CrickXpert

Manan Chugh 2021335
Kartikey Dhaka 2021534

Abhijeet Anand 2021509
Abhijay Singh 2021226

Abstract

The project employs advanced machine learning techniques to accurately predict outcomes during cricket innings. This encompasses vital metrics such as runs scored, wickets taken, boundaries hit, and even the likelihood of specific events unfolding during a game. By meticulously analyzing historical data and identifying recurring patterns, the system offers invaluable insights. These insights are instrumental for teams and coaches in formulating tactical game strategies and making real-time decisions during matches. Beyond strategy, this application of machine learning also significantly enriches the understanding of cricket for enthusiasts and professionals alike. It underlines the ever-increasing role of data science in sports, transforming traditional cricket viewing into a more informed and immersive experience. Through this initiative, the project demonstrates the potential of merging technology with sports to enhance comprehension and enjoyment of the game.

● Introduction

Merging our love for sports and technology, the allure of predicting cricket scores using machine learning has captivated us. This endeavor enriches our passion for the game, sharpening our skills and introducing an added layer of thrill. Driven by a shared ambition, we embrace this journey, recognizing the transformative potential of data science in sports. This project harnesses machine learning to forecast cricket inning results, including key metrics like runs, wickets, boundaries, and event likelihoods. By revealing patterns, it aims to inform strategies and facilitate real-time analysis, highlighting data science's role in amplifying cricket comprehension and enjoyment. Our goal is to create adept machine learning models that predict crucial metrics in an IPL cricket match. By delving deep into the game's dynamics, we anticipate offering unparalleled insights. This initiative will not only contribute to the growing realm of sports analytics but also underscore the potential of AI in predicting cricket match outcomes. By merging technology with sports, we're setting the stage for a paradigm shift in how fans, team managements, and broadcasters engage with cricket. The culmination of our efforts will be a well-documented, functional system making real-time predictions during live matches, reshaping cricket analytics and augmenting fan engagement

● Literature Survery

1. https://www.tandfonline.com/doi/epdf/10.1080/08839514.2018.1442991?needAccess=true
   The paper on "Machine Learning Applications in Baseball" discusses the utilization of machine learning in baseball analytics, particularly focusing on Regression, Binary, and Multiclass Classifiation problems. It highlights the dominance of Support Vector Machines and k-nearest neighbors in the literature and suggests a potential increase in the use of neural networks for future analyses.

2. https://www.emerald.com/insight/content/doi/10.1016/j.aci.2019.11.006/full/pdf?title=sport-analytics-for-cricket-game-results-using-machine-learning-an-experimental-study
   The paper investigating "Sport analytics for cricket game results using machine learning" explores the use of machine learning techniques to predict IPL cricket

match outcomes. It details the process of feature selection and the application of models like Naïve Bayes and Random Forest, noting the latter's superior performance in accuracy and recall, especially for features based on the home team advantage.

- **Dataset**

1. **Data Description**

The dataset consists of the data of **IPL matches** from 2007 to 2017. The dataset is of 76015x15 size. It consists of the following columns

1. **Mid:** Match identifier number.
2. **Date:** Date of the match.
3. **Venue:** Stadium where the match was played.
4. **Bat_Team:** Batting team's name.
5. **Bowl_Team:** Bowling team's name.
6. **Batsman:** Name of the batsman playing.
7. **Bowler:** Name of the bowler bowling.
8. **Runs:** Runs scored by the batsman.
9. **Wickets:** Number of wickets taken by the bowler.
10. **Overs:** Number of overs bowled.
11. **Runs_Last_5:** Runs scored in the last 5 overs.
12. **Wickets_Last_5:** Wickets taken in the last 5 overs.
13. **Striker:** Runs scored by the batsman currently facing.
14. **Non-Striker:** Runs scored by the batsman not facing.
15. **Total:** Total runs scored by the batting team in the match.
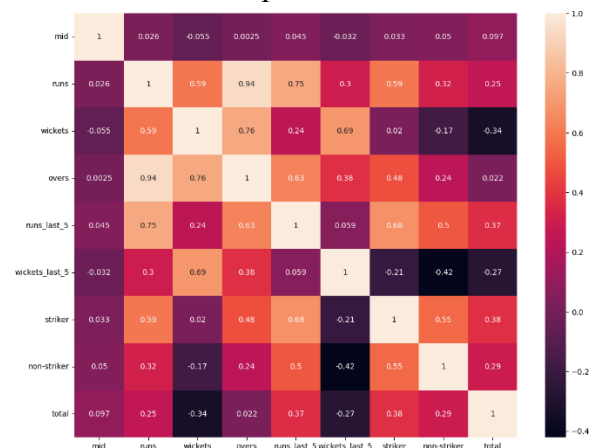
2. **Data Extraction**

We scraped the internet for IPL matches data and we were able to collect the data for all the IPL matches from 2008 - 2017. After extracting this data we were left with 75000+ rows to work upon.
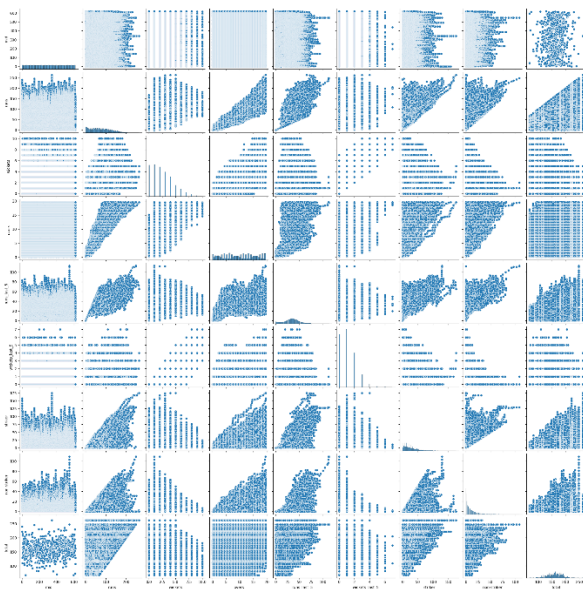
3. **Data Preprocessing**

For optimal results we deleted all the rows which had any missing column in them. Then we thoroughly analyzed the data and came upon a conlusion that only 11 features were important for our model development("mid", "date", "bat_team", "bowl_team", "runs", "wickets", "overs", "runs_last_5", "wickets_last_5", "total", "runs_in_each_ball"). Out of the 14, We have singled out 8 consistent team namely, Kolkata Knight Riders, Chennai Super Kings, Rajasthan Royals, Mumbai Indians, Kings XI Punjab, Royal Challengers Bangalore, Delhi Daredevils, Sunrisers Hyderabad to train our model on hence reducing our dataset size from 76014x11 to 53811x11 improving our accuracy. We took more than or equal to 5 overs because the first 5 overs constitute the powerplay in the IPL. Consequently, performance during this period is inconsistent, with the potential for either a substantial number of runs being scored, a very low score, or the loss of a significant number of wickets. For more accurate results we computed the feature "runs_in_each_ball" and added it to our dataset. For better comparison of the dates we converted the "date" feature from String datatype to Date_Time datatype. We used One Hot Encoding to encode the categorical data ("bat_team", "bowl_team").
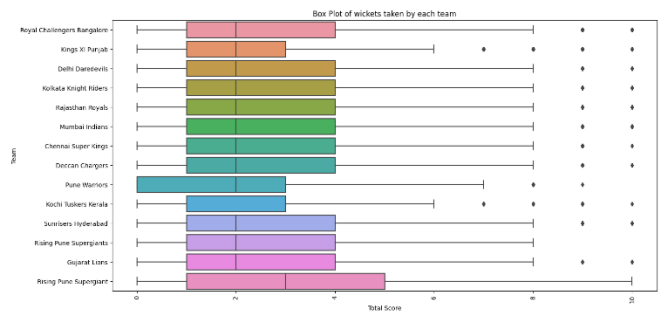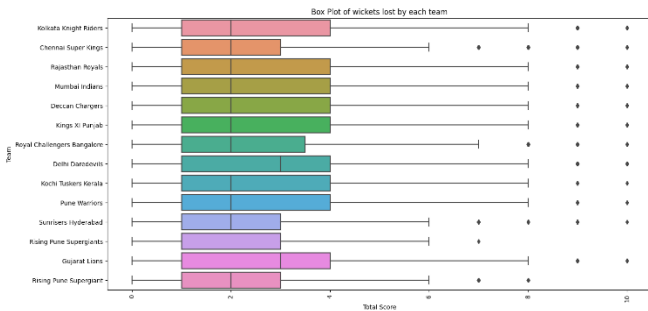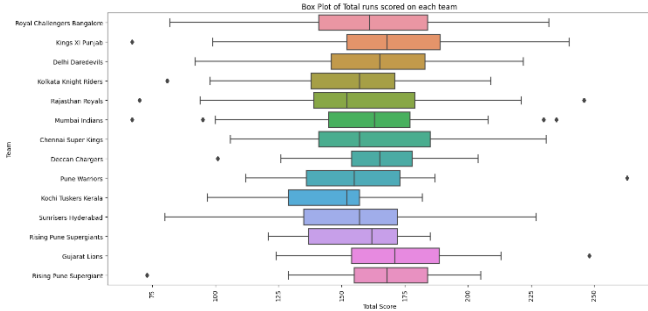
4. **Data Visualization**

Correlation HeatMap:

Pair Plots:

Box Plots:

Box Plot of Total runs scored by each team

Box Plot of Total runs scored on each team

Box Plot of wickets lost by each team

Box Plot of wickets taken by each team

Bar Graphs:

Bar graph of Batting teams and Total runs

Bar graph of Bowling teams and Total runs

Scatter Plot:

Scatter plot of Overs vs Runs in each ball
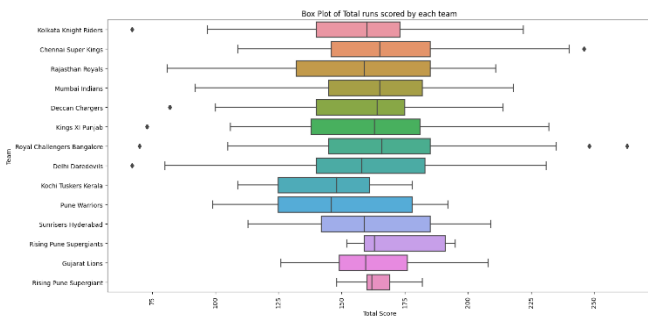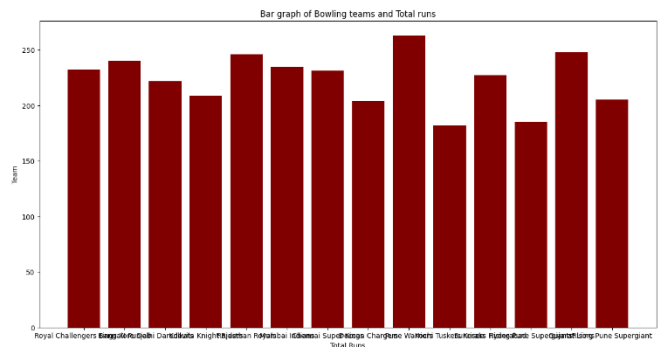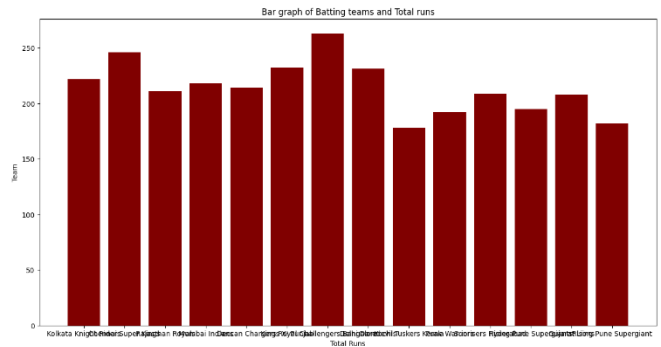
Pie Chart:

Total Runs Scored by Each Batting Team

Legend:
- Chennai Supe
- Chennai Super Kings
- Deccan Chargers
- Delhi Daredevils
- Kings XI Punjab
- Kochi Tuskers Kerala
- Kolkata Knight Riders
- Mumbai Indians
- Pune Warriors
- Rajasthan Royals
- Royal Challengers Bangalore
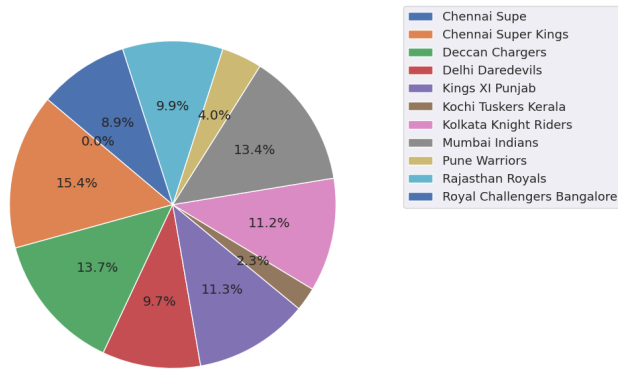
Insights:

1. From the correlation heatmap we inferred the following:
   a) Runs and overs show very high positive correlation.
   b) Wickets and overs also show high positive correlation.
   c) Wickets and total runs are negatively correlated.

2. Runs and overs show linear dependency to some extent.

3. From the box plots we inferred:
   a) The median runs scored by each team is in the range of 150-175.
   b) Matches in which all 10 wickets fall are computed to be outliers.
   c) The median wickets taken in each match are 2-3.

4. There is only one team (Pune Warriors) which had matches with 0 wickets taken by them.

5. Mumbai Indians and Chennai Super Kings scored the highest number of runs per match on average.

6. Rising Pune Super Giants and Kochi Tuskers Kerala were the least-scoring teams per match on average.

7. The maximum number of runs that can be achieved on one ball is 7 assuming a wide ball and 6 runs.

● **Methodology**

We have trained, tested and evaluated two models. The main objective of our first model is to predict the detailed insights of a IPL Matches. Firstly, we will split the encoded dataset according to date, the datapoints before the year 2016 will be used in training the model and the datapoints after 2016 will be used in testing. We are dividing it along time to maintain a timeline continuity in the data. We will try many Regression techniques such as Linear Regression, Decision Tree Regression, Random Forest Regression and AdaBoosting to increase and evaluate the performance of our model. Adaboost is a powerful technique that converts the weak learners(multiple models) to strong learners(final combined model). At the end, we have performed K-Cross Validation by dividing the test and train dataset according to different years. In the 2nd model we compute the runs scored on each ball. We then removed rows with NULL values and selects a subset of columns for further analysis. Categorical data is handled by one-hot encoding. The dataset is scaled using StandardScaler, and used to train three different classifiers: Logistic Regression, Random Forest, and Multilayer Perceptron (MLP).

● **Result and Analysis**

MODEL 1

The evaluation metrics for the Linear Regression model came out as following: -
1. Mean Absolute Error (MAE): 12.15
2. Mean Squared Error (MSE): 254.06
3. Root Mean Squared Error (RMSE): 15.94

The evaluations metrics for the Decision Tree Regression came out as following: -
1. Mean Absolute Error (MAE): 16.92
2. Mean Squared Error (MSE): 486.94
3. Root Mean Squared Error (RMSE): 22.06

The evaluations metrics for the Random Forest Regression came out as following: -

1. Mean Absolute Error (MAE): 13.159
2. Mean Squared Error (MSE): 294.41
3. Root Mean Squared Error (RMSE): 17.16

The evaluations metrics for the Ada Boosting came out as following: -

4. Mean Absolute Error (MAE): 12.06
5. Mean Squared Error (MSE): 245.46
6. Root Mean Squared Error (RMSE): 15.66

After testing all the different models, the Linear Regression Model is giving the best accuracy, so we have continued with the Linear Regression Model.

MODEL 2

Logistic Regression:
Accuracy: 0.477

	Precision	Recall	f1-Score	Support
Accuracy			0.48	15203
Macro Avg	0.16	0.15	0.13	15203
Weighted Avg	0.4	0.48	0.41	15203

Random Forrest Classifier:
Accuracy: 0.390

	Precision	Recall	f1-Score	Support
Accuracy			0.39	15203
Macro Avg	0.14	0.14	0.14	15203
Weighted Avg	0.35	0.39	0.37	15203

wMulti-Level Perceptron:
Accuracy: 0.476

	Precision	Recall	f1-Score	Support
Accuracy			0.48	15203
Macro Avg	0.17	0.16	0.14	15203
Weighted Avg	0.41	0.48	0.42	15203

Considering the provided metrics, the Logistic Regression and MLP models have similar performances, both outperforming the Random Forest classifier in terms of accuracy. The MLP has a slightly better macro average precision and f1-score compared to Logistic Regression, which might make it the preferable model among the three, especially if the balance between classes is a concern. Despite the low absolute accuracy figures, the context of having eight possible outcomes for each ball (12.5% chance if guessing randomly) puts these results in a more positive light. Achieving accuracy figures close to 48% indicates that the models are learning and making predictions that are significantly better than random guessing. In predictive tasks with a high number of classes, such as this one, even small improvements over random chance can be considered a good performance. The models are providing valuable insights and may be further optimized with more data or feature engineering.

• **Conclusion**
1. **Learnings from the Project**
In this project focused on Indian Premier League (IPL) data, we gained hands-on experience in constructing a machine learning pipeline. We realized the crucial role of data collection and utilized the IPL API to gather comprehensive match statistics, player performances, and team records. Utilizing various data visualization tools like matplotlib and seaborn, we analyzed trends and patterns in the IPL dataset. Additionally, we learned

about mutual information classification for effective feature selection. Overall, this project equipped us with essential skills for handling IPL data and building predictive models for cricket analytics.

2. Future Work

We have successfully adhered to the outlined project timeline. We curated the necessary datasets, developed, and tested two models for predicting key metrics in a cricket match, achieving satisfactory results. Looking ahead, our plans for future enhancements include:

a. **Deep Learning:** Explore deep learning architectures, such as recurrent neural networks (RNNs) or Long Short-Term Memory networks (LSTMs), to better model the time-series nature of the data.

b. **User Interface:** Create a user-friendly interface for the model's deployment that could be used by coaches, analysts, or fans to make real-time decisions or analyses.

c. **Hyperparameter Optimization:** Conduct a more extensive hyperparameter search using techniques like grid search or randomized search to fine-tune the models.

d. **Feature Engineering:** Investigate more sophisticated feature engineering techniques. For instance, include features that capture the sequential nature of balls within an over or the match situation like current run rate, required run rate, wickets in hand, etc.

● **Member Contribution**

All four members worked collaboratively on all the tasks in our timeline. The work was distributed equally during this deadline based on one's expertise in a certain topic and their availability at the time.

References

1. https://www.analyticsvidhya.com/blog/2021/09/adaboost-algorithm-a-complete-guide-for-beginners/#:~:text=AdaBoost%20algorithm%2C%20short%20for%20Adaptive,assigned%20to%20incorrectly%20classified%20instances.
2. https://www.geeksforgeeks.org/decision-tree/
3. https://www.javatpoint.com/machine-learning-random-forest-algorithm