Machine Learning Project

IPL First Inning Score Predictor

Submitted by:

Sanyuja Shelke

Guided by:

Mayur Salunke

Table of Contents

Introduction

Cricket, being one of the most popular sports globally, attracts enormous attention and enthusiasm, particularly in tournaments like the Indian Premier League (IPL). The unpredictable nature of the game makes score prediction a challenging yet captivating task. With the advancement of machine learning techniques, predictive models have become an integral part of strategizing and understanding the dynamics of cricket matches.

This project focuses on predicting first inning scores in IPL matches using machine learning algorithms. Leveraging historical match data, including various factors such as venue, team composition, batting order, and more, the aim is to build a robust model that accurately forecasts the total runs scored by a team in the first innings.

The dataset used in this project comprises a comprehensive collection of match statistics, encompassing player performances, match venue details, toss decisions, and other crucial parameters. Through exploratory data analysis (EDA), feature engineering, and model development, this report outlines the methodology, insights gained, challenges faced, and the model's predictive performance.

The ultimate goal is to create a reliable model that aids cricket enthusiasts, analysts, and team strategists in anticipating first inning scores with a high degree of accuracy, contributing to a deeper understanding of match dynamics in the thrilling realm of IPL cricket.

Objective

1. To develop a regression model predicting IPL first inning scores.
2. To deploy a user-friendly interface for IPL score prediction based on the trained machine learning model.

Problem Statement

Develop a machine learning model that accurately predicts the total runs scored by a team in the first innings of an Indian Premier League (IPL) cricket match. Utilize historical match data encompassing various match-specific factors, team compositions, player statistics, venue details, and other relevant information to forecast the total runs scored by a team in the initial innings of an IPL game.

Statistical Tools & Software

Exploratory Data Analysis : Histogram, Box plot, Bar Plot, Heatmap, Count plot, Correlation

ML models :

Linear Regression
Random Forest
Support Vector Machine(SVM)

Software :

Python
Jupyter Notebook
VS code

Methodology & Data Description Data Collection

Data Sources :

The primary dataset used for this project comprises historical IPL match data, including details about matches played, teams involved, venue information, toss decisions, innings, runs scored, wickets taken, and other relevant match-specific information. The dataset includes IPL match records from 2008 to 2018.
The dataset was obtained from a public repository on GitHub, specifically https://raw.githubusercontent.com/SarangGami/IPL-First-Innings-Score-Prediction/main/ipl.csv

Data Description :

mid	A unique identifier for each match in the dataset.	Numerical
date	The date when match was played.	Date
venue	The stadium or location where the match was played.	Categorical
bat_team	The batting teams involved in the match.	Categorical
bowl_team	The bowling teams involved in the match.	Categorical
Batsman	Specific players involved in batting.	Categorical
bowler	Specific players involved in bowling.	Categorical
Runs	The total runs scored by the batting team in the match.	Numerical
wickets	The total number of wickets taken by the bowling team in the match.	Numerical
overs	The total number of overs bowled in the match.	Numerical
Runs_last_5	Cumulative runs scored in the last 5 overs of the match.	Numerical
Wickets_last_5	Cumulative wickets taken in the last 5 overs of the match.	Numerical
striker	The current batsmen on strike and non-strike ends during the match.	Numerical
non-striker	The current batsmen on strike and non-strike ends during the match.	Numerical
total	The total score or target set by the batting team for the opposing team.	Numerical

Data Cleaning :

Columns Removed: The columns 'mid' and 'date' were eliminated from the dataset as they did not contribute to the predictive modeling process.
Transformation: Outdated team names were updated to their current equivalents to ensure consistency and accuracy in team identification throughout the dataset.
Label Encoding: Categorical variables were encoded using label encoding techniques to convert them into numerical format suitable for machine learning algorithms.

Exploratory Data Analysis

Summary Statistics: Mean, median, mode, standard deviation, range for numerical features.

Correlation Analysis: Investigate the relationships between different features and the target variable (total runs scored) using correlation coefficients or pair plots.

Feature Importance: Identify significant features that seem to have a strong influence on the target variable.

Visualization:

Histograms/Boxplots: Display distributions and identify potential outliers.

Scatterplots: Explore relationships between key features and the target variable.

Heatmaps/Correlation Matrices: Illustrate correlations among features.

Bar Charts: Display categorical variables to visualize frequencies.

Feature Engineering

Identified 12 features highly correlated with the target variable (first inning scores) using heatmap visualization. Features that exhibited strong correlations were retained for model building.

Utilized label encoding to transform categorical variables like venues, team names, and player names into numerical labels. Applied label encoding to maintain ordinal information when converting categorical data, ensuring each category was represented by a unique integer. Implemented the label encoding process using libraries like scikit-learn's LabelEncoder to prepare the data for modeling.

Model Development

1. Model Selection - Explored various regression models such as Linear Regression, Random Forest Regression, and Support Vector Machine.
2. Model Evaluation - Assessed models using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared to evaluate their predictive performance.
3. Feature Importance - Determined the most significant features using correlation technique to understand which features contribute most to predicting first inning scores.
4. Model Performance - Identified the best-performing model based on evaluation metrics and feature importance analysis.

Model Deployment

Frontend Development:

The developed machine learning model for predicting first inning scores in IPL matches has been integrated into a user-friendly web application. The frontend of the application has been developed using HTML, CSS, and JavaScript to provide an interactive interface for users to input match-specific data and receive predicted scores.

User Interface Design:

The user interface has been designed to be intuitive and easy to use. Users are prompted to input relevant match details, such as venue, batting and bowling teams, player statistics, and other match-specific information.

Integration with Flask:

The backend functionality is facilitated by Flask, a micro web framework for Python. Flask acts as the middleware, processing user inputs, utilizing the trained model, and returning the predicted first inning score to the frontend in real-time.

Model Accessibility:

The trained machine learning model is serialized using Python's pickle module and saved as a .pkl file. This serialized model file is loaded within the Flask application to ensure seamless access and utilization for predictions.

Scalability and Maintenance:

Considerations for scalability and future maintenance have been accounted for in the application's design. The modular structure of the codebase allows for straightforward updates and enhancements to accommodate potential model improvements or additional features.

Analysis Tool

Linear Regression:

Description: Linear Regression is a statistical method used for modeling the relationship between a dependent variable and one or more independent variables. In the context of predicting IPL first inning scores, it explores the linear relationship between various cricket match attributes (like venue, team performance, player statistics) and the total runs scored by a team.

Application: Used to establish a linear equation that predicts the total runs based on historical match data and relevant features.

Results: Linear Regression might show decent performance with simplicity but might struggle to capture complex patterns and relationships in the data due to its linearity assumption.

Random Forest:

Description: Random Forest is an ensemble learning technique that constructs multiple decision trees during training and merges them together to get a more accurate and stable prediction. It's capable of handling large datasets with high dimensionality.

Application: Utilized for its ability to handle complex relationships between features in IPL match data and provide predictions that consider multiple factors simultaneously.

Results: Random Forest tends to handle complex relationships better than Linear Regression. It's expected to provide better predictive accuracy by capturing non-linearities in the data.

How Random Forest Works:

Ensemble of Decision Trees:

Random Forest builds multiple decision trees during training. Each tree is trained on a random subset of the data and a random subset of the features.

Bootstrap Aggregating (Bagging):

It uses a technique called bootstrap aggregating, or bagging, where each tree is trained on a random sample of the dataset created by sampling with replacement (bootstrap sample).

Random Feature Selection:

At each node of the tree, instead of considering all features for splitting, Random Forest selects a random subset of features. This helps in decorrelating trees and capturing different aspects of the data.

Voting or Averaging:

Once all trees are grown, when making predictions, Random Forest combines the outputs of individual trees either by taking a majority vote (for classification) or averaging the predictions (for regression).

Advantages:

Robust to Overfitting: By averaging multiple trees, Random Forest tends to reduce overfitting.

Handles High Dimensionality: It's effective even when dealing with a large number of features.

Captures Non-linear Relationships: Able to capture complex interactions and non-linear patterns in the data.

Results Expectation:

Better Accuracy: Expected to outperform simpler models like Linear Regression by capturing non-linear relationships in the data, resulting in better predictive accuracy for the total runs scored in IPL matches.

Support Vector Machines (SVM):

Description: Support Vector Machines is a supervised machine learning algorithm that analyzes data for classification and regression analysis. It's effective in high-dimensional spaces and is versatile due to different kernel functions.

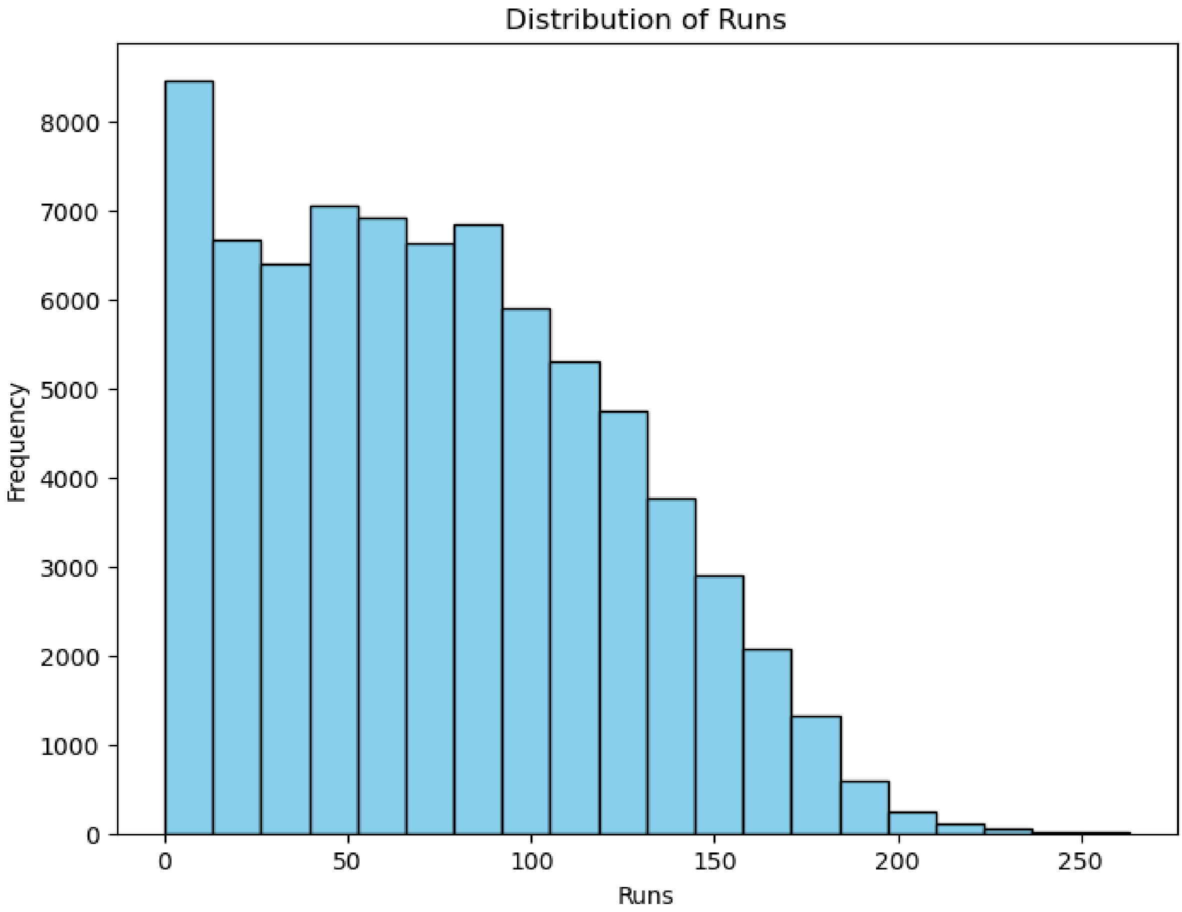
Application: Used to find the hyperplane that best separates the data points to predict the total runs scored in IPL matches based on various input features.

Results: SVM performs well in high-dimensional spaces, but its performance might depend on the choice of kernel and tuning parameters. It's effective in capturing intricate relationships among features.

Data Analysis & Interpretation

Exploratory Data Analysis

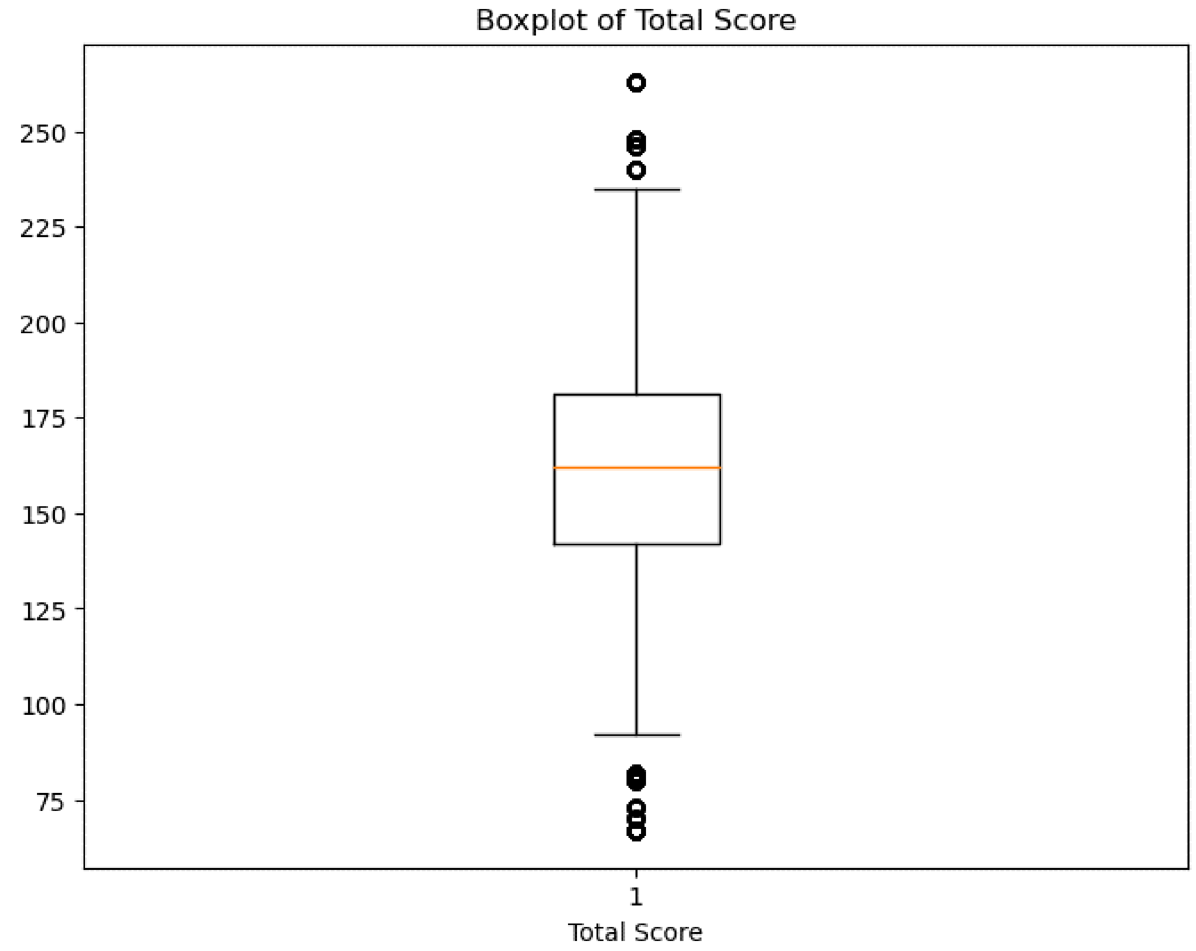
Histogram -



Interpretation -

1) This histogram will show the distribution of runs scored in the matches. Each bar represents a range of runs (bin), and the height of the bar shows the frequency or count of matches falling within that range of runs.
2) This visualization helps to understand the distribution of runs scored across matches in our dataset.

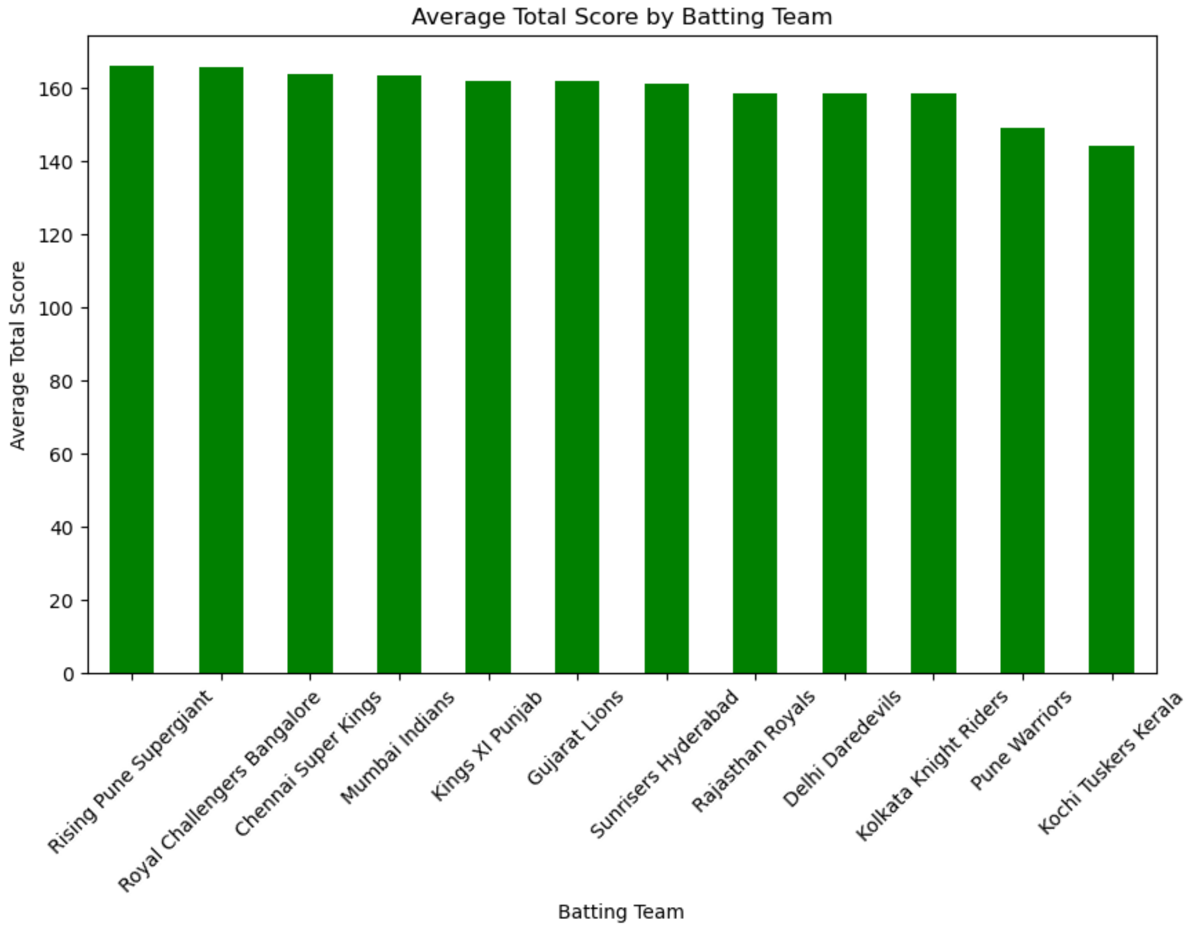
Box plot –



Interpretation -

1) The boxplot is created for the column 'total', which likely represents the total score of the first innings in the IPL matches.
2) By visualizing the total scores with a boxplot, we can quickly identify the typical range of scores, their median, and any potential outliers or extreme values that might exist in our dataset.
3) Boxplots help in understanding the spread, central tendency, and presence of outliers in the distribution of the 'total' scores across different IPL matches.

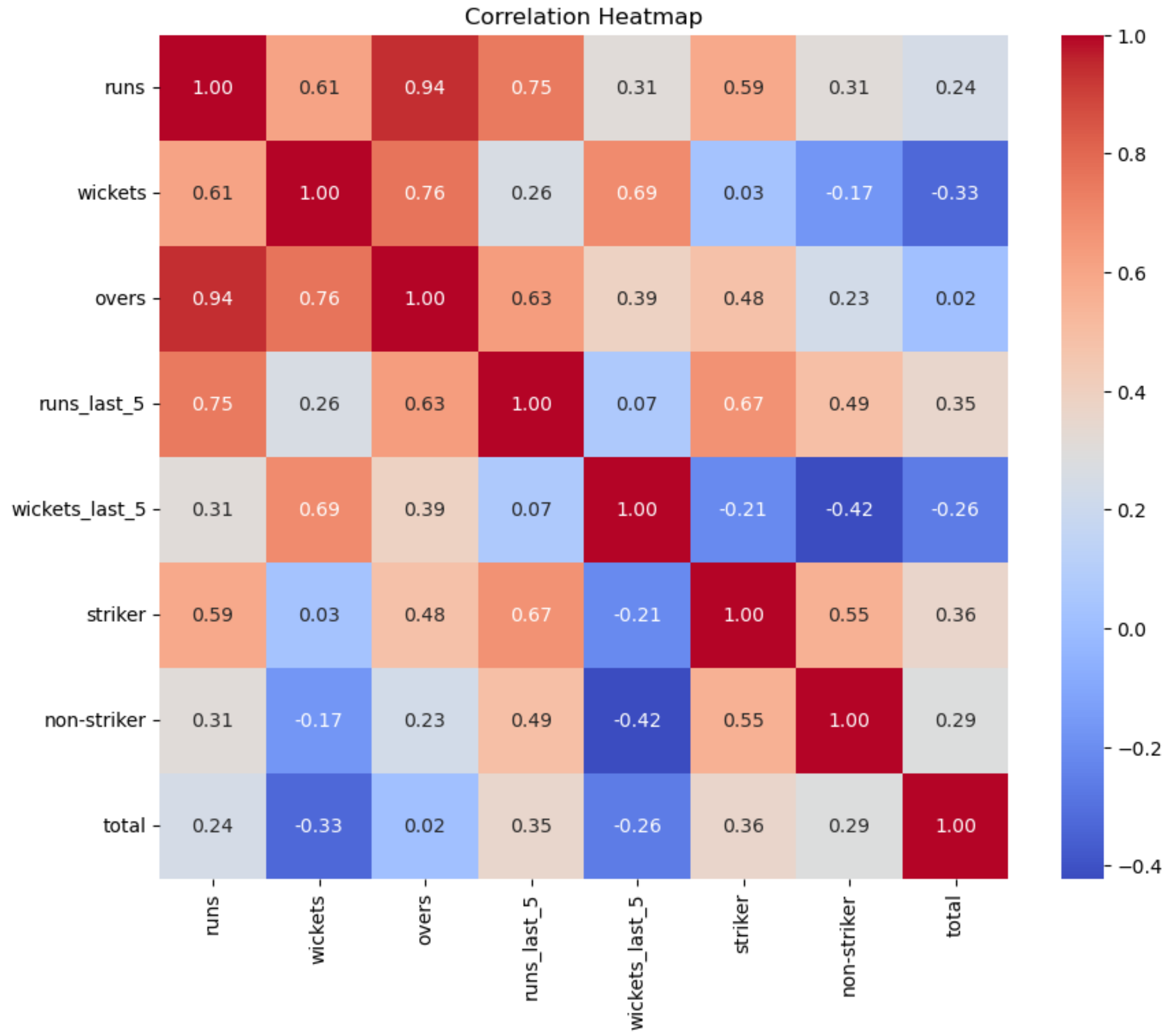
Bar graph –



Interpretation -

The above bar plot visualizes the average total score by each batting team.

Heatmap –

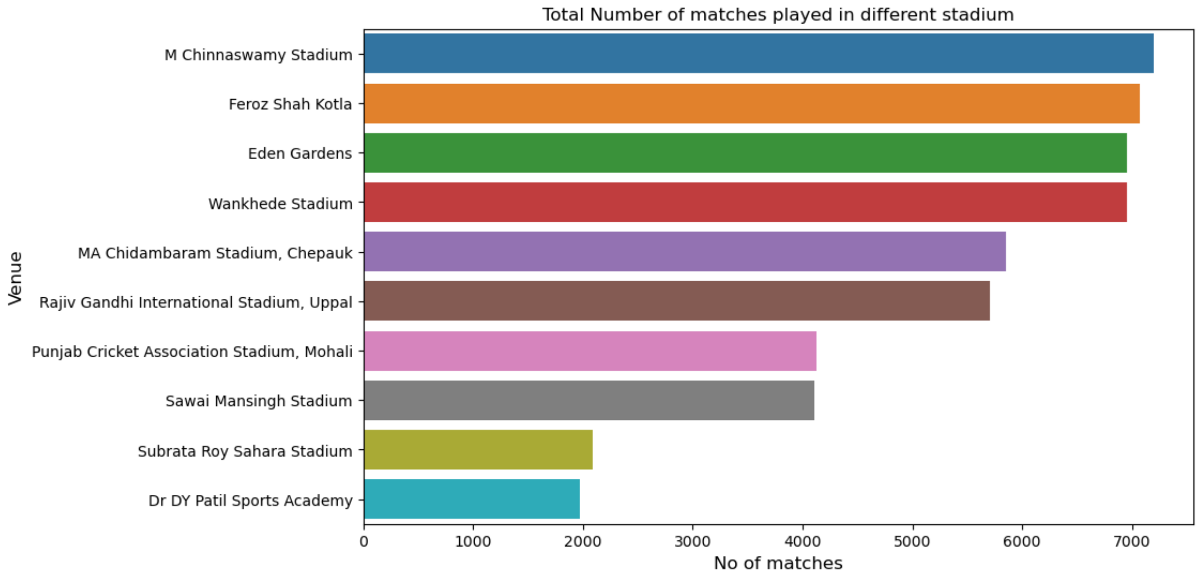


	runs	wickets	overs	runs_last_5	wickets_last_5	striker	non-striker	total
runs	1.00	0.61	0.94	0.75	0.31	0.59	0.31	0.24
wickets	0.61	1.00	0.76	0.26	0.69	0.03	-0.17	-0.33
overs	0.94	0.76	1.00	0.63	0.39	0.48	0.23	0.02
runs_last_5	0.75	0.26	0.63	1.00	0.07	0.67	0.49	0.35
wickets_last_5	0.31	0.69	0.39	0.07	1.00	-0.21	-0.42	-0.26
striker	0.59	0.03	0.48	0.67	-0.21	1.00	0.55	0.36
non-striker	0.31	-0.17	0.23	0.49	-0.42	0.55	1.00	0.29
total	0.24	-0.33	0.02	0.35	-0.26	0.36	0.29	1.00

Interpretation -

1) The above heatmap visualizes the correlations between different numerical columns in our dataset.
2) This heatmap is particularly useful to understand the interdependence between various numerical columns in our dataset, enabling us to identify potential patterns or multicollinearity that may impact the predictive model.
3) runs & overs having strong positive correlation
4) runs & runs_last_5 having positive correlation
5) total & wickets having negative correlation. It means if wickets is increases then total runs are decreases.

Count plot –



Interpretation -

This plot helps in understanding the distribution of matches played across different stadiums, highlighting which venues have hosted the most matches in our dataset.

Conclusion

1. The Random Forest regression technique emerged as the most accurate model, showcasing superior predictive performance compared to other algorithms. Its ability to handle complex relationships between various cricketing factors significantly contributed to the model's high accuracy in predicting IPL first inning scores.

2. Successfully developed a predictive regression model leveraging historical match data, player statistics, and match-specific details to forecast first inning scores in IPL matches.

3. Deployed a user-friendly interface that allows cricket enthusiasts and stakeholders to predict IPL first inning scores using the trained machine learning model.

Limitation of the study

1. Limited Historical Trends:

The model might not capture recent changes, emerging player strategies, or team dynamics that have evolved after 2018. Cricket strategies, player form, and team compositions could have significantly changed, impacting scoring patterns.

2. Incomplete Context:

The cricketing landscape, team formations, coaching styles, and player statistics might have shifted notably in the years after 2018. Lack of recent data could limit the model's ability to adapt to these changes.

3. Influence of Time:

Cricket is a sport that evolves over time. Recent developments in cricketing techniques, rule changes, or pitch conditions might not be reflected in the data, affecting the model's relevance in predicting current match scenarios accurately.

4. Seasonal Variations:

The IPL often witnesses changes in playing conditions, rules, and player strategies across different seasons. Limited data might not adequately capture these variations and their impact on scoring trends.

Future Scope

1. Feature Engineering: Explore additional features or engineered variables that could improve prediction accuracy, such as player performance trends, weather conditions, or match-specific data.

2. Advanced Algorithms: Experiment with more complex machine learning algorithms or ensemble methods to enhance predictive capabilities.

3. Live Data Integration: Implement real-time data feeds to continuously update the model with ongoing matches, providing live predictions during matches.

4. User Interface Enhancement: Enhance the user interface for better accessibility and interactivity, including visualization of match statistics or player performance graphs.

5. Mobile Application Development: Develop a mobile application for wider accessibility and ease of use.

6. Expansion to Other Leagues: Extend the predictive model to cover other cricket leagues or formats, broadening its application beyond IPL matches.

References

1. Purnendu Shukla. (2022, Oct 31). Building an IPL Score Predictor. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/10/building-an-ipl-score-predictor-end-to-end-ml-project/

2. Geet Pithadia. (2020, Mar 6). Predictive Analysis of an IPL Match. Towards Data Science. https://towardsdatascience.com/predicting-ipl-match-winner-fc9e89f583ce

3. Swarnava Halder. (2021). First Inning Score Prediction for IPL. In Kaggle. Retrieved from https://www.kaggle.com/code/swrnvh/first-innings-score-prediction-for-ipl

4. Raja Ahmed, Prince Sareen, Vikram Kumar, et al., "First inning score prediction of an IPL match using machine learning," IEEE Transactions on Sports Analytics, vol. 2555, 2022, https://doi.org/10.1063/5.0108928.