IPL Score Prediction Using Machine Learning(The Boys,**Nikhil Sharma,**Kaavya Dhir,Jitesh Maurya)

**1. Introduction & Problem Statement**

Cricket is a highly unpredictable game influenced by multiple factors such as team composition, pitch conditions, and player form. Predicting match outcomes using **machine learning** enables analysts, coaches, and fans to gain deeper insights into match progressions.

This project aims to build an IPL score prediction model using a Random Forest Regressor, focusing on forecasting:

- Final score of the innings

- Total wickets lost

- Total boundaries scored

- Total dot balls bowled

The model utilizes ball-by-ball IPL match data to make accurate predictions, providing a

**data-driven approach** to understanding match scenarios.

## 2. Dataset Overview

The dataset used for this project is a publicly available \*ball-by-ball IPL dataset\* from Kaggle, containing:

- **Match ID, Innings, Ball Number** – Identifies match progression.

- **Batting & Bowling Team** – Determines match conditions.

- **Runs Off Bat, Extras** – Helps calculate total score.

- **Total Score, Wickets Lost** – Used as prediction targets.

- **Cumulative Runs & Wickets** – Tracks team performance.

-**Total Boundaries & Dot Balls** – Key metrics for batting impact.

The dataset spans multiple IPL seasons, making the model robust across various match conditions.

## 3. Technology Stack

The project is implemented using:

-Mandatory Libraries: numpy, pandas, matplotlib, seaborn, scikit-learn, statsmodels

- Optional: statsmodels for statistical validation

-Visualization Tools:matplotlib & seaborn

## 4. Machine Learning Model Implementation

Preprocessing Steps

**- Handling missing values & non-numeric data**

**- Feature Engineering: Extracting overs, cumulative runs, total boundaries, and dot balls**

**- Converting categorical team names into numerical codes**

**- Train-Test Split (80-20%)**

# Model Selection:

We use a **Random Forest Regressor,** which is well-suited for handling non-linear relationships in cricket data. The following target variables are predicted:

**1. Total Score**

**2. Wickets Lost**

**3. Total Boundaries**

**4. Total Dot Balls**

## Evaluation:

- **Root Mean Squared Error (RMSE)** is used to assess model performance.

- Feature importance is analyzed to understand key influencing factors.

## 5. Results & Insights

Predicted Outputs:

**Final Score** – Estimated total runs scored by the batting team.

**Wickets Lost** – Predicted number of wickets lost.

**Total Boundaries** – Predicted number of boundaries (4s & 6s).

**Total Dot Balls** – Predicted dot balls bowled by the fielding team.

## 6. Challenges & Future Improvements

Challenges Faced:

**- Handling massive data and ensuring proper feature transformations.**

**- Balancing model complexity and interpretability.**

**- Accounting for external factors (weather, pitch conditions) that influence IPL matches.**

## Future Enhancements:

- Implementing \*Neural Networks (LSTMs)\* for time-series cricket prediction.

- Player-specific impact analysis (e.g., batsman form, bowler economy rate).

- Win probability estimation based on real-time match updates.

## 7. Conclusion & Learnings

This project successfully demonstrates how \*machine learning can predict IPL match outcomes\* using \*ball-by-ball data\*. The model provides real-time match insights, making it valuable for analysts and broadcasters.

Key takeaways:

**ML models improve cricket analytics** by leveraging historical data.

**Feature selection is crucial** for accurate predictions.

**Future enhancements can incorporate advanced models** for better real-time performance.

## 8. References

- Kaggle IPL Dataset

- scikit-learn Documentation

-Supervised Machine Learning: Regression and Classification <https://www.coursera.org/learn/machine-learning>

-Videos that break down complex ML concepts into simpler explanations.

<https://www.youtube.com/user/joshstarmer>

-Demand Prediction with LSTMs using TensorFlow 2 and Keras in Python

<https://towardsdatascience.com/demand-prediction-with-lstms-using-tensorflow-2-and-keras-in-python-1d1076fc89a0>