# Cricket Match Analytics

Manideep G
*manideepg@iisc.ac.in*

Manish S B
*manishs@iisc.ac.in*

Brinal Jason Machado
*brinalm@iisc.ac.in*

Saurav Kumar Khandelwal
*sauravkk@iisc.ac.in*

Sandeep Kumar
*ksandeep@iisc.ac.in*

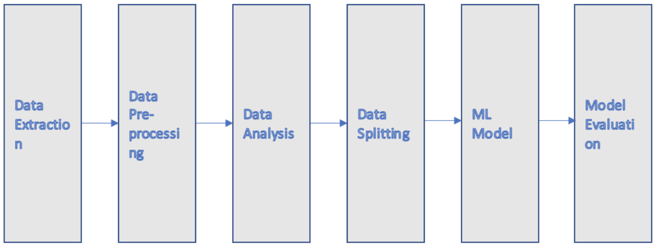

Fig. 1. High Level Design.

***Abstract*—This project aims to use statistical analysis and machine learning algorithms to identify patterns and insights that can be used to improve the performance of cricket players and teams. We collected and analyzed data from multiple cricket matches, including player performance statistics, ball-by-ball commentary, and weather conditions. We used various data analytics tools and techniques to identify key factors that influence the outcome of cricket matches.**

## I. Introduction

Applying Pandas for data processing and scikit learn for ML models to analyze cricket statistics and predict helpful information which can help improve players' or teams' performance.

## II. Motivation

Cricket is one of the most liked, played, and exciting sports. With the increasing number of matches with time, the data related to cricket matches and individual players are increasing rapidly.

This requires a proper advancement of data processing and analysis for utilizing the data effectively to use it in machine learning models for any business use cases, such as the selection process of players in the team, predicting the winner, predicting the team or individual score, etc.

## III. Approach

### *A. High Level design*

The high-level design approach for the problem statement is highlighted in Figure 1. The steps contain the proper hierarchy of the model-building processes, such as first - data extraction from the source; second - pre-processing of the data, such as cleaning up the irrelevant data; third - feature extraction needed for the model; finally, the model to predict the result.

### *B. Data source*

The data for this project is collected from Kaggle

## IV. Analysis

In our project analysis part, we have done statistical analysis on batsmen, bowlers, and team/individual performance for Indian Premier Leagues cricket matches. We have pulled the raw data from Kaggle, which is ball-by-ball metadata for each ball across all matches containing information about the happening for that particular ball, starting from the bowler, batsman, non-striker, score, whether it is a legal ball, etc. With the amount of information starting from 2008-2022, there is a huge scope of data analysis to be done on that data for more insights. This type of analysis is helpful during live telecasts. We have tried many such analytics in the following three categories and included a few of the insights in the form of graphs in our report.

### *A. Analysis on Batsman*

- Which players have the most sixes in IPL
- Which players have the most runs in boundaries
- Who are the highest run-scorers in IPL
- Which players have the best strike rate in death overs

### *B. Analysis on Bowlers*

- Most expensive overs in IPL
- Players with Most Wickets in IPL
- Which player has conceded the least number of runs when bowled all four overs
- Which bowler has the best economy rate in IPL

### *C. Analysis on Teams*

- Percentage toss win between two teams
- Percentage match wins head to head
- Head to Head win stats at a venue
- Winning percentage when the team wins the toss

## V. Prediction Model

As part of the project prediction part, we built a classic Linear Regression model using the scikit-learn library to predict batsman scores and bowlers' economy based on the data analyzed above. Two LR models have been built to predict individual batting scores and bowling averages.

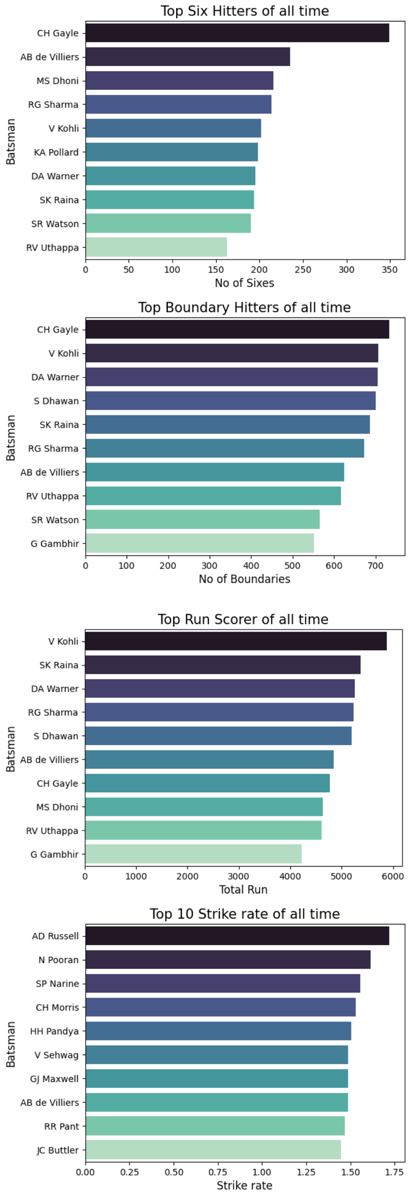

Fig. 2. Batsman Analysis

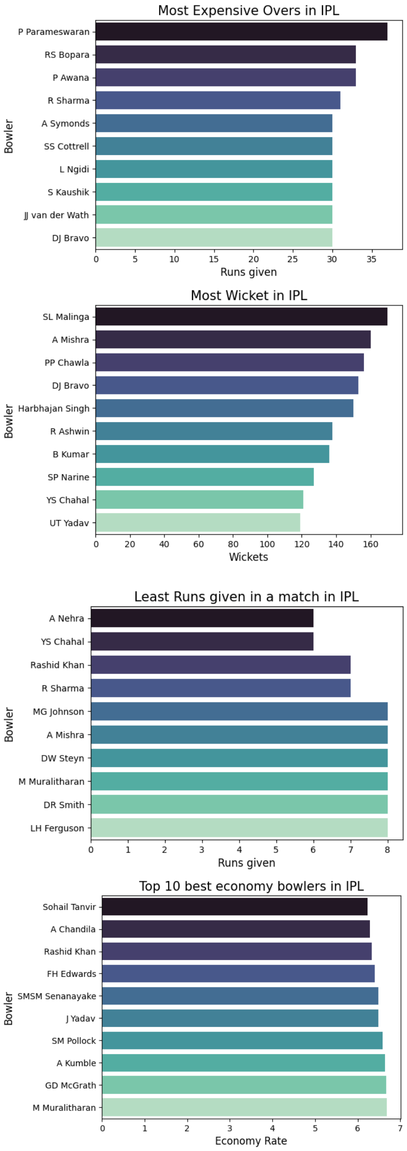

Fig. 3. Bowler Analysis

### A. Batsman Score Predictor

We have used the scikit-learn Gradient Boosting Regressor model and Random Forest Regressor models to predict batsman scores with the below features.

1) Features
   a) Balls faced
   b) Innings
   c) Fours
   d) Sixes
   e) Batsmen Avg
   f) Total runs
   g) Strike rate
2) Label
   a) Runs per match

## VI. Performance metrics and Evaluation

We have divided the data into two parts with the split ratio of initial 80% matches for training and rest matches for 20% and test. While the model is trained with only train data and test data is used for evaluation purposes. The performance metrics for the model are shown below.

TABLE I
Model performance metrics on Test data

| Model | RMSE | R2 |
|---|---|---|
| Linear Regression | 4.14 | 0.85 |
| Random Forest | 3.40 | 0.90 |
| Gradient Boost | 2.82 | 0.93 |

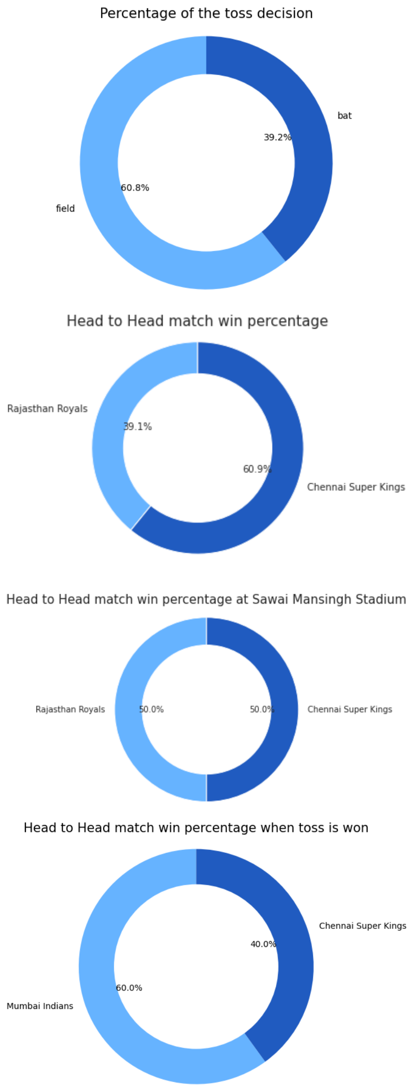

Fig. 4. Match Analysis

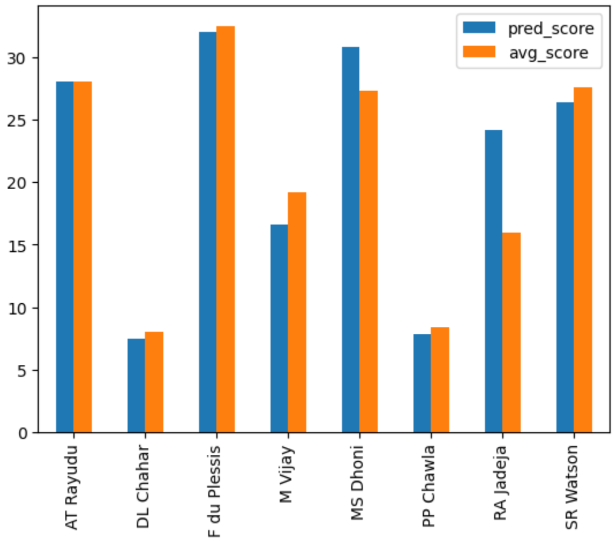

Fig. 5. Result: Batsman Run Prediction

## VII. Summary

Historical data can provide real-time analysis of Cricket Matches, which can provide insights related to matches and players. Pandas and scikit-learn combined are very powerful tools to do this task efficiently. We have analyzed only for IPL T20 matches, but the same can be extended for other formats. We wanted to build a single model to predict the match winner based on the batting and bowling prediction. But because of the volatility of the format of the game, which hugely depends upon the venue and playing condition, there needs to be a complex prediction model to predict which number of overs bowled by the bowler, etc., comes into consideration and the same must be the feature for the model. We tried a naive team winner predictor by adding the total predicted score for each batsman and comparing the score the predict the winner.