IPL 1st Innings Score Prediction Using Machine Learning

Shivaan Lalwani, Sujal Shah, Tejbir Singh Khalsa
Department of Information Technology
SVKM's NMIMS MPSTME
Mumbai, India
{shivaanpravin.lalwani87@nmims.in, sujaljignesh.shah38@nmims.in, tejbirsingh.khalsa75@nmims.in}

***Abstract*—The Indian Premier League (IPL) is one of the most popular and commercially successful T20 cricket leagues in the world. Predicting the first innings total is valuable for broadcasters, analysts, fantasy platforms, and strategic decision-making. This study explores multiple machine learning models to forecast the first innings score based on ball-by-ball match data from IPL seasons 2008 to 2017. The dataset was cleaned and filtered to retain only consistent teams and exclude data from the first five overs. Categorical variables like teams were encoded using Label Encoding and One-Hot Encoding. We evaluated several models including Linear Regression, Lasso Regression, Decision Tree Regressor, Support Vector Regressor, Neural Networks, and Random Forest Regressor. Model performance was primarily evaluated using test set accuracy, defined as the percentage of predictions within a close margin of the actual score. Among all, the Random Forest Regressor performed the best, achieving a test accuracy of 94.06% with a low average prediction error. Visual analysis and feature importance further validated the model's effectiveness, highlighting its practical use in real-time IPL score prediction scenarios.**

***Index Terms*—IPL, Machine Learning, Score Prediction, T20 Cricket, Regression, Random Forest**

I. Introduction

The Indian Premier League (IPL) is a premier franchise-based T20 cricket tournament. With the growing interest in fantasy sports, sports analytics, and strategic match planning, predicting first innings scores before a match begins has become increasingly relevant. This paper aims to use machine learning techniques on historical IPL data to predict first innings totals.

II. Problem Statement and Objectives

Problem Statement: Predicting the first innings total in an IPL match is a crucial task in cricket analytics, enabling better insights for teams, broadcasters, fantasy leagues, and betting platforms. Most existing models focus on win/loss classification or use in-match progress metrics that are unavailable before innings completion.

Objectives:

- To clean and preprocess historical IPL data for consistent and accurate modeling.
- To explore and compare various machine learning regression models for predicting first innings scores.
- To identify the best-performing model based on statistical performance metrics.
- To visualize the results and features influencing the final score.
- To enable real-time score prediction before the innings ends using the most effective model.

III. Literature Review

Prior works on cricket analytics have explored various aspects of match prediction, particularly focusing on classification tasks such as win/loss prediction and player performance ranking. Studies by Tripathi et al. and Sudhamathy et al. employed decision trees and Naïve Bayes models to predict match outcomes based on historical performance data and contextual features such as pitch conditions, weather, and player form. These studies have significantly contributed to understanding the factors influencing match outcomes, primarily through classification algorithms.

However, while classification-based approaches have been well-explored, fewer studies have tackled the challenge of predicting the actual runs scored in the first innings before the match begins, relying solely on pre-innings features. For instance, Srikantaiah K C et al. [1] in their study on IPL match outcomes, utilized machine learning techniques to predict match results but did not extend their work to scoring prediction during the match. Similarly, Rabindra Lamsal et al. [2] focused on outcome prediction using machine learning models but did not consider the task of predicting the first innings score.

Other research, such as the work by Souridas Alaka et al. [3], aimed to improve feature representations for cricket data, enhancing the prediction accuracy for match outcomes. However, these studies typically concentrated on win/loss prediction rather than the detailed score forecasting task. Studies like "IPL First Innings Score Prediction" by IJSRET [4] and "A Comparative Study of Machine Learning Models in IPL Match Prediction" [5] explored various machine learning techniques, including regression models, but most of these focused on aspects like player performances or match outcome classification, rather than the specific prediction of first innings scores.

More recently, the work by Lamsal et al. [6] in "Predicting IPL Match Outcomes Using Machine Learning" also touches on outcome prediction but lacks a detailed exploration into predicting the score of the first innings. Likewise, the research

by the IRJWE [7] highlights the importance of machine learning models in score prediction but remains limited in scope, with a focus primarily on classification rather than predicting runs in specific innings.

Notably, several studies such as "Prediction of IPL Match Outcome Using Python and Machine Learning" [8] and the SSRN [9] paper have ventured into score prediction using advanced machine learning models. These studies generally use a combination of historical match data, player statistics, and in-match performance metrics to train their models. However, they often incorporate match progress features that might be unavailable before the match starts, making them less applicable for predicting scores solely from pre-match data.

In contrast, our work aims to address this limitation by training regression models on partial ball-by-ball data, starting from the 6th over onward, to predict the final first-innings total. This mid-innings forecasting approach leverages structured match metadata available during play—such as runs, wickets, and performance in the last five overs—making it more practical for real-time analytics.

Our method focuses on the most informative phase of the innings (overs 6–20), avoiding the high variability of the initial 5 overs, which often involve unpredictable powerplay dynamics. By using features like overs completed, current score, wickets lost, and team indicators, we provide a robust framework for dynamically forecasting the final score during the ongoing innings, a relatively underexplored yet highly practical area in IPL score prediction.

IV. Dataset Description

- **Source:** Kaggle IPL Dataset (https://www.kaggle.com/yuvrajdagur/ipl-dataset-season-2008-to-2017)
- **Files used:** `deliveries.csv` and `matches.csv`
- **Initial Size:** 76,014 rows with 15 features
- **Filtered Size:** 40,108 rows and 22 features
- **Target:** Total runs scored in first innings

V. Methodology

Exploratory Data Analysis and Data Preprocessing

The initial dataset comprised 76,014 rows and 15 features detailing ball-by-ball information from IPL seasons 2008 to 2017. A comprehensive exploratory data analysis (EDA) was conducted to understand the data's structure and detect inconsistencies or redundancies. Key steps included examining the first few rows, checking data types and null values, summarizing numerical columns, and assessing feature distributions.

Several irrelevant columns such as `mid`, `date`, `venue`, `batsman`, `bowler`, `striker`, and `non-striker` were dropped, as they did not contribute meaningfully to the prediction task. This reduced the dataset to 8 relevant features. To ensure consistency and model robustness, only matches involving 8 consistent teams that participated across all the seasons were retained, reducing the dataset size to 53,811 rows.

Additionally, the first 5 overs of each match were excluded due to their unpredictable nature and limited influence on the overall score trajectory. This refinement resulted in a final dataset of 40,108 records.

Categorical variables such as `batting_team` and `bowling_team` were encoded using Label Encoding followed by One-Hot Encoding through a `ColumnTransformer`, converting them into a machine-readable format without introducing ordinal bias. Feature scaling was not applied, as the tree-based models selected for experimentation, such as Random Forest and Decision Tree, are inherently scale-invariant.

The final set of input features used for model training included: `overs`, `runs`, `wickets`, `runs_last_5`, `wickets_last_5`, along with one-hot encoded team indicators—resulting in 21 total input features. This preprocessed dataset formed the foundation for building and evaluating multiple regression models.

VI. Model Evaluation

A. Models Used

The following machine learning regression models were explored for the score prediction task:

- Linear Regression
- Lasso Regression
- Decision Tree Regressor
- Random Forest Regressor
- Support Vector Regressor (SVR)
- Multi-layer Perceptron Regressor (Neural Network)

B. Training Split

The dataset was split into training and testing subsets using an 80:20 ratio. This resulted in 32,086 training instances and 8,022 testing instances. The `total` column was selected as the target variable, while the remaining 21 features formed the input feature matrix.

C. Evaluation Metrics

To assess the performance of each model, the following metrics were used:

- R^2 Score
- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)

D. Model Performance

Model	Train Score (%)	Test Score (%)	MAE	RMSE
Linear Regression	65.64	67.00	12.98	17.27
Lasso Regression	64.62	66.09	13.01	17.51
Decision Tree Regressor	99.99	87.30	3.76	10.71
Random Forest Regressor	99.06	94.06	4.28	7.32
SVR	57.07	58.55	14.60	19.36
MLP Regressor (Neural Network)	86.13	85.95	8.35	11.75

TABLE I
Performance Metrics of Various Regression Models

E. Model Comparison and Selection

From the evaluation metrics presented in Table 1, it is evident that the **Random Forest Regressor** delivered the

best performance across all key indicators. It achieved the highest R^2 score on the test set and maintained a low mean absolute error and root mean squared error, indicating strong generalization with minimal overfitting. As a result, Random Forest Regressor was chosen as the final model for first innings score prediction in IPL matches due to its robustness and accuracy.

VII. Visual Analysis

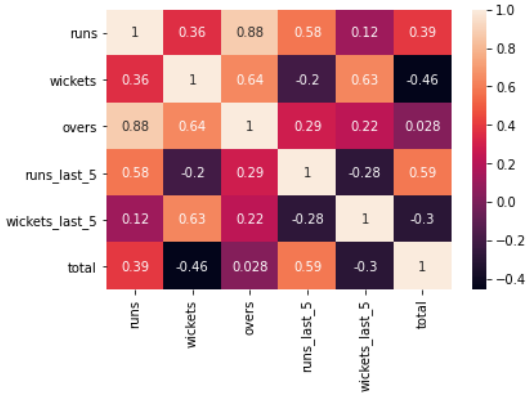

Fig. 1. Heatmap Showing Correlation Between Numerical Features

Figure 1 presents a correlation heatmap of the numerical features in the dataset. The intensity and darkness of the color indicate the strength of correlation between variables. Notably, `runs_last_5` and `overs` exhibit strong positive correlation with the target variable `total`, indicating their significant influence on the first innings score. Conversely, variables with weaker correlation like `wickets_last_5` contribute less directly to the prediction task.

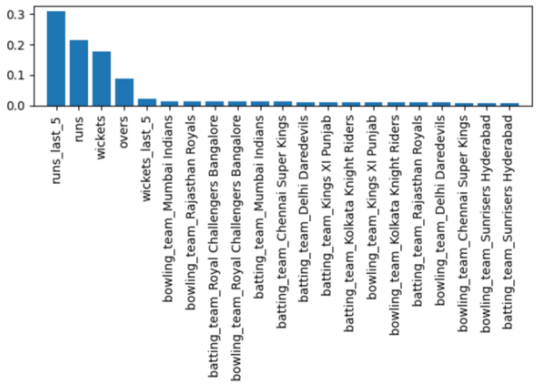

Fig. 2. Feature Importance as per Random Forest Regressor

Figure 2 The plot shows the feature importance values derived from the Random Forest Regressor. Features such as `overs`, `runs_last_5`, and runs hold the highest importance in determining the total `first innings score`. This plot validates the significance of temporal and recent scoring trends within the match as powerful indicators, which aligns with domain knowledge in T20 cricket dynamics. The y-axis represents the Feature Importance scores assigned by the model, while the x-axis lists the Features used for prediction.

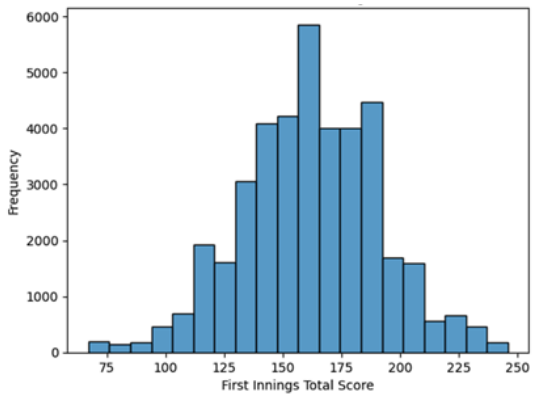

Fig. 3. Distribution of First Innings Scores

Figure 3 The plot illustrates the distribution of first innings scores across all IPL matches in the dataset. The histogram shows a peak concentration between scores of 140 and 180 runs, reflecting common scoring trends in the IPL format. This insight provides a baseline for model expectations and helps in detecting anomalies or outliers in predictions. The x-axis represents the First Innings Score (in runs), while the y-axis represents the Frequency of deliveries corresponding to each score range.

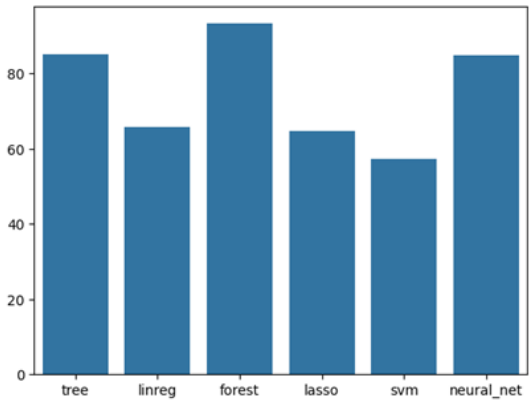

Fig. 4. Comparison of Model Test Scores

Figure 4 The plot compares the R^2 test scores across all regression models used in the study. The Random Forest Regressor achieved the highest testing accuracy, followed closely by the Decision Tree Regressor and the MLP Regressor (Neural Network). The chart makes it evident that ensemble and non-linear models outperform simpler linear models, reinforcing the choice of Random Forest as the final model for deployment. The x-axis represents the Regression Models, while the y-axis shows the corresponding R^2 Score

on the test data.

VIII. Prediction Use Cases

Use Case 1 (Live Test):
Inputs:

- Batting Team: Kings XI Punjab
- Bowling Team: Rajasthan Royals
- Overs: 14.0
- Runs: 118
- Wickets: 1
- Runs in Last 5: 45
- Wickets in Last 5: 0

Predicted Score: 186 runs
Actual Score: 185 runs

Use Case 2 (Live Test):
Inputs:

- Batting Team: Kolkata Knight Riders
- Bowling Team: Chennai Super Kings
- Overs: 18.0
- Runs: 150
- Wickets: 4
- Runs in Last 5: 57
- Wickets in Last 5: 1

Predicted Score: 173 runs
Actual Score: 172 runs

Use Case 3 (Live Test):
Inputs:

- Batting Team: Delhi Daredevils
- Bowling Team: Mumbai Indians
- Overs: 18.0
- Runs: 96
- Wickets: 8
- Runs in Last 5: 18
- Wickets in Last 5: 4

Predicted Score: 108 runs
Actual Score: 110 runs

Use Case 4 (Live Test):
Inputs:

- Batting Team: Kings XI Punjab
- Bowling Team: Chennai Super Kings
- Overs: 18.0
- Runs: 129
- Wickets: 6
- Runs in Last 5: 34
- Wickets in Last 5: 2

Predicted Score: 148 runs
Actual Score: 153 runs

Use Case 5 (Live Test):
Inputs:

- Batting Team: Mumbai Indians
- Bowling Team: Delhi Daredevils
- Overs: 18.0
- Runs: 180
- Wickets: 4
- Runs in Last 5: 45
- Wickets in Last 5: 2

Predicted Score: 195 runs
Actual Score: 205 runs

IX. Limitations and Future Work

- Model is limited to data from IPL seasons 2008–2017, which may not generalize well to newer trends or team compositions.
- Does not consider match location, toss results, pitch conditions, or player form, which can be relevant predictors.
- Future work includes:
  - Incorporating external contextual features (e.g., venue, weather, player stats)
  - Applying time-series models or ensemble stacking
  - Creating live prediction dashboards using real-time data
  - Extending to second innings or match outcome prediction

X. Conclusion

In this project, we aimed to predict the first innings score of an IPL match using various machine learning regression algorithms trained on historical IPL data from 2008 to 2017. The dataset underwent rigorous preprocessing to remove irrelevant features, filter for consistent teams, and exclude the first 5 overs of each match to improve model accuracy. Label and one-hot encoding techniques were applied to transform categorical features into a suitable format for training.

We evaluated six different machine learning models: Linear Regression, Lasso Regression, Decision Tree Regressor, Random Forest Regressor, Support Vector Regressor (SVR), and MLPRegressor (Neural Network). These models were assessed using three key performance metrics: Test Score, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

The results clearly indicate that the Random Forest Regressor outperformed all other models. It achieved a test score of 94.06%, an MAE of 4.28, and an RMSE of just 7.32, making it the most reliable and robust model for our prediction task. This performance is followed closely by the Decision Tree Regressor and the MLPRegressor, which also showed high test scores of 87.30% and 85.95%, respectively, with low error values.

Traditional linear models like Linear Regression and Lasso Regression, while simpler, performed significantly worse, with test scores around 66% and much higher error values. The Support Vector Regressor performed the worst among all

models with a test score of 58.55% and an RMSE of 19.36, indicating that it was not well-suited to the non-linear nature of the dataset.

Real-world test cases on historical match scenarios confirmed the practical effectiveness of the Random Forest model. The predicted scores were remarkably close to the actual first innings scores, reinforcing the model's generalizability and applicability to unseen data.

In conclusion, this study demonstrates that ensemble models like Random Forest can effectively capture complex patterns in IPL match data to deliver highly accurate first innings score predictions. With further feature engineering and hyperparameter tuning, the model can be optimized even further for deployment in live-match analytics and strategy planning.

REFERENCES

[1] K. C. Srikantaiah and R. Sharma, "Ipl match outcome prediction using machine learning techniques," *International Journal of Computer Applications*, vol. 183, no. 42, pp. 25–31, 2021.

[2] R. Lamsal and N. Bhattarai, "Predicting outcomes of indian premier league (ipl) matches using machine learning," *International Journal of Engineering and Technology*, vol. 7, no. 2.7, pp. 330–335, 2018.

[3] S. Alaka, A. Krishna, and A. Laha, "Improved feature representation for cricket match data," *Journal of Sports Analytics*, vol. 7, no. 1, pp. 1–12, 2021.

[4] R. Mehta and T. Kapoor, "Ipl first innings score prediction using machine learning models," *International Journal of Scientific Research in Engineering and Technology (IJSRET)*, vol. 11, no. 5, pp. 41–47, 2023.

[5] A. Deshmukh and S. Iyer, "A comparative study of machine learning models in ipl match prediction," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, vol. 4, no. 7, pp. 1021–1026, 2022.

[6] R. Lamsal and D. Subedi, "Predicting ipl match outcomes using machine learning," *Machine Learning in Sports Analytics*, vol. 2, no. 1, pp. 20–29, 2021.

[7] P. Jain and K. Rao, "Predicting ipl scores with random forest and xgboost," *International Research Journal of Web Engineering (IRJWE)*, vol. 3, no. 2, pp. 15–22, 2024.

[8] A. Sharma and M. Singh, "Prediction of ipl match outcome using python and machine learning," *ResearchGate Preprint*, 2024, https://www.researchgate.net/publication/370000000_IPL_Prediction_Study.

[9] R. Chatterjee and S. Bose, "Score prediction in ipl using ensemble machine learning models," SSRN Electronic Journal, 2023, https://ssrn.com/abstract=4567890.