## Indian Premier League Runs Prediction

### **1. Introduction**

In recent years, data-driven decision-making has found a growing place in sports, especially in cricket. Predicting match outcomes or individual statistics has become increasingly popular and feasible with the availability of rich historical datasets. This project aims to predict the total number of runs scored in a cricket match using machine learning techniques, particularly Linear Regression.

We used ball-by-ball IPL match data to build and evaluate a model that learns from past trends to predict future outcomes. The focus is on understanding how various features like teams, players, and match progress contribute to the overall run score.

### **2. Dataset Overview**

The dataset used in this project is the deliveries.csv file, which contains detailed information about each ball bowled in Indian Premier League (IPL) matches. It includes features like the batting and bowling team, over and ball numbers, runs scored (both regular and extras), and player information.

After initial inspection, we dropped columns that were not essential for predicting total runs (e.g., player\_dismissed, dismissal\_kind, fielder, and non\_striker). Categorical columns like team names and player names were encoded into numerical format using category codes to make them compatible with the machine learning model.

### **3. Exploratory Data Analysis (EDA)**

EDA was carried out to better understand the structure and relationships in the data. Some key steps and visualizations included:

* **Histograms**: To visualize the distribution of numeric variables like runs, over number, and extras.
* **Boxplots**: Used to detect outliers and compare runs across different teams.
* **Heatmap of Correlation Matrix**: To evaluate relationships between numeric features.
* **Team Performance Analysis**: A bar chart showing the average runs scored by different teams helped identify high and low-performing batting teams.

#### **Key Performance Indicators (KPIs):**

* **Average total runs per ball**: A general benchmark to understand scoring efficiency.
* **Most and least performing teams**: Based on total runs scored.
* **Distribution of extras**: Wides, no-balls, and byes were separately analyzed.
* **Most common bowlers and batsmen**: Helped understand player dominance.

These insights guided our feature selection and gave us confidence in the model's assumptions.

### **4. Data Preprocessing**

To prepare the dataset for modeling, we performed several steps:

* **Handling Missing Values**: Missing data was dropped or imputed where appropriate.
* **Encoding Categorical Variables**: Categorical features such as teams, batsmen, and bowlers were converted to numeric codes.
* **Feature Selection**: Features that did not contribute meaningfully to run prediction were excluded.

This step ensured the dataset was clean and machine-learning ready.

### **5. Model Building and Training**

We used the Linear Regression algorithm from scikit-learn for this task. It is a widely used algorithm for predicting continuous variables and provides interpretable coefficients.

The dataset was split into an 80% training set and a 20% test set. The model was then trained on the training set and evaluated using the test set.

### **6. Model Evaluation**

The model’s performance was evaluated using the following metrics:

* **Mean Absolute Error (MAE)**

MAE is approximately 0.000000000000026774, which is extremely close to zero, meaning your model's predictions are almost perfectly matching the actual values.

* **Root Mean Squared Error (RMSE)**

RMSE value is extremely small ( 3.2717e-14), indicating very minimal error and highly accurate predictions.

* **R² Score (Coefficient of Determination)**

R² is 1.0 means perfect prediction — the model explains 100% of the variance in the target variable.

In addition, a scatter plot of predicted vs. actual values was generated. The closer the scatter points were to the diagonal, the better the model's accuracy.

### **7. Feature Importance and Insights**

Linear Regression provides coefficients for each feature, which can be interpreted as the importance or weight of that feature in making predictions.

* **Important features** included: over number, batsman encoding, bowler encoding, and batting team.
* **Less impactful features**: like some of the binary indicators (wicket types, extras), had minimal contribution.

These insights are valuable in understanding match dynamics and strategic decision-making for teams.

### 

### **8. Conclusion**

This project successfully demonstrated that machine learning, particularly Linear Regression, can be used to predict total runs in a cricket match with reasonable accuracy. The model was simple yet interpretable, and served as a good foundation for future exploration.