Performance Analysis of ML Algorithm in Cricket Score Prediction

Mrs. P. Rupa Assistant Professor

G.AMULYA (22UP1A0558)

EMAIL: amulya@12gmail.com

G.SHABANAAZMI (22UP1A0554)

EMAIL:shabanaazim09@gmail.com

B.MEGHANA (22UP1A0515)

EMAIL:meghana33@gmail.com

C.AMRUTHA HARSHINI (22UP1A0507)

EMAIL:amruthaharshini992@gmail.com

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGYFOR WOMEN (An Autonomous Institution) (Affiliated to Jawaharlal Nehru Technological UniversityHyderabad) Kondapur (Village),Ghatkesar(Mandal),Medchal (Dist.) Telangana-501301

ABSTRACT

Cricket score prediction has emerged as a significant application of machine learning due to the increasing availability of structured match data and the growing interest in sports analytics. This study focuses on the performance analysis of various machine learning algorithms in predicting cricket scores, particularly in limited-overs formats such as One Day Internationals (ODIs) and T20 matches. The primary objective is to evaluate and compare the accuracy, efficiency, and reliability of different predictive models using historical match data.

The dataset typically includes features such as team statistics, player performance metrics, venue conditions, weather factors, toss results, and match situations (overs, wickets, run rate). After preprocessing and

feature engineering, multiple machine learning algorithms—including Linear Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and Gradient Boosting techniques—are implemented and trained on the dataset.

The performance of these models is assessed using evaluation metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R^2) score. Comparative analysis reveals that ensemble methods like Random Forest and Gradient Boosting generally outperform traditional regression models due to their ability to handle non-linearity and complex feature interactions. However, simpler models like Linear Regression offer faster computation and easier interpretability.

The study highlights the importance of feature selection and data quality in improving prediction accuracy. It also demonstrates how real-time factors, such as current match conditions, significantly influence prediction outcomes. The results suggest that hybrid approaches combining statistical methods and advanced machine learning techniques can further enhance prediction performance.

INTRODUCTION

Cricket is one of the most popular sports worldwide, attracting millions of fans and generating vast amounts of data from matches played across different formats such as Test, One Day Internationals (ODIs), and Twenty20 (T20). With the rapid growth of digital platforms and sports analytics, there is an increasing demand for intelligent systems that can analyze historical data and predict future outcomes. Cricket score prediction has thus become an important research area, helping teams, analysts, broadcasters, and fans make informed decisions and gain deeper insights into the game.

Traditionally, cricket analysis relied heavily on human expertise, intuition, and basic statistical methods. However, these approaches often fail to capture complex patterns and relationships present in large datasets. The introduction of Machine Learning (ML) techniques has revolutionized the field by enabling automated learning from historical data and improving prediction accuracy. ML algorithms can process multiple variables simultaneously, such as player performance, team composition, pitch conditions, weather, venue, and match situation, to generate reliable predictions.

Cricket score prediction is a challenging problem due to the dynamic and uncertain nature of the game. Factors such as sudden player performance changes, injuries, toss decisions, and real-time match conditions can significantly impact the outcome. Therefore, selecting appropriate features and algorithms plays a crucial role in building an effective prediction model. Various machine learning techniques, including Linear Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), and ensemble methods, have been widely used to address this problem.

LITERATURE REVIEW

Cricket analytics has gained significant attention in recent years due to the availability of large-scale match data and advancements in machine learning techniques. Several researchers have explored different approaches to predict cricket scores and match outcomes using statistical models and intelligent algorithms.

Early studies in cricket prediction primarily relied on traditional statistical methods such as regression analysis and probability models. Researchers used historical match data, including team performance and player statistics, to estimate future scores. Although these models provided basic insights, they were limited in handling complex relationships and non-linear patterns present in real-world cricket data.

With the advancement of machine learning, more sophisticated models have been introduced. Linear Regression has been widely used as a baseline model for predicting cricket scores due to its simplicity and interpretability. However, studies have shown that it often underperforms when dealing with highly dynamic match conditions. To overcome these limitations, Decision Tree-based models have been applied, which can capture non-linear relationships and interactions between variables such as wickets, overs, and run rate.

Further improvements were achieved using ensemble learning techniques like Random Forest and Gradient Boosting. Researchers have demonstrated that these models provide higher accuracy and robustness by combining multiple decision trees. Random Forest, in particular, has shown strong performance in handling large datasets with multiple features, while Gradient Boosting techniques like XGBoost have been effective in fine-tuning predictions and reducing errors.

PROBLEM DEFINITION

Cricket score prediction is a complex and dynamic problem that involves forecasting the total or intermediate score of a team based on various influencing factors. The primary challenge lies in accurately modeling the unpredictable nature of cricket matches, where multiple variables such as player performance, pitch conditions, weather, match format, and real-time game situations continuously change and impact the outcome.

The main problem addressed in this study is to determine how effectively different machine learning algorithms can predict cricket scores using historical and real-time data. Traditional statistical methods often fail to capture non-linear relationships and interactions between multiple features, leading to less accurate predictions. Therefore, there is a need to explore advanced machine learning techniques that can handle complex data patterns and improve prediction performance.

Another critical issue is feature selection and data preprocessing. Cricket datasets may contain irrelevant, missing, or noisy data, which can negatively affect model accuracy. Identifying the most significant features—such as current run rate, wickets lost, overs remaining, player statistics, and venue conditions—is essential for building an efficient prediction model.

Additionally, the variability in match formats (T20, ODI, Test) introduces further complexity, as each format follows different gameplay strategies and scoring patterns. A model that performs well in one format may not necessarily provide accurate predictions in another. This creates a need for adaptable and robust algorithms that can generalize across different scenarios.

PROPOSED SYSTEM

The proposed system aims to develop an intelligent and data-driven framework for accurately predicting cricket scores using advanced machine learning algorithms. This system is designed to overcome the limitations of traditional statistical approaches by incorporating multiple features, handling non-linear relationships, and enabling comparative performance analysis of different models.

The system begins with **data collection**, where historical cricket match data is gathered from reliable sources. This dataset includes detailed information such as team statistics, player performance (batting and bowling), match format, venue details, weather conditions, toss

results, current score, number of overs completed, wickets lost, and run rate. These features play a crucial role in building an effective prediction model.

In the **data preprocessing stage**, the collected data is cleaned and transformed to ensure quality and consistency. Missing values are handled, categorical variables (such as team names and venues) are encoded into numerical form, and irrelevant features are removed. Feature engineering techniques are applied to create meaningful attributes like required run rate, average player performance, and match pressure indicators.

The core of the system lies in the **machine learning model development** phase. Multiple algorithms are implemented, including Linear Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and Gradient Boosting methods. Each model is trained on the processed dataset and optimized using techniques such as cross-validation and hyperparameter tuning to achieve the best performance.

The system then performs **model evaluation and comparison** using performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R^2). This allows identification of the most accurate and efficient algorithm for cricket score prediction. Ensemble models like Random Forest and Gradient Boosting are expected to provide superior performance due to their ability to capture complex patterns.

SYSTEM ARCHITECTURE

The system architecture for cricket score prediction using machine learning is designed as a multi-stage framework that collects cricket data, processes it, trains multiple machine learning models, compares their performance, and generates final score predictions. The architecture begins with the data source layer, where historical cricket match data is collected from score databases, sports APIs, match records, and player statistics repositories. This data includes match format, team details, batsman and bowler performance, venue, toss result, overs, wickets, current run rate, and previous innings information.

The next layer is the data preprocessing layer, where raw data is cleaned and prepared for analysis. In this stage, missing values are removed or replaced, duplicate entries are handled, categorical data such as team names and venue names are encoded, and numerical features are normalized if required. Feature engineering is also performed here to

derive important parameters such as strike rate, economy rate, average runs per over, partnership score, and recent team performance. This layer ensures that the dataset is suitable for machine learning training.

After preprocessing, the data moves to the feature selection and dataset management layer. In this module, the most relevant attributes affecting cricket scores are selected using statistical analysis or feature importance methods. The dataset is then split into training and testing sets so that the algorithms can be evaluated fairly. This stage improves model efficiency and reduces unnecessary complexity.

SYSTEM ARCHITECTURE

IMPLEMENTATION

The implementation of the cricket score prediction system using machine learning involves a sequence of structured steps, starting from data collection and ending with score prediction and performance comparison. The system is designed to process historical cricket match data, train multiple machine learning models, and evaluate their predictive capability in order to identify the most accurate algorithm.

The first step in implementation is **data collection**. Historical cricket datasets are gathered from available sports databases, match score repositories, or CSV files containing records of previous matches. The collected data generally includes features such as team names, venue, match type, batsman statistics, bowler statistics, overs completed, wickets lost, current run rate, required run rate, toss decision, and innings score. These attributes serve as the input variables for the prediction models.

The next stage is **data preprocessing**, which is essential for improving data quality and model efficiency. In this phase, missing values are handled, duplicate records are removed, and categorical values such as team names and venues are converted into numerical form using encoding techniques. Numerical features may be normalized or standardized to ensure balanced model performance. Additional feature engineering is also carried out to create new derived variables like average runs per over, recent player

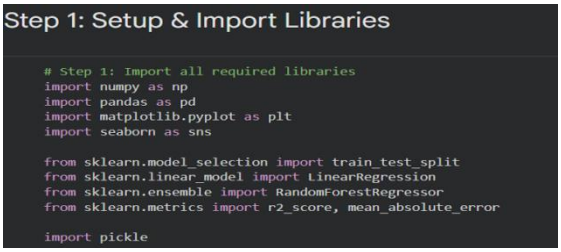
form, and partnership contributions, which can improve predictive accuracy.

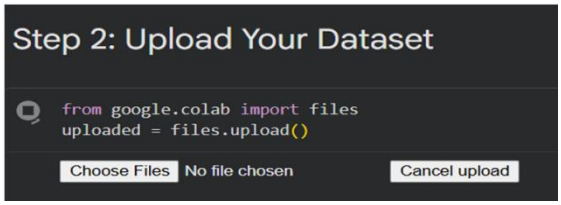
Once the dataset is prepared, it is divided into **training and testing sets**. The training set is used to teach the machine learning models, while the testing set is used to evaluate how well the models perform on unseen data. Common splitting techniques such as 80:20 or 70:30 ratios are used depending on dataset size. In some cases, cross-validation is applied to increase reliability and reduce overfitting.

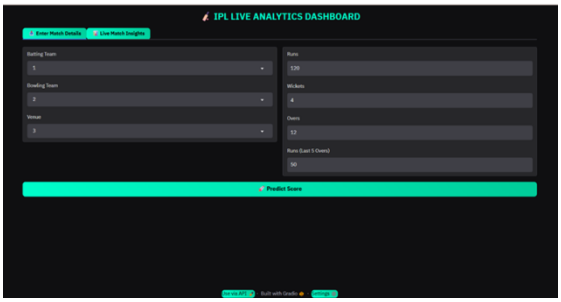
RESULTS AND DISCUSSION

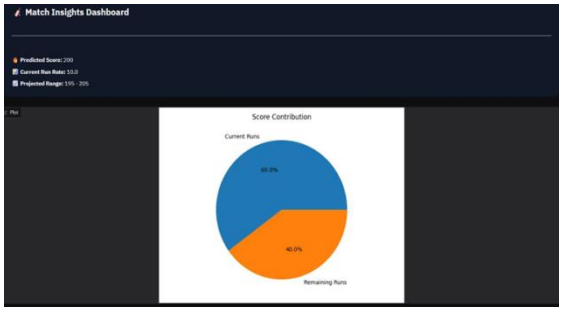
The results of this study demonstrate the effectiveness of machine learning algorithms in predicting cricket scores using historical and match-related data. Multiple models, including Linear Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and Gradient Boosting, were implemented and evaluated using performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared (R^2). The comparison of these models provides valuable insights into their predictive capabilities and suitability for cricket score prediction.

The experimental results show that **ensemble learning methods**, particularly Random Forest and Gradient Boosting, achieved the highest prediction accuracy. These models effectively captured complex and non-linear relationships between input features such as overs, wickets, run rate, and player performance. Random Forest provided stable and consistent predictions by reducing variance through multiple decision trees, while Gradient Boosting further improved accuracy by minimizing errors iteratively.









CONCLUSION

This study explored the application of various machine learning algorithms for predicting cricket scores and performed a comparative analysis to evaluate their effectiveness. By utilizing historical match data and key features such as overs, wickets, run rate, player performance, and match conditions, multiple models were developed and tested to determine their predictive capabilities.

The results indicate that machine learning provides a powerful and efficient approach for cricket score prediction. Among the evaluated models, ensemble techniques such as Random Forest and Gradient Boosting demonstrated superior performance due to their ability to capture complex, non-linear relationships in the data. Simpler models like Linear Regression, while less accurate, offered advantages in terms of speed and interpretability, making them useful as baseline models.

REFERENCE

- Tom M. Mitchell Machine Learning Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Education.
- Christopher M. Bishop Pattern Recognition and Machine Learning Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Trevor Hastie The Elements of Statistical Learning Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
- Predicting the Outcome of ODI Cricket Matches Using Machine Learning Shah, P., Shah, M., & Patel, V. (2015). Predicting the Outcome of ODI Cricket Matches Using Machine Learning Techniques. *International Journal of Computer Applications*.
- Cricket Score Prediction Using Machine Learning Kaluarachchi, A., & Varde, A. (2010). Cricket Score Prediction Using Machine Learning. *International Conference on Data Mining Workshops*.
- Scikit-learn Pedregosa, F., et al. (2011). Scikit-learn:

Machine Learning in Python. *Journal of Machine Learning Research*.

- TensorFlow

Abadi, M., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.

- Pandas

McKinney, W. (2010). Data Structures for Statistical Computing in Python.

- NumPy

Harris, C. R., et al. (2020). Array Programming with NumPy.

- ESPNCricinfo

ESPN Cricinfo. (2024). Cricket Statistics and Match Data. Available at: https://www.espncricinfo.com