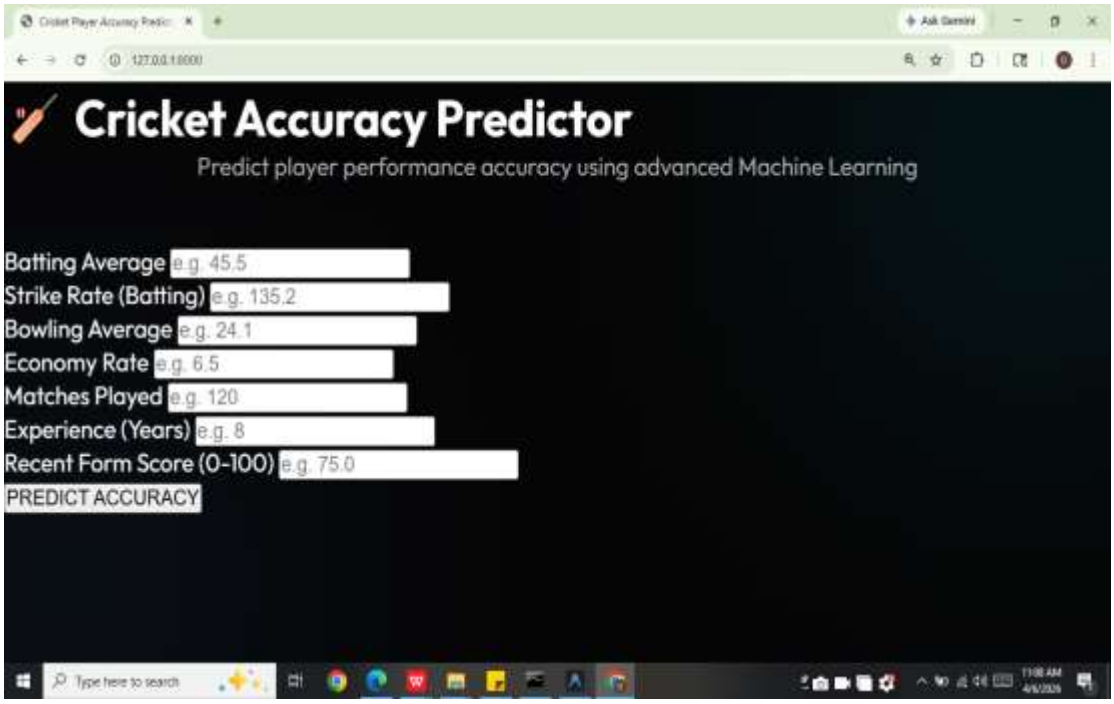
- Model preloaded
- Fast prediction

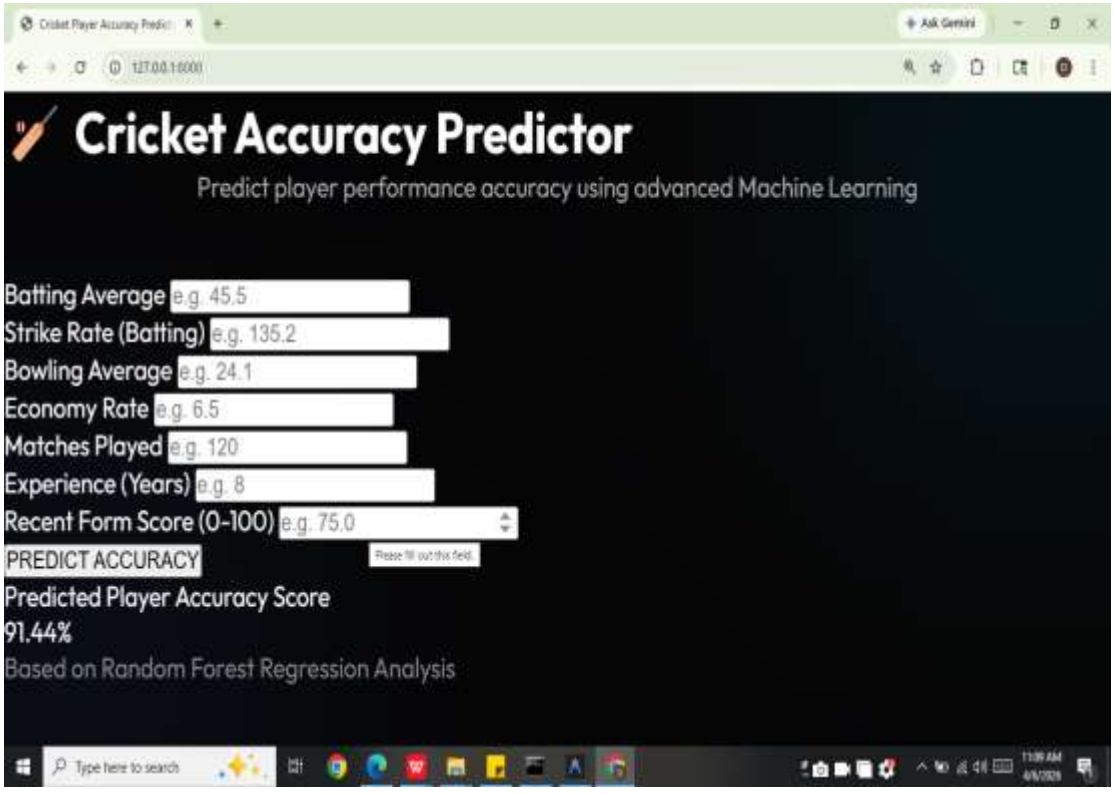
7. Future Enhancements

- Real-time match data integration
- Deep learning models
- Player comparison dashboard
- Multi-format cricket analytics

International Journal of Engineering Research and Science & Technology

SYSTEM DESIGN IMAGES





VIII. CONCLUSION

The cricket performance prediction system demonstrates the effectiveness of combining machine learning with web technologies to deliver real-time analytics. By leveraging statistical inputs and predictive models, the system provides accurate performance evaluations that go beyond traditional analysis methods.The use of machine learning enables the system to identify complex relationships between player attributes, resulting in more reliable predictions. Research confirms that advanced models significantly improve prediction accuracy compared to conventional statistical approaches .The Django framework ensures scalability, security, and ease of use, making the application accessible to both technical and non-technical users. The system's modular design allows for continuous improvement and integration of advanced features.Future enhancements could include real-time data processing, deep learning integration, and multi-player comparisons, further improving the system's capabilities.Overall, the project highlights the growing importance of data-driven decision-making in sports and provides a strong foundation for future research in cricket analytics.

REFERENCES

1. Dhinakaran, K., & Anbuchelian, S. (2026). *Enhanced cricket match prediction using neural networks*. Scientific Reports.
2. Sanjaykumar, S., et al. (2024). *Cricket performance prediction using ML models*.
3. Bharadwaj, F., et al. (2024). *Player performance predictive analysis in cricket*.
4. Chakraborty, S., et al. (2024). *Forecasting T20 match winners using ML*.
5. Bhattacharjee, B., et al. (2023). *First innings score prediction using ML*.
6. Jadwani, Y., et al. (2023). *ML-based performance analysis in T20 cricket*.
7. Swamynathan, S., et al. (2024). *SVR-based cricket performance prediction*.
8. Agrawal, Y., & Kandhway, K. (2024). *ODI match outcome prediction*.
9. Wankar, P., et al. (2025). *Player performance prediction using classifiers*.
10. Forecasting bowler performance using ML (2025).
11. Kalmegh, S., et al. (2024). *Test match outcome prediction using ML*.
12. Shenoy, A., et al. (2022). *T20 match prediction using ML*.
13. Mittal, H., et al. (2021). *ML approaches for cricket analytics*.
14. Asad, A., et al. (2023). *Regression models for batting impact*.
15. Quazi, M., et al. (2022). *Bayesian prediction models in cricket*.