Predicting IPL Victory: Machine Learning Techniques for Pre-Match Winner Prediction

SUJAN

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

8th September 2023

Abstract

This project pertains to multiclass prediction, specifically focusing on the Indian Premier League (IPL), a prominent Twenty-20 cricket league. The primary objective is to predict match outcomes based on pre-match data, including team compositions, toss results, player lineups, and venue details. While previous attempts have prioritized achieving higher accuracy, often neglecting potential feature leakage issues, this project places significant emphasis on data integrity and the prevention of feature leakage. The project follows a systematic approach, beginning with the development of an initial baseline model utilizing a limited set of pre-match features. Subsequently, the project involves extensive data cleaning, preprocessing, and the application of various data science techniques to refine the predictive model. Furthermore, the project seeks to identify and rectify limitations in the baseline model by incorporating additional derived features obtained through comprehensive web scraping and enhanced feature engineering. The ultimate goal is to improve prediction accuracy beyond that of the baseline model while ensuring data integrity throughout the process.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

Name: SUJAN Signature: SUJAN

Acknowledgements

I would like to express my heartfelt gratitude to my project supervisor, Nikos Ntarmos, and my project coordinator, Jose Cano Reyes, for their unwavering guidance and support. I'm also thankful to my academic advisors, Kevin Bryson and Una Marie Darragh, for their valuable insights. Special thanks to the Department of Computing at the University of Glasgow for the resources, and to the university for this opportunity.

Contents

Chapter 1 Introduction

1.1 Motivation

The Indian Premier League (IPL) has evolved into a global sports phenomenon, commanding a brand value of ₹90,038 crore (US$11 billion) in 2022 (Darji and Dave, 2013). Beyond its economic prowess, the IPL's impact transcends boundaries, inspiring similar leagues and contributing ₹1,150 crore (US$140 million) to India's GDP (Darji and Dave, 2013). The league's value reached a remarkable US$10.9 billion in 2022, showcasing its unparalleled growth (Darji and Dave, 2013).

The motivation behind our project is clear: as the IPL's value soars and its influence increases rapidly in number, informed betting decisions become paramount. The immense scale of annual online betting on cricket, valued at around INR 3-4 trillion ($40-$50 billion) according to a report by the Federation of Indian Chambers of Commerce and Industry (FICCI) in 2019, underscores the critical need for accurate and data-driven insights (Editor, 2023) . Therefore, it's crucial to create a project that tackles the task of turning a typically guessing-based activity into a data-driven and strategic one. Through the development of predictive models for pre-match predictions, this project seeks to change the way cricket fans and bettors interact with the sport.

1.2 Aim and Objectives

Aim: The ultimate aim is to identify any limitations in the existing approaches, and if such flaws exist, to invent solutions and create an optimized predictive machine learning technique. This effort aims to provide invaluable assistance to cricket enthusiasts and bettors, enhancing their ability to make informed decisions with the best possible predictive accuracy.

Objectives:

1.Evaluating and Enhancing Existing Approaches: The primary objective of this project is to thoroughly research existing approaches, conduct a critical analysis, experiment with their methodologies, identify any flaws with supporting evidence, and subsequently propose effective solutions.

2.Historical Data Management: In cricket, players can retire or new ones emerge, teams and venues change names, and new players join. Our aim is to manage these shifts in historical data, so we have accurate information for analysis and prediction.

3. Strategic Feature Selection: Explore cricket fundamentals to extract correlated features that provide meaningful insights into match outcomes, ensuring a solid foundation for the predictive techniques.

1.3 Report Structure

Chapter 2: Background and Previous Approaches

In this chapter, we explore cricket's domain knowledge, delve into previous approaches, assess their strengths and weaknesses, and critically analyze their effectiveness.

Chapter 3: Proposed Approaches and Comparison

Here, we present our innovative methods, evaluate them, and compare their performance.

Chapter 4: Conclusion and Future Work

The final chapter concludes our findings, discusses results, and outlines future research directions to enhance the predictive techniques discussed.

Chapter 2 Survey

2.1 Background Knowledge of cricket and importance of highly correlated features.

Cricket is a game where two teams, each comprising eleven players, participate in a bat-and-ball sport (ESPNcricinfo, n.d.). The cricket ground is a circular area with a rectangular pitch in the middle where the game is played. On the opposing side, the objective is to dismiss the batters and restrict their score, while the game's goal is for one team to accumulate runs by hitting the ball and dashing between two sets of wooden stumps located at either end of the pitch (ESPNcricinfo, n.d.). At the conclusion of the game, the team with the most runs wins.

Key Aspects of Cricket and Their Importance:

A. **Team1 and Team2:** The two opposing teams in a match. Identifying the strengths and weaknesses of each team contributes to predicting their performance and potential outcome(ESPNcricinfo, n.d.).
B. **Toss Winner and Decision:** The team winning the coin toss decides whether to bat or field first (ESPNcricinfo, n.d.). The decision affects the match strategy and the team's chances of winning.
C. **Toss Decision (Toss_Decision):** The choice made by the toss-winning team, either to bat or field(ESPNcricinfo, n.d.). This decision can impact the match's dynamics, depending on pitch conditions and team strengths.
D. **DL Applied:** Duckworth-Lewis method applied in rain-affected matches(ESPNcricinfo, n.d.). Reflects the impact of weather interruptions on the match's course.
E. **Venue:** The location where the match is held. Venue conditions, such as pitch type and dimensions, influence team strategies and outcomes.
F. **Current Run Rate:** The average number of runs scored per over. Indicates a team's momentum and scoring rate.
G. **Required Run Rate**: The rate at which the batting team needs to score to win. It gauges the challenge faced by the batting side.
H. **Weather:** Environmental conditions like rain, humidity, and visibility can influence player performance, especially bowlers.
I. **Pitch:** The playing surface where the ball interacts with the bat and the bowler's skill. The pitch's behavior can impact batting and bowling strategies.
J. **Runs Left and Balls Left:** Reflects the batting team's progress and how many runs they need to win, considering the remaining balls.
K. **Wickets:** The number of batsmen dismissed. Affects the team's remaining batting resources and strategy.
L. **Total Runs:** Cumulative runs scored by a team. A crucial metric for determining match status and potential outcomes.
M. **Batsman Strike Rate:** The rate at which a batsman scores runs(ESPNcricinfo, n.d.). Indicates a player's aggressiveness and ability to maintain scoring momentum.

N. **Bowler's Economy:** The average number of runs conceded by a bowler per over (ESPNcricinfo, n.d.). Reflects a bowler's effectiveness in limiting runs.

Understanding these features allows us to analyze the cricket matches, helping to predict outcomes and inform strategic decisions.

2.2 Previous related works

The literature survey serves as a great start for understanding the existing landscape of IPL match outcome prediction using Machine Learning techniques. This section dives into prior studies, highlighting their methodologies, findings, and identified strengths and limitations. This comprehensive review forms the basis for our project's innovation and aims to build upon existing knowledge to enhance accuracy, feature engineering, and robustness in predicting IPL match outcomes.

2.2.1 Prediction of IPL Match Outcome Using Machine Learning Techniques by Srikantaiah et al. (2021).

In their research, Srikantaiah et al. (2021) aimed to predict the outcomes of Indian Premier League (IPL) cricket matches using machine learning techniques. They employed algorithms such as Support Vector Machines (SVM), Random Forest Classifier (RFC), Logistic Regression, and K-Nearest Neighbor to achieve accurate match outcome predictions.

Strengths:

i) Wide Range of Data: The researchers utilized an extensive dataset spanning nine years of IPL matches, covering team performance, player statistics, team wise home and away dataset and match details (Srikantaiah et al., 2021).

ii) Simple Model Architecture: Their straightforward model architecture is easily comprehensible, even for individuals new to machine learning. This simplicity facilitates broader understanding and adoption of their approach.

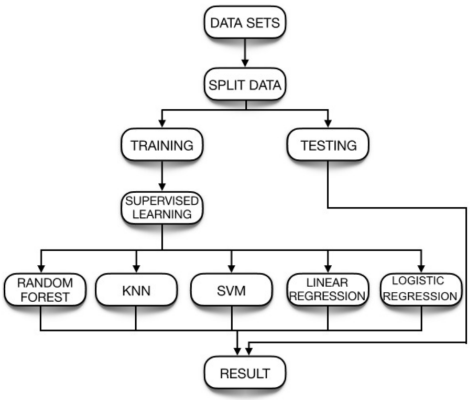


Fig 2.1 Architecture of the model used (Srikantaiah et al., 2021)

Weakness:

The researchers achieved an impressive accuracy of 88% in their predictions; however, this extraordinary accomplishment was flawed by significant mistakes.

i) Temporal Data Leakage : They included features in their model that are not suitable for match outcome prediction. Specifically, their use of post-match details like "win_by_runs," "win_by_wickets," and "player_of_the_match" which is shown in the Fig 2.2 undermines the predictive integrity of their approach (Srikantaiah et al., 2021). Because these features are only accessible after the match finishes, using them goes against the idea of predicting outcomes before the match starts, making the accuracy of such predictions less dependable. It's surprising that they intentionally allowed data leakage to achieve accuracy. This error is elaborated upon in the experimentation section.

Win by runs	The number of runs by which the team won.
Win by wickets	The number of wickets by which the team won.
Player of match	Name of player who was awarded player of the match.

Fig 2.2 Temporal Data Leakage by (Srikantaiah et al., 2021)

ii) Lack of feature Engineering: Another noticeable error in their approach becomes clear when examining the architectural models in Fig 2.1 : the absence of feature engineering. Despite the crucial role it plays in model training, they overlooked this essential step.

2.2.2 Analysis and Prediction for the Indian Premier League by Barot et al., 2020.

Conducted by H. Barot, A. Kothari, P. Bide, B. Ahir, and R. Kankaria, the project "Analysis and Prediction for the Indian Premier League" stands out for its robust methodology and insightful findings. Two datasets were employed: the first providing detailed ball-to-ball information, whereas the second about match summaries including teams, winners, and toss outcomes (Barot et al., 2020). Key factors affecting cricket matches were identified, encompassing the toss, weather, pitch conditions, batting/bowling strength, form, and more (Barot et al., 2020). The dataset was pre-processed, focusing on the last five years (2015-2019) and comprising 298 matches (Barot et al., 2020).

Strengths:

i) Enhanced Feature Engineering: The project showcases robust feature engineering techniques, incorporating factors like team form, batting index, and bowling index. By leveraging these indices along with other relevant features, the model gains a more comprehensive understanding of player and team performance.

$$\text{BattingIndex} = ((\text{BattingAverage}) * (\text{BattingSR}))/100$$ (Barot et al., 2020).

BowlingIndex =((BowlingAverage)*(BowlingSR)) /100 (Barot et al., 2020).

ii) In-depth Exploratory Data Analysis (EDA): The project performs a comprehensive EDA, exploring various factors that impact cricket match outcomes. Noteworthy insights from the analysis were as follows:

1. Toss Factor: About 55% of teams that won the toss went on to win the match, highlighting the importance of the toss victory (Barot et al., 2020).
2. Batting vs. Chasing: In IPL, teams pursuing targets won 57% of the matches, emphasizing the inclination toward chasing as a strategy (Barot et al., 2020).
3. Targets Chased: Chasing targets of 200 or more runs occurred in only about 15.6% of instances (Barot et al., 2020).
4. Runs Scored per Over: Teams often scored more runs between the 1st and 6th overs (Powerplay) and at the end of the innings (Barot et al., 2020).
5. Batting and Bowling Index: Batting and bowling indices were introduced, combining performance metrics like average and strike rate for evaluating players (Barot et al., 2020).
6. Form: Form was calculated for teams based on their recent five matches (Barot et al., 2020).

Weakness:

i) Future Data Leakage: One notable weakness in the project by H. Barot, A. Kothari, P. Bide, B. Ahir, and R. Kankaria is their use of all-time averages for features such as batting index and bowling index. This approach accidentally includes future performance data, resulting in data leakage and compromising the model's training process by introducing information from beyond the historical context.

ii) Limited Temporal Scope: The study's utilization of a dataset covering only the years 2015 to 2019, despite the availability of data spanning from 2008 to 2019, neglects the ever-changing nature of IPL cricket. This oversight results in an insufficient representation of historical transformations, as player compositions and strategies evolve over time.

iii) Inadequate Historical Span: The dataset's narrow time frame fails to capture the broader historical context of IPL's evolution, resulting in a skewed understanding of player performance and team dynamics across various seasons. This is a form of "selection bias," as it doesn't consider the full spectrum of changes that occurred over the years.

2.2.3 Predicting Results of Indian Premier League T-20 Matches using Machine Learning by Agrawal, Singh and Sharma, 2018.

The research carried out by Agrawal, Singh, and Sharma (2018), which focused on using Machine Learning techniques to predict outcomes in Indian Premier League (IPL) T-20 matches, has established a basis for comparable explorations in this domain. In their approach, key features such as Average Run Rate, Average Strike Rate, and Power Play Strike Rate were employed to make predictions (Agrawal, Singh and Sharma, 2018). By employing Support Vector Machine (SVM), Naïve Bayes, and CTree classifiers, the study aims to predict

match results based on mainly two datasets i.e Matches and Deliveries dataset(Agrawal, Singh and Sharma, 2018).

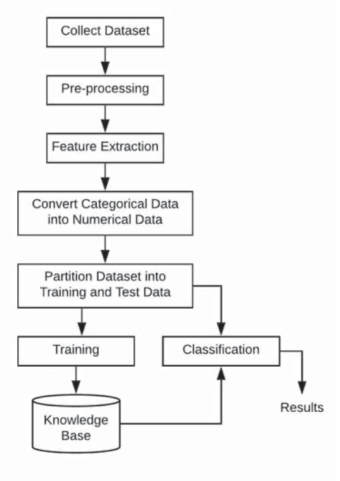


Fig 2.3 System Architecture (Agrawal, Singh and Sharma, 2018)

Strengths and Weakness:

They integrated crucial factors like Average Strike Rate, Average Run Rate, and Power Play Strike Rate into their study, factors of substantial relevance in match prediction (Agrawal, Singh and Sharma, 2018). These factors serve as strengths in their approach, contributing to accurate match outcome forecasts, especially during ongoing matches.However, the paper overlooks a crucial aspect – the consideration of Average Strike Rate, Average Run Rate, and Power Play Strike Rate, which necessitates waiting for a certain match phase to conclude. This limitation could make the approach less effective, as these metrics only become relevant once a specific range of the match has completed.

2.2.4 A DEA Model for Selection of Indian Cricket Team Players by Chaudhary, Bhardwaj and Sakshi Lakra, 2019 .

The study by Chaudhary et al. (2019) introduces an innovative approach known as Data Envelopment Analysis (DEA) to assess player efficiency, facilitating the selection of a potent Indian cricket team. DEA, first introduced by Charnes & Cooper in 1978, calculates the efficiency of Decision Making Units (DMUs) where the weighted outputs to inputs ratio is maximized, helping in the formation of a successful team (Chaudhary, Bhardwaj and Sakshi Lakra, 2019).

Distinguishing Our Approach from Chaudhary et al.'s (2019) Method :

Our Approach	Chaudhary et al.'s (2019) Method
• Weighted Feature Summation(www.sciencedirect.com, n.d.).	• Data Envelopment Analysis (DEA) (Chaudhary, Bhardwaj and Sakshi Lakra, 2019).
• Involves feature selection	• Aims to assess the efficiency of

<table>
<tr>
<td>based on correlation analysis, assigning weights to features, computing a weighted sum, and normalizing to predict outcomes.</td>
<td>Decision Making Units (DMUs) by maximizing the ratio of weighted outputs to weighted inputs, considering other DMUs' efficiency scores (Chaudhary, Bhardwaj and Sakshi Lakra, 2019).</td>
</tr>
<tr>
<td>• Focuses on individual player performances within specific seasons to assess their impact on outcome prediction.</td>
<td>• Aggregates player performance across all innings using linear constraints (Chaudhary, Bhardwaj and Sakshi Lakra, 2019).</td>
</tr>
<tr>
<td>• (F1*w1+F2*w2+.....+Fn*wn) / (w1+w2+.....+wn) where F1,F2…Fn are the Features w1,w2….wn are the weights (www.sciencedirect.com, n.d.).</td>
<td>• 3933λvijay + 6331λkohli + ... + 1025λkartik + 56λvihari >= 3933β (Chaudhary, Bhardwaj and Sakshi Lakra, 2019).</td>
</tr>
<tr>
<td>Final Result:
<table><tr><th>Player</th><th>Year</th><th>Scaled_Score</th></tr><tr><td>Shaun Marsh</td><td>2008</td><td>0.6755872862662300</td></tr><tr><td>Gautam Gambhir</td><td>2008</td><td>0.5867247824541020</td></tr></table></td>
<td>Final Result:
<table><tr><th>Players</th><th>Runs</th><th>Ave</th><th>SR</th><th>100s</th><th>50s</th><th>4s</th><th>6s</th><th>DEA Score</th></tr><tr><td>M Vijay</td><td>3933</td><td>39.33</td><td>46.56</td><td>12</td><td>15</td><td>463</td><td>33</td><td>1</td></tr><tr><td>V Kohli</td><td>6331</td><td>54.57</td><td>58.26</td><td>24</td><td>19</td><td>700</td><td>18</td><td>1</td></tr></table>Fig. Final DEA score for each player (Chaudhary, Bhardwaj and Sakshi Lakra, 2019) .</td>
</tr>
</table>

2.3 Experimenting Past Works

In cricket or any similar games, claiming to predict the winner with 90% or any other high percentage is highly questionable. If this were the case, everyone would be using such a system to predict winners and make significant profits. The accuracy claims raised doubts in our mind, prompting us to conduct an experiment on their methodology. The findings are detailed below.

2.3.1 Experiment 1: Elevated Accuracy due to Future Leakage

In our first experiment, we examined various pre-match cricket prediction methods used in research and online sources, such as javapoint. Surprisingly, we discovered that many of these approaches suffered from future data leakage. For instance, when "win_by_runs" was not zero and "win_by_wickets" was zero, it implied that the team batting first had won. Conversely, when "win_by_runs" was zero and "win_by_wickets" was not zero, it indicated that the team batting second had won. This kind of information indirectly revealed which team was likely to win, constituting data leakage, specifically future leakage. Additionally, these features were only available after the match, rendering them unusable for predicting upcoming matches.

With Future Leakage	Without Future Leakage

Metric	Value
Accuracy	0.95302
F1 Score	0.95302
Precision	0.95302
Recall	0.95302

Fig 2.4.1 Evaluation Metrics

Metric	Value
Accuracy	0.469799
F1 Score	0.469799
Precision	0.469799
Recall	0.469799

Fig 2.4.2 Evaluation Metrics

The training score curve shows a steep initial increase, followed by a relatively straight plateau, while the validation curve exhibits a steep and consistent increase throughout clearly showing Overfitting situation.

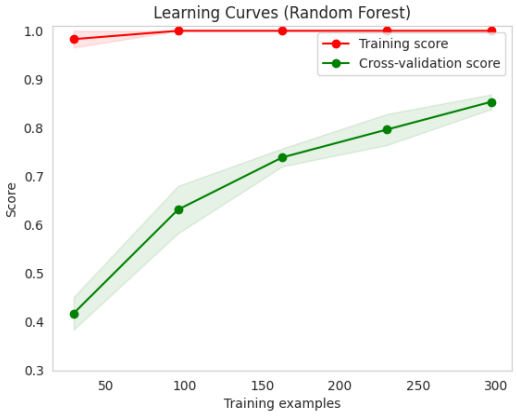


Fig 2.5.1 Learning curve clearly showing Overfitting

The curves exhibit divergence, with the training score curve decreasing initially and the validation curve increasing at the outset but decreasing towards the end showing clear Underfitting situation.

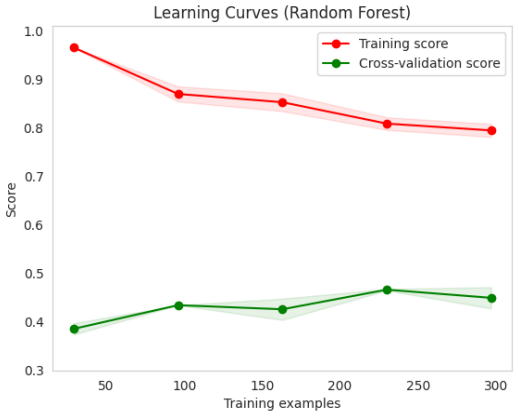


Fig 2.5.2 Learning curve clearly showing underfitting

2.4 Overall Critical Analysis

After referring to the research papers and conducting many experiments on the previously conducted works, we criticized the loopholes presents in the past works and to be worked on these problem statements.These are mentioned below

1. Predicting pre-match winner without Data leakage / Future leakage.
2. Managing the historical data such as player performance and their retirements, change in the team names.
3. Maximizing accuracy and F1 score by extraordinary feature engineering while avoiding model overfitting.

Chapter 3 Design And Implementation

3.1 Approach I (Base Model)

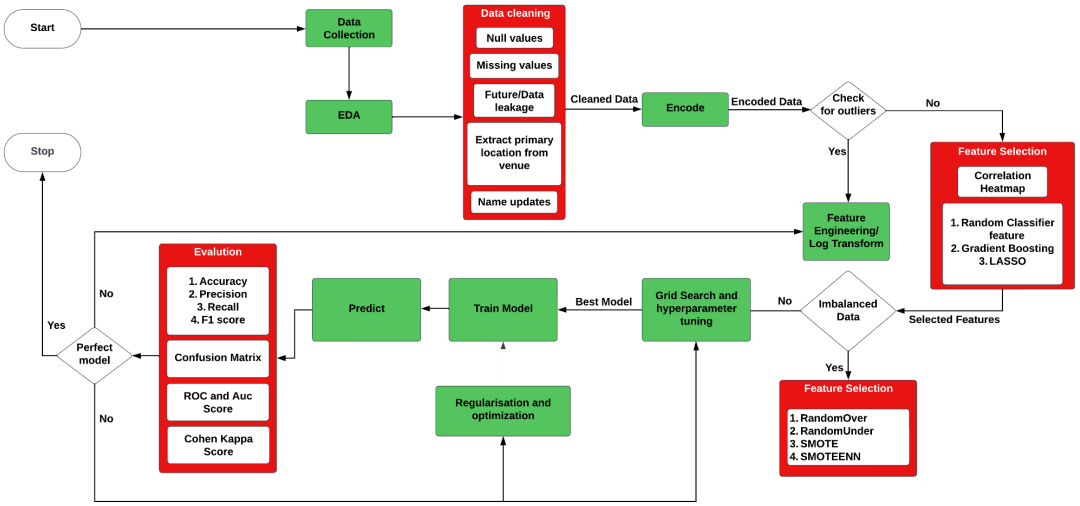


Fig 3.1 Base model Pipeline

3.1.1 Data Collection

The dataset utilized for this study includes IPL matches spanning from 2008 to 2022. This comprehensive dataset includes diverse information such as match dates, participating teams, toss results, match outcomes, winners, and Venues where the match was held. The dataset was obtained from Kaggle. The data allows for an exploration of historical team performances, toss dynamics, and match results, all of which are essential for the models' construction and evaluation.

The dataset comprises 950 rows, reflecting the total number of matches completed during this period. The relatively small dataset size is a result of the limited number of matches played. This presents a significant challenge in training models, as having a smaller dataset can be more demanding in achieving accurate predictions. While working with larger datasets is comparatively easier, handling a smaller dataset like this poses a distinct challenge that this research aims to address.

3.1.2 Exploratory Data Analysis

The Exploratory Data Analysis (EDA) process offered valuable insights into the dataset's characteristics (Sharma, 2022). Notably, certain attributes exhibit high cardinality, such as "Date," "MatchNumber," "Player_of_Match," "Umpire1," and "Umpire2," each with a substantial number of distinct values. Moreover, relationships between variables were identified; for instance, "ID" showed strong correlation with "Season" and another parameter, while "Margin" demonstrated a significant correlation with "SuperOver " and another field. Instances of missing data were detected, particularly in "City" (5.4% missing) and "Margin" (1.9%

missing), with the "method" attribute indicating a noteworthy absence of values (98.0% missing). An imbalance was evident in the "SuperOver" attribute, with one category accounting for 88.9% of the instances. These findings emphasize the need to address missing data, manage imbalanced features, and consider correlations when advancing to subsequent modeling stages.

3.1.3 Data Preprocessing

3.1.3.1 Data Cleaning

1. The feature with high missing values such as “method” should be dropped completely.
2. The feature which has a future leakage such as "SuperOver", "WonBy", "Margin", "Player_of_Match" (“win_by_runs” and “win_by_wickets” if present in other dataset) should also be dropped.
3. The 'Venue' column was modified to extract the primary location information by splitting values at commas and selecting the first part.
4. We are excluding the 'Date', "ID," and "MatchNumber" columns from the analysis since they are not relevant for predicting the match winner.
5. Replacing the team names which has changed overtime,eg
   'Delhi Daredevils' changed their names to 'Delhi Capitals’ and 'Kings XI Punjab' changed to 'Punjab Kings'.
6. Encoding the Categorical variables using LabelEncoder().

	City	Season	Team1	Team2	Venue	TossWinner	TossDecision	WinningTeam	Umpire1	Umpire2
0	Ahmedabad	2022	Rajasthan Royals	Gujarat Titans	Narendra Modi Stadium	Rajasthan Royals	bat	Gujarat Titans	CB Gaffaney	Nitin Menon
1	Ahmedabad	2022	Royal Challengers Bangalore	Rajasthan Royals	Narendra Modi Stadium	Rajasthan Royals	field	Rajasthan Royals	CB Gaffaney	Nitin Menon
2	Kolkata	2022	Royal Challengers Bangalore	Lucknow Super Giants	Eden Gardens	Lucknow Super Giants	field	Royal Challengers Bangalore	J Madanagopal	MA Gough
3	Kolkata	2022	Rajasthan Royals	Gujarat Titans	Eden Gardens	Gujarat Titans	field	Gujarat Titans	BNJ Oxenford	VK Sharma
4	Mumbai	2022	Sunrisers Hyderabad	Punjab Kings	Wankhede Stadium	Sunrisers Hyderabad	bat	Punjab Kings	AK Chaudhary	NA Patwardhan
...	...	...	...	...	...	...	...	...	...	...
945	Kolkata	2007/08	Kolkata Knight Riders	Deccan Chargers	Eden Gardens	Deccan Chargers	bat	Kolkata Knight Riders	BF Bowden	K Hariharan
946	Mumbai	2007/08	Mumbai Indians	Royal Challengers Bangalore	Wankhede Stadium	Mumbai Indians	bat	Royal Challengers Bangalore	SJ Davis	DJ Harper
947	Delhi	2007/08	Delhi Daredevils	Rajasthan Royals	Feroz Shah Kotla	Rajasthan Royals	bat	Delhi Daredevils	Aleem Dar	GA Pratapkumar
948	Chandigarh	2007/08	Kings XI Punjab	Chennai Super Kings	Punjab Cricket Association Stadium	Chennai Super Kings	bat	Chennai Super Kings	MR Benson	SL Shastri
949	Bangalore	2007/08	Royal Challengers Bangalore	Kolkata Knight Riders	M Chinnaswamy Stadium	Royal Challengers Bangalore	field	Kolkata Knight Riders	Asad Rauf	RE Koertzen

950 rows x 10 columns

Fig 3.2 Final dataframe after data cleaning and preprocessing

3.1.3.2 Detecting Outliers using The Interquartile range (IQR) Technique.

In the data preprocessing phase, quartile-based outlier detection was conducted to identify potential anomalies within the dataset. This involved calculating the 1st quartile (Q1) and 3rd quartile (Q3) for the selected columns using the `quantile()` function with quantile values of 0.25 and 0.75, respectively (Vinutha, Poornima and Sagar, 2018). These quartiles signify the data values below which 25% and 75% of the data points lie. Subsequently, the interquartile range (IQR) was computed as the difference between Q3 and Q1. Outliers were considered as data points that fell below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR (Vinutha, Poornima and Sagar, 2018). The process aimed to highlight potential outliers that could skew results or impact model performance. It is noteworthy that no outliers were identified in the current dataset based on these criteria.

```
Q1 = data[columns_to_check].quantile(0.25)
Q3 = data[columns_to_check].quantile(0.75)
IQR = Q3 - Q1
outlier_threshold = 1.5
outliers = ((data[columns_to_check] < (Q1 - outlier_threshold * IQR)) |
            (data[columns_to_check] > (Q3 + outlier_threshold * IQR)))
outlier_rows = data[outliers.any(axis=1)]
print("Rows with outliers:")
print(outlier_rows)
```

```
Rows with outliers:
Empty DataFrame
Columns: [City, Season, Team1, Team2, Venue, TossWinner, TossDecision, WinningTeam, Umpire1, Umpire2]
Index: []
```

Fig 3.3 Output showing no outliers present in the dataset

3.1.3.3 Feature selection using Correlation HeatMap

It's essential to focus only on features that are available before the match begins. Some features might show a high correlation with the "WinningTeam" column, but they could be derived after the match concludes. Using such features wouldn't make sense for prediction purposes. After examining the values, we have learned more about how the different factors connect to the "WinningTeam." Some things stood out. The "team1," "team2," and "toss decision" have a noticeable link to the match result. On the other hand, features like "toss winner," "city," "season," "umpire1," "umpire2," and "venue" don't seem to impact the outcome as much. This insight tells us that factors like the teams playing and the choice made after the toss can make a difference in predicting the winner. But things like who won the toss, the city, season, umpires, and venue might not have a strong effect.

Final Features selected after analyzing correlation heatmap is 0,1,2,3,5,6 and 8 columns in dataframe i.e. “Team1”, “Team2”, “TossDecision”, “TossWinner”, “Season”, “Umpire1” and Venue ".

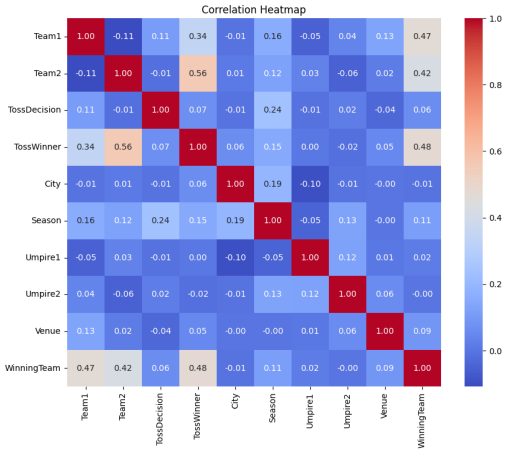


Fig 3 .4 Correlation HeatMap showing the correlation between different features

3.1.3.4 Splitting the dataset into Train,Validation and Test

The data was divided into three sets: training, validation, and test, using an 80/10/10 split. To mitigate the constraints of limited data, a larger portion was allocated for training the model.

3.1.3.5 Feature selection Using Random Forest, Gradient Boosting and LASSO

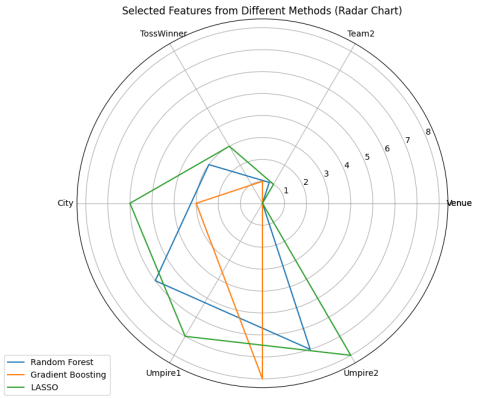


Fig 3.5 Spider graph showing the selected features for each models

1. Random Forest Features: The Random Forest method assigns high importance to features indexed at 0, 1, 3, 6, and 7 in the dataframe. Therefore the features selected are 'Team1" , "Team2", "TossWinner", "Umpire1" and "Umpire2".
2. Gradient Boosting Features: Gradient Boosting, on the other hand, focuses on features at indices 0, 1, 3 and 8 in the data frame i.e . 'Team1" , "Team2", "TossWinner" and "Venue".
3. LASSO Features: LASSO, in its feature selection process, covers a broader range of attributes at indices 0, 1, 3, 6 and 8 in the dataframe ie 'Team1" , "Team2", "TossWinner","Umpire1", "Umpire2" and " Venue".

3.1.3.6 Handling Imbalanced Data Using RandomOver, RandomUnder, SMOTE and SMOTEENN.

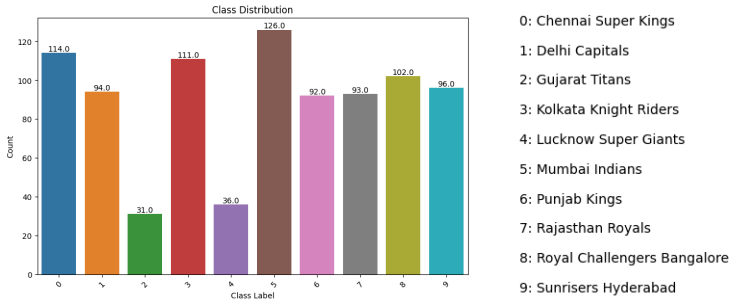


Fig 3.6 CountPlot showing high class Imbalance in dataset

The dataset exhibits a slight class imbalance, with certain team labels such as "Gujarat Titans" and "Lucknow SuperGiants" having comparatively fewer occurrences. This could potentially impact the model's ability to accurately predict these underrepresented classes.

To address the issue of class imbalance in our dataset, we can explore various techniques such as RandomOver, RandomUnder, SMOTE, and SMOTEEN (Kumar, Lalotra and Kumar, 2022). By comparing the performance of these

methods, we can determine which approach is more effective in improving the model's ability to handle the imbalanced class distribution.

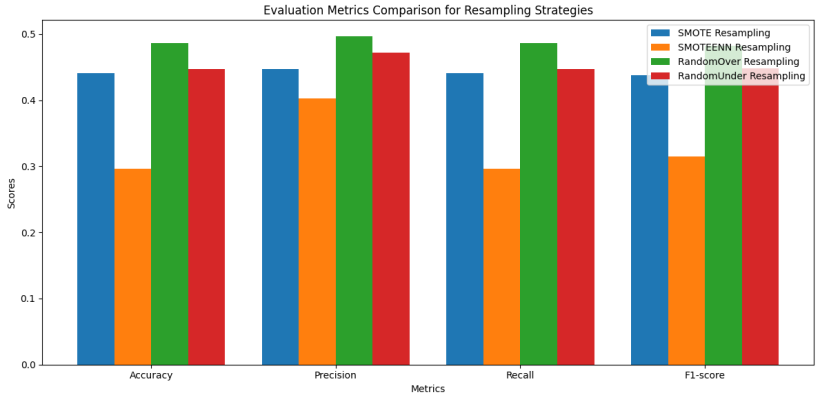


Fig 3.7 grouped bar graph showing comparision between the methods

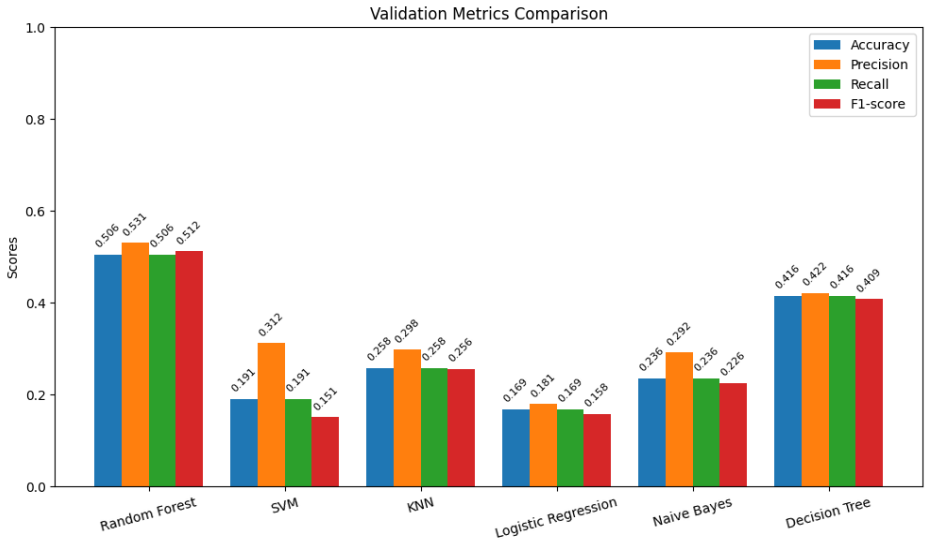
Based on the information displayed in the graph, it's clear that the RandomOver sampling technique outperforms the others. As a result, we should opt for RandomOver sampling to effectively handle the class imbalance present in our dataset.

3.1.4 Model Selection with Resampling and Hyperparameter Tuning

We employed the RandomOverSampler to address class imbalance, ensuring a balanced dataset for robust model training. Six machine learning models, including Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression, Naive Bayes, and Decision Tree, were evaluated with hyperparameter tuning through GridSearchCV. Model performance was assessed using accuracy, precision, recall, and the F1-score. The model with the highest accuracy, precision, recall, and F1-score on the validation dataset was selected as the best model and further tested on an independent dataset.

3.1.5 Evaluation of Model

3.1.5.1 Accuracy, Precision, Recall and F1 Score.



Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.505618	0.530725	0.505618	0.511996
SVM	0.191011	0.31236	0.191011	0.151037
KNN	0.258427	0.298335	0.258427	0.256172
Logistic Regression	0.168539	0.180573	0.168539	0.158498
Naive Bayes	0.235955	0.291729	0.235955	0.225976
Decision Tree	0.41573	0.421978	0.41573	0.409043

Metric	Value
Test Accuracy	0.544444
Test Precision	0.56631
Test Recall	0.544444
Test F1 Score	0.524151

Fig 3.8 Comparison of Evaluation Metrics with 6 Models and Test metric of Best model

The model's accuracy is around 50.6%. This means that approximately 50.6% of the predictions made by the model are correct. The precision here is around 53%, indicating that when the model predicts a class as positive, it is correct about 53% of the time. The recall is approximately 50.6%, meaning the model can correctly identify around 50.6% of all actual positive instances. The F1-score here is around 51.1%, indicating a balance between precision and recall for the model's predictions. The "Random Forest Classifier"model has the highest precision and recall, leading to a relatively balanced F1-score. From the Fig 3.8 it is clear that the higher metric values observed during testing in comparison to the validation phase indicate overfitting in the model.

3.1.5.2 Confusion Matrix

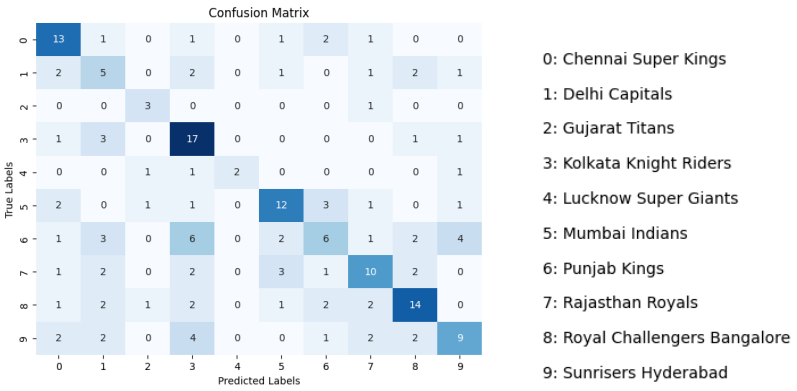


Fig 3.9 Confusion Matrix of Base model

The matrix highlights the model's success in accurately predicting prominent teams, such as Chennai Super Kings, Delhi Capitals, Mumbai Indians, Rajasthan Royals, Royal Challengers Bangalore, and Sunrisers Hyderabad, as indicated by high counts along the diagonal. An interesting finding is that the model often confused Punjab Kings (class 6) with Kolkata Knight Riders (class 3) in several instances. This suggests that there might be similarities between their playing styles or characteristics, which requires further investigation. Another point to note is that the model had difficulty confidently predicting Gujarat Titans (class 2) and Lucknow SuperGiants (class 4), likely due to limited data for these teams. To improve accuracy, we need to address these specific challenges, such as refining features and adding more data for certain teams. This will help the model make better predictions overall.

3.1.5.3 ROC (Receiver Operating Characteristic) Curve and AUC (Area Under Curve) score.

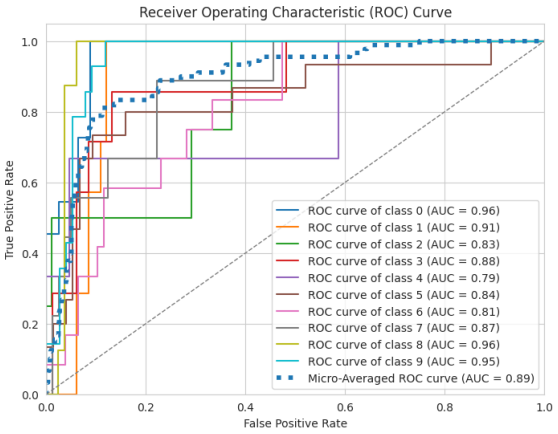


Fig 3.10 ROC curve of Base Model

Firstly, the curves do not tightly hug the left corner, indicating that the model's sensitivity at low false positive rates may be suboptimal. Additionally, the lack of a steep rise in some curves implies that the model's true positive rate does not increase rapidly as the false positive rate rises, reflecting potential areas for enhancement. Moreover, the varying proximity of the ROC curves to the diagonal line underlines the model's varying efficacy in distinguishing between different classes. Although the macro and micro AUC values of 0.88 and 0.89 respectively suggest an overall satisfactory performance, the irregularity in the smoothness of the ROC curves hints at inconsistencies in the model's ability to discriminate among different classes.(AUC 1 means a perfect model.)

3.1.5.4 Overfitting, Underfitting or a Good fit ?

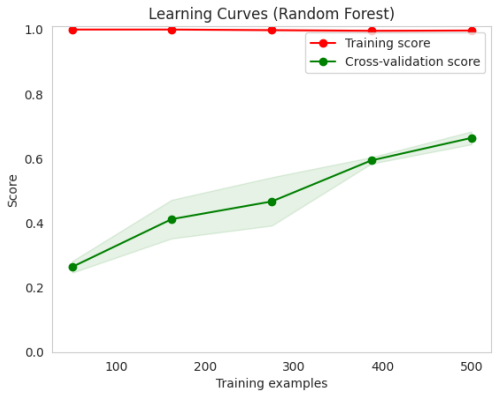


Fig 3.11 Overfitting in the Base model

The training score line maintaining a consistent high value of 1 signifies that the model could be overfitting the training data. However, the increasing but still relatively low cross-validation score suggests that the model struggles to generalize well to unseen data. This divergence between the two curves typically indicates overfitting, where the model has learned the training data too well but struggles to perform well on new, diverse data points.

3.1.5.5 Cohen's Kappa Score

A Cohen's Kappa score of 0.46 indicates a moderate level of agreement between your predicted and actual class labels. This suggests that there is a meaningful agreement between your model's predictions and the actual outcomes, beyond what could be attributed to random chance. It's a positive sign, but still not a good score for the better model.

3.2 Approach II : Enhancing Model Performance through Incorporating Player Performance and Homeground: Derived Features.

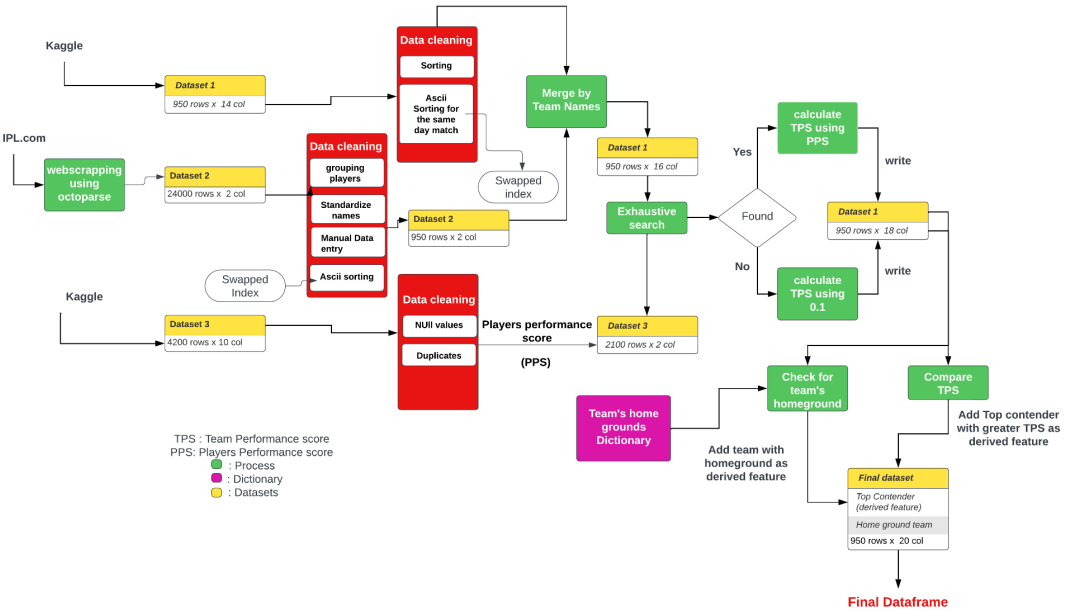


Fig 3.12 Expanded view of Data collection and feature engineering in approach II

The figure in 3.12 illustrates an extended view of the data collection and vigorous feature engineering process conducted in Approach II, while all other processes remain almost consistent with those depicted in baseline model pipeline i.e. Figure 3.1

3.2.1 Data collection

i)Dataset 1: For this approach, we continued using the same dataset mentioned in the previous approach which is the “matches dataset” with size 950 rows.

ii)Dataset 2: Additionally, we employed web scraping using the Octoparse Web Scraping Automator tool to scrap the required dataset from the website named as 'https://www.iplt20.com/matches/results' (www.iplt20.com, n.d.), an official platform of IPL where all the details of the players as well as the teams from 2008 to 2022 can be obtained(Shadiqin Thirafi and Rahutomo, 2018).The scraped dataset comprises totally 30 csv files with almost 24,000 rows in total, with each row representing a player and their respective team in a particular match of

every season. The extracted data will go under different cleaning and preprocessing stages to get the final dataframe.There are totally 30 csv files each representing the players who were a part of that team at that particular year. These dataset should be handled carefully as it represents the historical spanning.

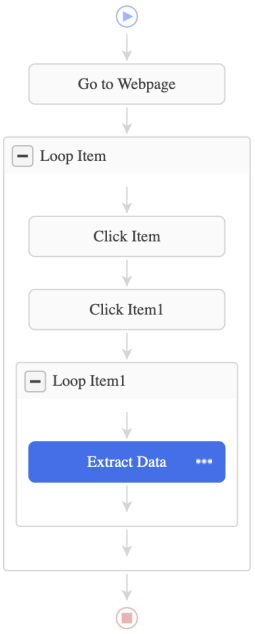


Fig 3.13 WorkFlow of Data Extraction

i) Dataset 3 : The third dataset is obtained from Kaggle, comprising 14 CSV files for batsman data and an additional 14 CSV files for bowlers, resulting in a comprehensive dataset of 4200 rows. Each file represents player performance across individual IPL seasons. This dataset provides historical insights, describing a player's performance within a particular year of a given season.

3.2.2 Data Preprocessing

3.2.2.1 Data Cleaning

i) Dataset 1: This dataset is the same one used previously, with some common data cleaning steps applied. In addition, the following modifications were made:

- **Sorting**: The dataset was sorted by date, and the "year" was extracted from it.
- **Ascii based sorting**: Addressing an issue where matches were played on the same day but at different times, we sorted only those rows by comparing the summed ASCII values of all the letters in the "Team1" and "Team2" columns. The indices of the sorted rows were stored for future reference.

ii) Dataset 2 : Cleaning and structuring the unstructured data in Dataset 2 presented significant challenges.

- **Player Team Alignment:** Grouped players into their respective teams. Addressed cases where some teams had 11 players while others had 10

due to missing values on the website.The size of the dataset 2 reduced to 950 rows by grouping the players into their individual teams.

- **Standardized Player Names:** Converted all player names to "Title Case" to ensure consistent formatting and prevent mismatching during scoring.
- **Manual Data Entry:** Manually entered player values for any missing matches found on the website.
- **Team Name Mapping:** Replaced abbreviated team names with their full counterparts to align with the team names in the "matches dataset".
- **ASCII-Based Sorting:** The previously stored swapped indices from Dataset 1 were applied during the sorting process of Dataset 2. This ensured that rows with corresponding indices were also swapped in Dataset 2, preserving the alignment between teams and their respective players.

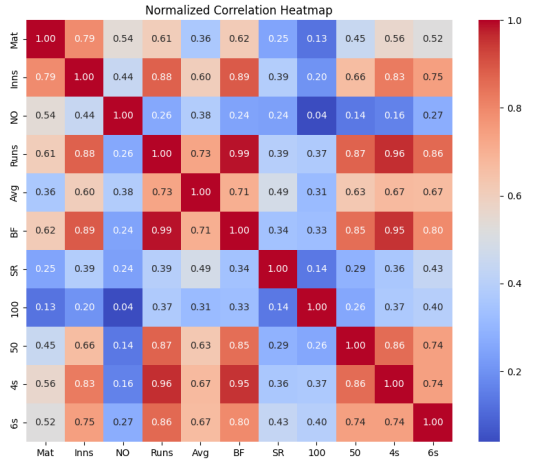
ii) Dataset 3: In dataset 3, both the batsman and bowler datasets were cleaned independently.

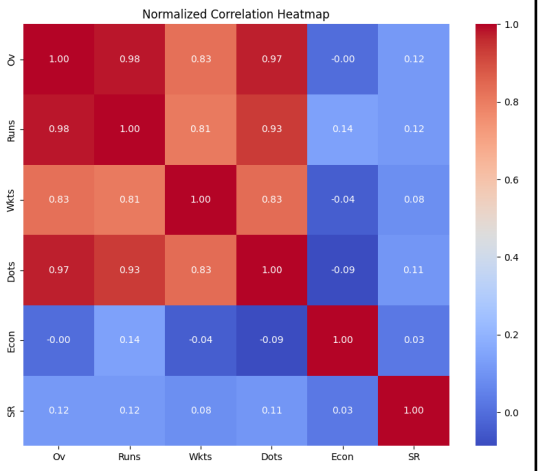
- **Null values** : Null values were removed from both datasets if present.
- **Duplicates:** In the batsman dataset, exact duplicates were identified and removed entirely. However, in the bowlers dataset, player performances were provided for multiple matches within the same season. To address this, columns such as 'Ov' (overs), 'Runs', 'Wkts' (wickets), and 'Dots' were summed, while average-based features like economy and strike rates were averaged.

3.2.2.2 Combining 3 Datasets for Team Performance Evaluation

i) Calculation of Player Performance Score in Dataset 3: Each feature within Dataset 3 was assigned a unique weight based on its correlation score, derived from the correlation matrix. These correlation scores quantified the relationships between different features.

Batsman Correlation Scores	Bowler Correlation scores
• Normalized correlation heatmap	• Normalized correlation heatmap





- Correlation scores for batsman's dataset derived from correlation heatmap.

Column	Weight
Mat	0.085
Inns	0.109
NO	0.054
Runs	0.116
Avg	0.096
BF	0.113
SR	0.063
100	0.052
50	0.099
4s	0.109
6s	0.105

- Correlation scores for bowlers's dataset derived from correlation heatmap.

Column	Weight
Ov	0.221
Runs	0.226
Wkts	0.199
Dots	0.213
Econ	0.059
SR	0.082

Subsequently, a weighted average was computed by multiplying each feature with its corresponding assigned correlation score. This resultant weighted sum constituted the **Player Performance Score (PPS)**.

PPS =(F1 *w1 +F2*2 +.....Fn*wn) / Total weights,

where F1, F2…Fn stands for features and w1, w2…wn stands for weights (www.sciencedirect.com, n.d.).

These Player Performance Scores were subsequently normalized to ensure a consistent and comparable representation of player performance across different seasons and teams. This process was executed for every player across all seasons spanning from 2008 to 2021, ensuring a comprehensive evaluation of player performance.

	Player	Year	Scaled_Score
0	Shaun Marsh	2008	0.675587
1	Gautam Gambhir	2008	0.586725
2	Sanath Jayasuriya	2008	0.327266
3	Shane Watson	2008	0.459372
4	Graeme Smith	2008	0.516421
...	...	...	...
2771	Imran Tahir	2021	0.069986
2799	Umran Malik	2021	0.085909
2806	Riley Meredith	2021	0.066646
2810	Amit Mishra	2021	0.084060
2812	Josh Hazlewood	2021	0.088672

2092 rows × 3 columns

Fig 3.14 Final dataset 3 showing Player Performance Score (PPS) of each player

ii) Merging dataset 2 with dataset 1: The players belonging to their respective teams in Dataset 2 are combined with Dataset 1, using "Team names" as the primary key for merging. Due to meticulous sorting prior to merging, it is guaranteed that the matches in Dataset 1 align seamlessly with the players' information in Dataset 2. This alignment ensures that players are correctly assigned to their respective teams for each match, preventing any cross-feature contamination. Maintaining consistency in dimensions, the number of rows in Dataset 1 matches that of Dataset 2, totaling 950 rows (encompassing all matches played thus far).

iii) Calculation of Team Performance Score (TPS) in dataset 1 : Before proceeding to calculate TPS, we must consider these Cases.

- **Case 1:** It is essential to confirm the presence of all players from the merged Dataset 1 in the Players Performance Score dataframe.

  Solution: Performed an exhaustive search between the player in the Dataset 1 and dataset 2 . The search resulted in a match of 500 unique players out of 561. The remaining player who was not present in the player performance dataset is considered to be New player and given him a Performance Score of 0.1.

- **Case 2 :** Player performance varies from season to season, resulting in variations in performance scores. A player who excelled in the previous year might perform poorly in the current year, while a player who struggled in the past might excel.

  solution: To address this, an approach is developed to calculate an aggregated performance score for a player in a specific year, accounting for their past performances. The solution involves assigning different weights to past performance scores based on the season.

  1. The previous season's performance score is assigned a higher weight (e.g., 0.6) since a player's current performance often relies significantly on their performance in the preceding year.
  2. The performance score for the ongoing season carries a weight of 0.3, considering that players are just starting a new season. Assigning a high weight to the current year could introduce future leakage, wherein the player's future performance influences their present score.
  3. The performance scores for seasons further in the past receive a weighting of 0.1, recognizing their diminishing impact on the current performance score.

By addressing all these cases, we have successfully managed historical data, handled missing data, and mitigated the risk of future leakage.

Team Performance Score (TSP) is computed by aggregating the performance scores of all players within a specific team, with careful consideration of the various cases outlined earlier. This TSP serves as an indicator of the team's overall performance in the current match. Notably, the team with the highest TSP is likely to have a greater chance of emerging victorious.

Aggregated Score of Player 1(ASP1) = [0.3*PPS + 0.6*PPS + 0.1*PPS + 0.1*PPS+.....+till 2008]

This process is repeated for all players, resulting in ASP2, ASP3, and so on. The TSP of Team 1 is obtained by summing up the ASP values for all its players.

TPS of Team 1 = [ASP1+ASP2+ASP3…….+ASP4]

A similar calculation is performed for Team 2 to derive its TSP.

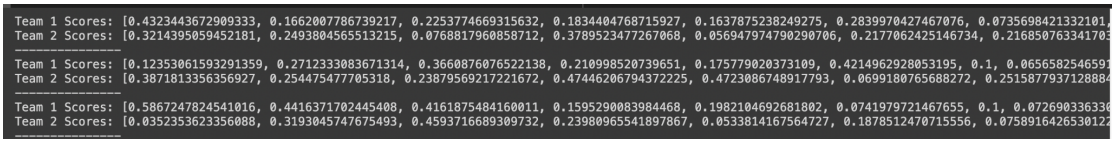


Fig 3.15 ASP1,ASP2,ASP3…..ASPn scores of all team players

This TPS score is used to derive the new feature called " Top Contender" which is added to indicate that it is a strong candidate for victory. This innovative feature helps identify the team that stands a higher likelihood of winning the match. By combining the historical and current performance data of players within a team, the TPS score provides insights into a team's overall strength. This, in turn, enables us to forecast the probable winner of the match.

3.2.2.3 Introducing new derived Feature: "HomeGround Team"

In this section, we introduce the concept of the "HomeGround Team." HomeGround signifies the venue where a team's match takes place. Teams often perform better on their home ground due to factors like crowd support, familiarity with conditions, and reduced travel stress. We explore how this HomeGround Advantage affects team performance and identify which teams gain the most from this edge. Even this section leads to cases

- **Case 1: Team with multiple home grounds.**

  Solution: A dictionary with "Team name" as key and their "home grounds" as values is maintained. We iterated through all the values of the "venue" column of the matches dataset (dataset 1) to get the respective team which has their venue as homeground.

- **Case 2 : Venue which has multiple names**

  solution: Replacing the duplicate names to original names.

3.2.2.4 Feature enhancement: TossDecison to "Batting_Team" and "Bowling_team"

In the previous approach, the toss decision was deemed less important based on various feature selection methods such as LASSO, Random Forest Features ,gradient Boosting and its limited correlation with the winning team. To enhance its significance, a new feature called "Batting_team" and 'Bowling_team" was derived from the toss decision.

- **Case 1 :** If the toss winner chose to bat, the team batting first becomes the batting team, and the opposing team becomes the bowling team.
- **Case 2:** Conversely, if the toss winner chose to bowl, the roles are reversed.

This process ensures that the toss decision, which was previously excluded, now contributes to the analysis by assigning teams specific roles based on the toss outcome.

	Team1	Team2	TossWinner	WinningTeam	Top_Contender	Batting_team	Bowling_team	home_ground_team
0	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	Kolkata Knight Riders	Royal Challengers Bangalore	Royal Challengers Bangalore
1	Punjab Kings	Chennai Super Kings	Chennai Super Kings	Chennai Super Kings	Chennai Super Kings	Chennai Super Kings	Punjab Kings	Punjab Kings
2	Delhi Capitals	Rajasthan Royals	Rajasthan Royals	Delhi Capitals	Delhi Capitals	Rajasthan Royals	Delhi Capitals	Delhi Capitals
3	Kolkata Knight Riders	Sunrisers Hyderabad	Sunrisers Hyderabad	Kolkata Knight Riders	Sunrisers Hyderabad	Sunrisers Hyderabad	Kolkata Knight Riders	Kolkata Knight Riders
4	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians	Royal Challengers Bangalore	Royal Challengers Bangalore	Mumbai Indians	Royal Challengers Bangalore	Mumbai Indians
...	...	...	...	...	...	...	...	...
945	Sunrisers Hyderabad	Punjab Kings	Sunrisers Hyderabad	Punjab Kings	Punjab Kings	Sunrisers Hyderabad	Punjab Kings	Punjab Kings
946	Rajasthan Royals	Gujarat Titans	Gujarat Titans	Gujarat Titans	Rajasthan Royals	Rajasthan Royals	Gujarat Titans	Gujarat Titans
947	Royal Challengers Bangalore	Lucknow Super Giants	Lucknow Super Giants	Royal Challengers Bangalore	Royal Challengers Bangalore	Royal Challengers Bangalore	Lucknow Super Giants	Royal Challengers Bangalore
948	Royal Challengers Bangalore	Rajasthan Royals	Rajasthan Royals	Rajasthan Royals	Rajasthan Royals	Royal Challengers Bangalore	Rajasthan Royals	Rajasthan Royals
949	Rajasthan Royals	Gujarat Titans	Rajasthan Royals	Gujarat Titans	Rajasthan Royals	Rajasthan Royals	Gujarat Titans	Gujarat Titans

946 rows × 8 columns

Fig 3.16 Final dataset with all the derived features

3.2.2.5 Validation of Derived Features: Exploring Enhanced Correlation with Winning Team through correlation HeatMap

The correlation heatmap clearly highlights the significant influence of the derived features on the “WinningTeam”. Notably, the top contender exhibits a substantial correlation of 0.56, signifying a 56% impact. "Batting_team" and "bowling_team" reflect enhanced correlations of 38% and 48%, respectively, surpassing the previous TossDecision's impact. Most remarkably, the "Home_ground_team" boosts a remarkably high correlation of 67%, underscoring its substantial role.

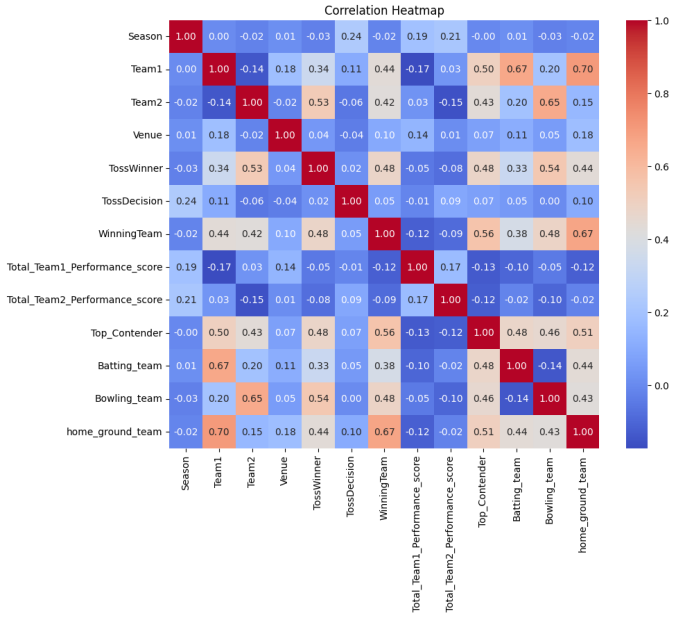


Fig 3.17 Correlation heatmap showing the correlation of newly derived features

3.2.2.6 Splitting the dataset into Train,Validation and Test

The data was divided into three sets: training, validation, and test, using an 80/10/10 split. To mitigate the constraints of limited data, a larger portion was allocated for training the model.

3.2.2.7 Handling Imbalanced Data Using RandomOver, RandomUnder, SMOTE and SMOTEENN .

In this section, we focus on handling imbalanced data, which is essential for accurate model performance. An initial class imbalance is observed, particularly in the case of new teams, such as "Gujarat Titans" and "Lucknow Super Giants," resulting in fewer instances. To address this, we apply various resampling strategies: SMOTE (Synthetic Minority Over-sampling Technique), SMOTEENN (SMOTE combined with Edited Nearest Neighbors), RandomOverSampling, and RandomUnderSampling(Kumar, Lalotra and Kumar, 2022). We evaluated these techniques' effectiveness using key metrics such as Accuracy, Precision, Recall, and F1-score. The results indicate improvements in certain metrics after applying these techniques. For instance, RandomOverSampling shows enhanced performance in terms of Accuracy, Precision, Recall, and F1-score compared to the original dataset.

	Resampling Strategy	Accuracy	Precision	Recall	F1-score
0	Original Data	0.605263	0.627320	0.605263	0.609149
1	SMOTE	0.610526	0.622663	0.610526	0.613255
2	SMOTEENN	0.468421	0.580997	0.468421	0.472543
3	RandomOver	0.621053	0.635632	0.621053	0.623470
4	RandomUnder	0.536842	0.572532	0.536842	0.547012

Fig 3.18 Comparision Between Class Imbalancing Techniques

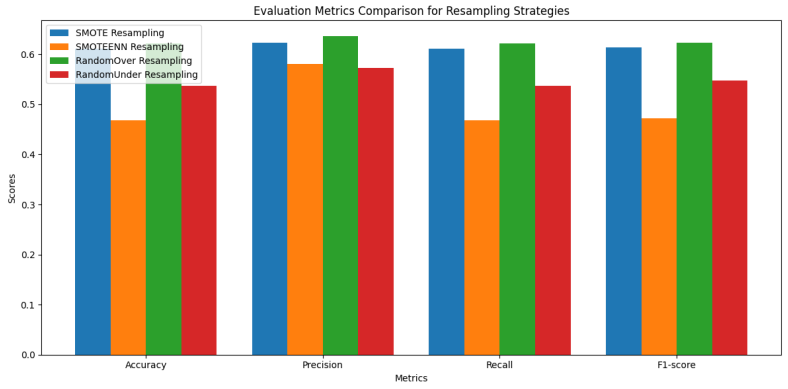


Fig 3.19 Bar graph showing RandomOver sampling is performing better than other techniques

3.2.3 Training the model and performing Grid Search and Hyper parameter Tuning

We revisited the familiar machine learning models and their hyperparameters, keeping the approach consistent for comparison. However, this time we incorporate the RandomOverSampler technique due to its superior performance observed in the previous checkpoint. After the RandomOverSampling process, the models undergo Grid Search with the same hyperparameters. We evaluate their performance on the validation set and record key metrics such as Accuracy, Precision, Recall, and F1-score. The best model, which demonstrates the highest

scores across these metrics, is retained for further analysis. Subsequently, this model is evaluated on the previously unseen test set.

3.2.4 Evaluation of Model

3.2.4.1 Accuracy, Precision, Recall and F1 Score.

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.687831	0.699618	0.687831	0.68559
SVM	0.560847	0.571485	0.560847	0.559267
KNN	0.428571	0.473048	0.428571	0.436273
Logistic Regression	0.275132	0.297249	0.275132	0.26506
Naive Bayes	0.291005	0.347346	0.291005	0.268642
Decision Tree	0.587302	0.601662	0.587302	0.587659

Metric	Value
Test Accuracy	0.678322
Test Precision	0.675701
Test Recall	0.678322
Test F1 Score	0.669916

Fig 3.20 Evaluation metrics showing performance of all models and Test metric of Best model

After incorporating the derived features "Top contender" and "home_ground_team," we observe notable improvements in the performance metrics across the models. The accuracy for Random Forest increased to almost 69%, with precision 70%, recall 69%, and F1-score 68%. The addition of these derived features seems to provide the models with more discriminative information, allowing them to better identify patterns related to being a "Top contender" and playing on "home_ground_team." This enhanced understanding contributes to the models' ability to make more informed predictions, leading to improved accuracy, precision, recall, and F1-score. From Fig 3.20 we can say that the test metric is almost same as validation metric which shows that the model is perfectly fit.

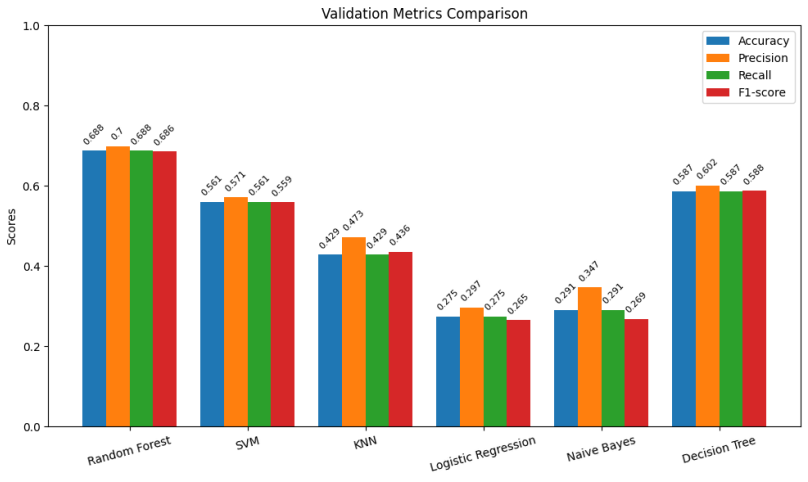


Fig 3.21 Bar graph showing RandomForest is performing better than other models

3.2.4.2 Confusion matrix

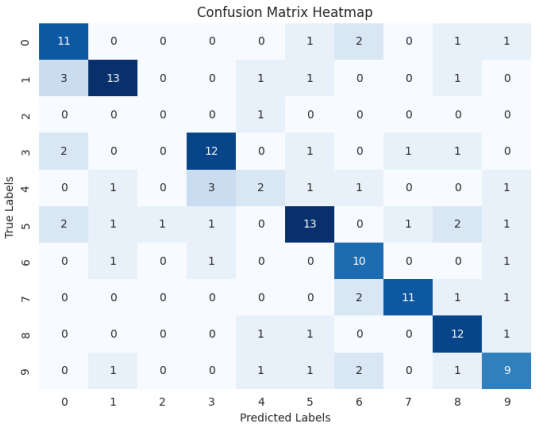


Fig 3.22 Confusion matrix showing improvement in the Prediction On test data

Contrary to the confusion matrix of the base model, the approach 2 model shows a considerable increase in accuracy and precision. The model's improved ability to predict well-known teams, such as Chennai Super Kings, Delhi Capitals, Mumbai Indians, Rajasthan Royals, Royal Challengers Bangalore, and Sunrisers Hyderabad, is indicated by the diagonal alignment of high counts. Importantly, the new model has successfully improved learning by dramatically reducing the previous misunderstanding between Kolkata Knight Riders (class 3) and Punjab Kings (class 6).

3.2.4.3 *ROC (Receiver Operating Characteristic) Curve and AUC (Area Under Curve) score.*

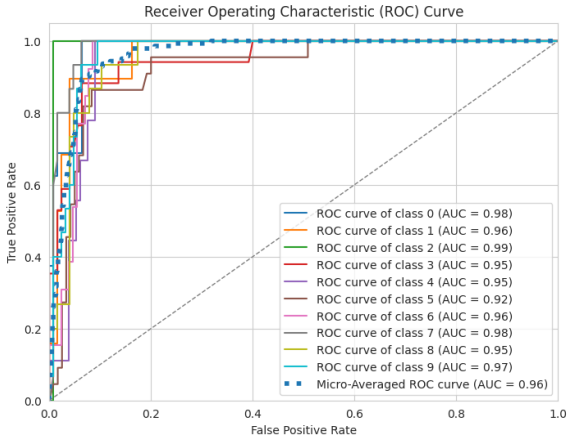


Fig 3.23 ROC curve and AUC score, improvements from Base model

The ROC curve for the improved model shows a significant improvement,with a very close to the left corner and rising steeply. The macro and micro AUC values of 0.96 indicate strong overall performance. The model's extraordinary capacity for class distinction is indicated by the ROC curves' constant higher position above the diagonal. The curve's smoothness also indicates that the model

consistently predicts correctly across a range of class probabilities. These positive indicators collectively suggest that the refined approach has effectively increased the model's ability to discriminate between different classes, resulting in significantly improved predictive accuracy.

3.2.4.4 Overfitting, Underfitting or a Good fit ?

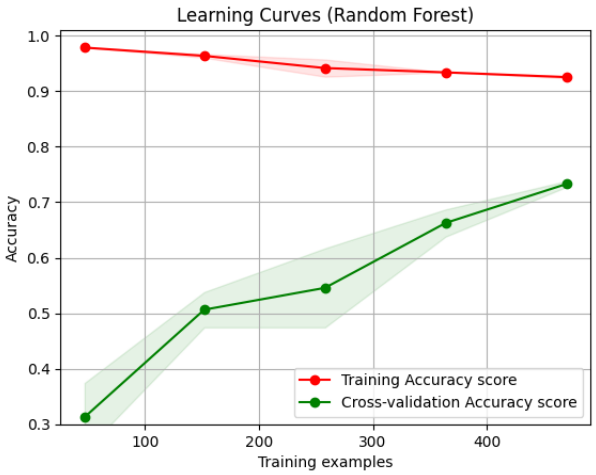


Fig 3.24 Converging Learning Curve showing Good Fit

The training score starts relatively high and gradually decreases to a stable level around 0.9, showcasing the model's ability to fit the training data well while avoiding overfitting. Conversely, the validation curve starts low but steadily rises and stabilizes at around 0.75, indicating improved generalization as the model encounters more data. The convergence of both curves indicates that the model has reached a point where further exposure to data does not significantly impact its performance. This behavior suggests that the model is effectively balancing between fitting the training data well and generalizing to unseen data, indicating a more optimal fit compared to the previous approach.

3.2.4.5 Cohen's Kappa Score

When we examined the performance of our model in predicting IPL (Indian Premier League) winners, we found that it achieved a Cohen's Kappa Score of 0.62. This score indicates strong agreement between the predictions made by our model and the actual results of IPL matches. In simpler terms, our model is good at making predictions about which team is likely to win an IPL match, and this level of accuracy is better than random chance."

Chapter 4 Conclusion

4.1 Discussion

In conclusion, predicting the winner of IPL cricket matches, a multiclass prediction problem, is challenging due to its high randomness. This randomness became evident during the project implementation and is reflected in the evaluation metrics and learning curve analysis performed before feature engineering. The small dataset made this challenge even tougher. Because of its small size, the model struggled to find clear patterns and make accurate predictions. This made it much harder to achieve high accuracy in our predictions. However, through exceptional feature engineering, we managed to achieve outstanding accuracy in this multiclass prediction problem.

From this project, it is clear that winning a match depends on several major factors, including player performance: If they play exceptionally well and show some good strengths and skill, then their team has a good chance of winning the match. Hence this factor has helped us in improving the accuracy exceptionally high. The pitch conditions, and even the weather are also considered to have a high influence in winning the match. Minor factors, such as 'Toss_winner,' 'Toss_decision,' and 'Homeground,' also play a role, though to a lesser extent. But it is always known in cricket that in certain pitches and ground's condition, the team who won the toss and then took a correct decision either to bat or bowl first has a greater extent to win the match.

Our project aimed to address issues identified in prior research, particularly avoiding the use of future data for present predictions and improving feature engineering. To address the mistakes made by previous work, we implemented a baseline model using a similar approach. This baseline model served as a point of comparison for our ultimate predictive model. Additionally, it helped us identify any drawbacks in our approach, allowing us to make improvements in our final model.

A significant focus of our work was dedicated to feature engineering in the final model, resulting in the development of 'Top Contender' from the performance score of the individual players and 'HomeGround' from the 'venue' as a derived feature in predicting match winners. This feature proved to be more influential than others in determining match outcomes as the performance of individual players directly correlates with the team's chances of winning.

By effectively managing historical and future player performance scores, we have created a robust prediction model that not only provides accurate results but also delivers high precision in predictions.

4.2 Key Findings of the Project

1. Achieved improved accuracy in a multiclass prediction problem despite the challenges posed by a small dataset, addressing concerns of feature leakage and model overfitting.

2. Developed a team performance scoring system that assigns scores to teams based on the players they field, resulting in the creation of a new feature known as "Top contender," which directly influences a team's likelihood of winning. This feature boosted the accuracy from 46% to 63%.
3. Introduced a novel feature called 'homeground team' to identify the team associated with a specific ground, revealing that teams playing on their home ground have a higher probability of winning.This feature boosted the accuracy from 63% to almost 70%.
4. Conducted rigorous experiments on prior research to identify voids in existing methodologies, substantiating claims with proper evidence.

4.3 Contributions to the field , Implications and Applications

1. **Data Resource:** Our Team players dataset with player performance scores of every year can be utilized by cricket analysts, researchers, and enthusiasts for various analytical purposes, contributing to a deeper understanding of the sport.
2. **Match Predictions:** Provides a predictive model capable of pre-match winner prediction using only pre-match data such as teams, toss winner, toss decision, players, and venue, achieving an accuracy of 70%.
3. **Betting Confidence:** Sports bettors can confidently use our predictive model to inform their betting decisions, potentially increasing their chances of success.
4. **Player Selection:** IPL team owners can use our player performance scoring system to make data-driven decisions when selecting players, ultimately strengthening their teams.

4.4 Limitations

1. **Retired Players:** The project relies on historical player performance data, but it doesn't account for players who may have retired since their last appearance. This limitation could affect the accuracy of predictions, as retired players' past data becomes irrelevant.
2. **New Players:** Every year, a significant number of new players are added to the IPL teams. As time passes, relying solely on current data to calculate a team's future performance becomes insufficient. This results in encountering different players who are not present in the current performance score datasets. Consequently, our approaches become less reliable and less adaptable to upcoming future data.
3. **Scoring New Players:** Scoring new players poses a substantial limitation, as these players bring unknown previous experiences and performance histories. Assigning a meaningful score under these circumstances is a challenging task. For this project, we have considered a starting score of 0.1 for new players, which, in turn, leads to another limitation outlined below.
4. **Multiple New Players:** Assigning a starting score of 0.1 to a new player in their first IPL match might lead to situations where a team has multiple high-performing new players, but the total team performance score would be comparatively lower than that of another team. This scenario could potentially introduce confusion in the prediction process, although such cases are rare in IPL history.

4.5 Future Research

1. Incorporating pitch conditions into the predictive model to determine which team performs better on different types of pitches, leading to more accurate predictions.
2. Investigating the impact of weather conditions on team performance, including how teams fare in moist and humid environments versus dry conditions. Additionally, considering the influence of temperature on player performance.
3. Exploring the scoring of players based on their physical attributes, weaknesses, energy levels, age, and other physical and mental features that contribute to a player's ability to perform well in a match.

Appendix A Exploratory Data Analysis

EDA of Dataset used for Base Model

Dataset statistics

Number of variables	18
Number of observations	950
Missing cells	1012
Missing cells (%)	5.9%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	988.1 KiB
Average record size in memory	1.0 KiB

Variable types

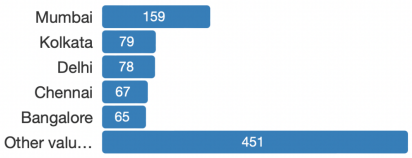
Numeric	2
Categorical	15
Boolean	1

City

Categorical

HIGH CORRELATION MISSING

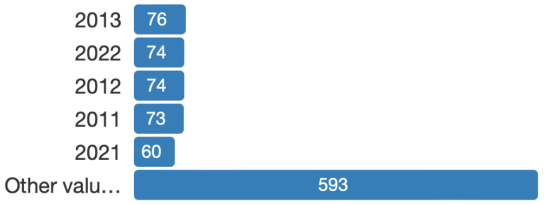
Distinct	33
Distinct (%)	3.7%
Missing	51
Missing (%)	5.4%
Memory size	58.2 KiB



Season

Categorical

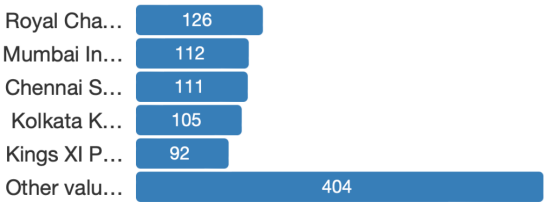
Distinct	15
Distinct (%)	1.6%
Missing	0
Missing (%)	0.0%
Memory size	57.2 KiB



Team1

Categorical

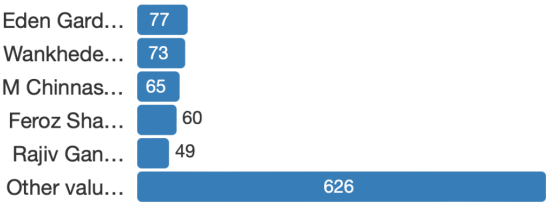
Distinct	18
Distinct (%)	1.9%
Missing	0
Missing (%)	0.0%
Memory size	69.8 KiB



Venue

Categorical

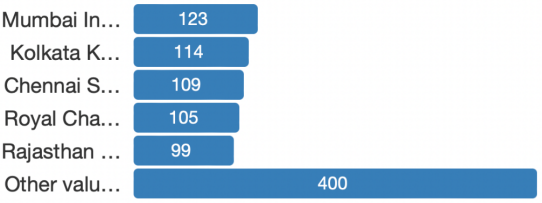
Distinct	49
Distinct (%)	5.2%
Missing	0
Missing (%)	0.0%
Memory size	77.0 KiB



TossWinner

Categorical

Distinct	18
Distinct (%)	1.9%
Missing	0
Missing (%)	0.0%
Memory size	69.6 KiB



TossDecision

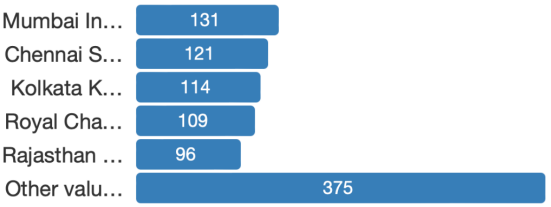
Categorical

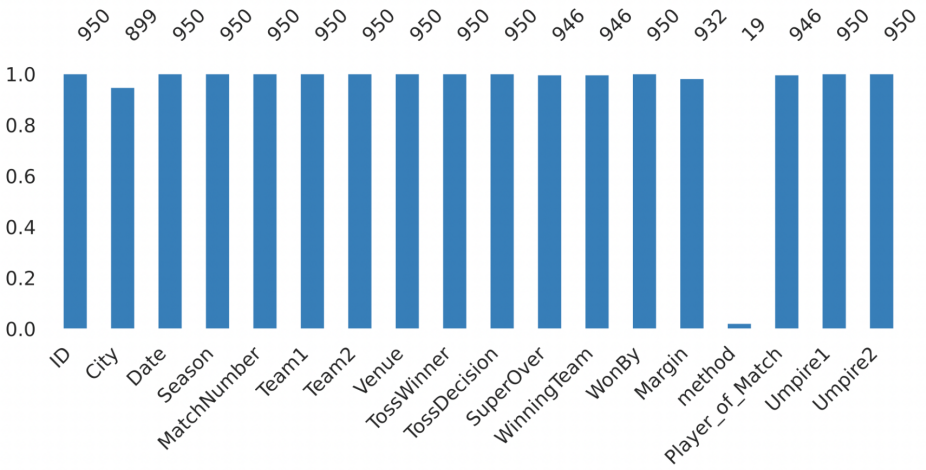
Distinct	2
Distinct (%)	0.2%
Missing	0
Missing (%)	0.0%
Memory size	57.0 KiB

WinningTeam

Categorical

Distinct	18
Distinct (%)	1.9%
Missing	4
Missing (%)	0.4%
Memory size	69.6 KiB





A simple visualization of nullity by column.

EDA of Dataset used for Final Model after feature engineering

Dataset statistics

Number of variables	16
Number of observations	950
Missing cells	986
Missing cells (%)	6.5%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	1.1 MiB
Average record size in memory	1.2 KiB

Variable types

Categorical	13
Numeric	3

players_team1
Categorical

HIGH CARDINALITY UNIFORM

Distinct	913
Distinct (%)	96.1%
Missing	0
Missing (%)	0.0%
Memory size	228.2 KiB

['Faf Du Ples… 3
['Rohit Shar… 3
['Manish Pan… 2
['Gautam Ga… 2
['Brendon M… 2
Other values … 938

players_team2
Categorical

HIGH CARDINALITY UNIFORM

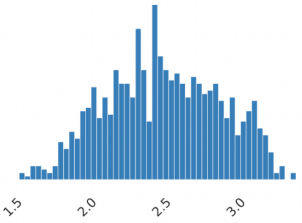
Distinct	924
Distinct (%)	97.3%
Missing	0
Missing (%)	0.0%
Memory size	227.7 KiB

['Parthiv Pate… 3
['Faf Du Ples… 2
['Shane Wats… 2
['Virat Kohli', … 2
['Lendl Simm… 2
Other values … 939

Total_Team1_Performance_score

Real number (ℝ)

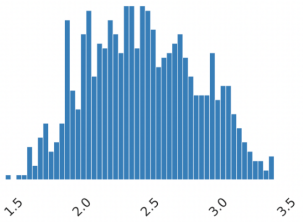
Distinct	808	**Minimum**	1.5405019
Distinct (%)	85.1%	**Maximum**	3.3998502
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	2.5091099	**Memory size**	7.5 KiB

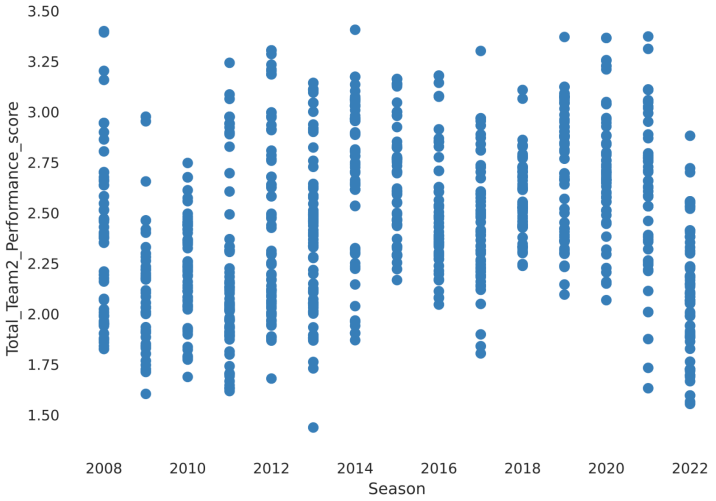


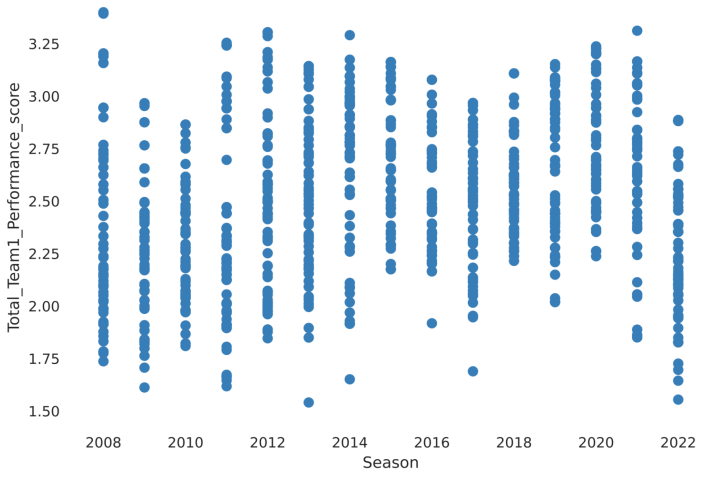
Total_Team2_Performance_score

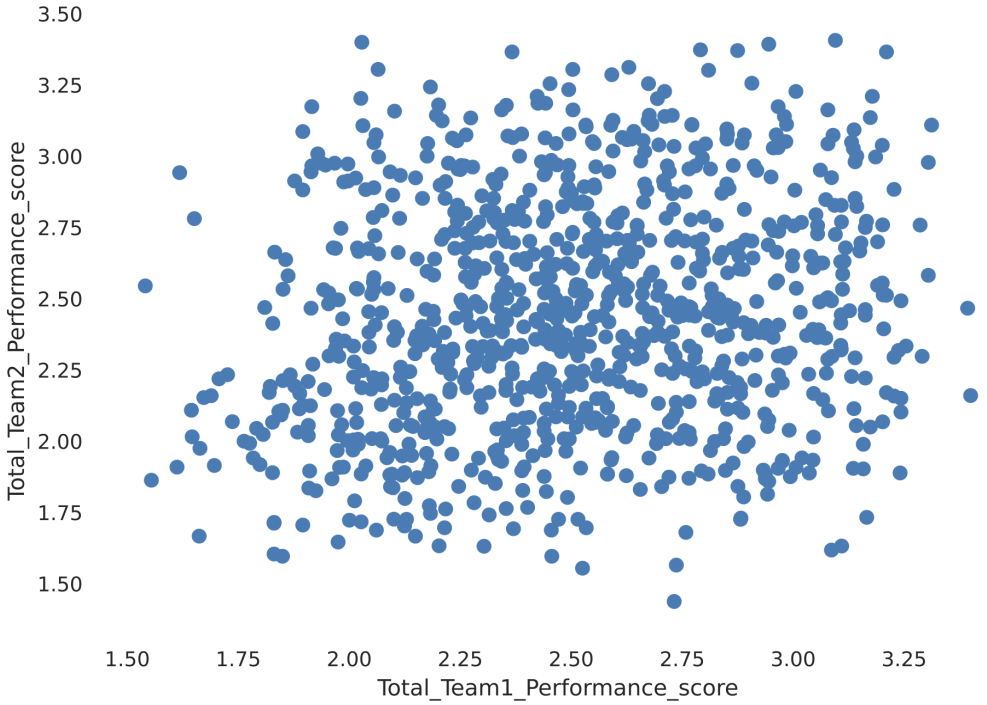
Real number (ℝ)

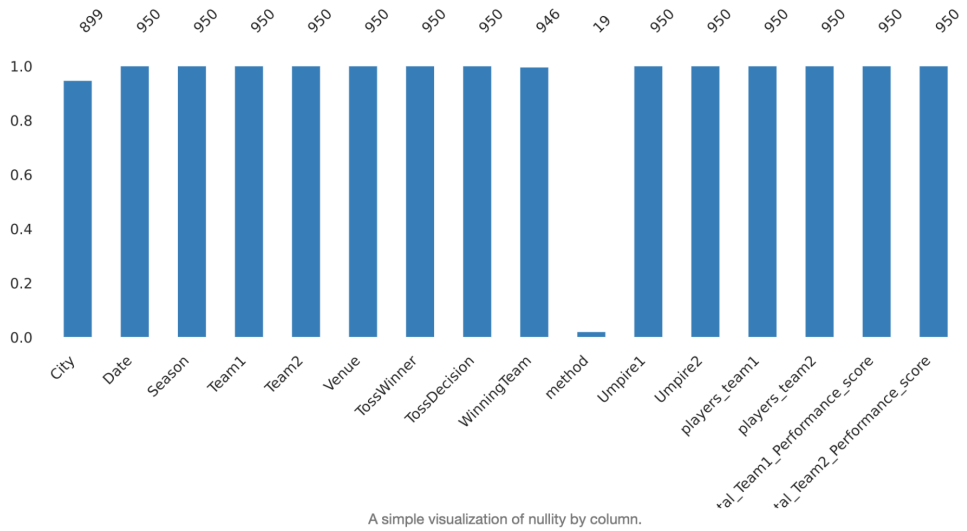
Distinct	854	**Minimum**	1.437399
Distinct (%)	89.9%	**Maximum**	3.406548
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	2.4448352	**Memory size**	7.5 KiB








Total_Team2_Performance_score
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.50
1.75
2.00
2.25
2.50
2.75
3.00
3.25
Total_Team1_Performance_score


899
950
950
950
950
950
950
950
946
19
950
950
950
950
950
950
1.0
0.8
0.6
0.4
0.2
0.0
City
Date
Season
Team1
Team2
Venue
TossWinner
TossDecision
WinningTeam
method
Umpire1
Umpire2
players_team1
players_team2
tal_Team1_Performance_score
tal_Team2_Performance_score
A simple visualization of nullity by column.

References

Agrawal, S., Singh, S.P. and Sharma, J.K. (2018). *Predicting Results of Indian Premier League T-20 Matches using Machine Learning*. [online] IEEE Xplore. doi:https://doi.org/10.1109/CSNT.2018.8820235.

Barot, H., Kothari, A., Bide, P., Ahir, B. and Kankaria, R. (2020). Analysis and Prediction for the Indian Premier League. *2020 International Conference for Emerging Technology (INCET).* doi:https://doi.org/10.1109/incet49848.2020.9153972.

Chaudhary, R., Bhardwaj, S. and Sakshi Lakra (2019). A DEA Model for Selection of Indian Cricket Team Players. doi:https://doi.org/10.1109/aicai.2019.8701424.

Chaudhary, S. (2020). *Why '1.5' in IQR Method of Outlier Detection?* [online] Medium. Available at: https://towardsdatascience.com/why-1-5-in-iqr-method-of-outlier-detection-5d07fdc82097.

Darji, H. and Dave, P. (2013). IPL betting: White collar pros join the high-roller rush, turnover to touch Rs 40,000 cr. *The Economic Times*. [online] 15 Apr. Available at: https://economictimes.indiatimes.com/ipl-betting-white-collar-pros-join-the-high-roller-rush-turnover-to-touch-rs-40000-cr/articleshow/19551375.cms?from=mdr [Accessed 29 Aug. 2023].

Editor, S. (2023). *The Scale of Cricket Betting Online in India: Growth Over the Years & The Future*. [online] sportzwiki.com. Available at: https://sportzwiki.com/news/the-scale-of-cricket-betting-online-in-india-growth-over-the-years-the-future/ [Accessed 29 Aug. 2023].

ESPNcricinfo. (n.d.). *A glossary of cricket terms*. [online] Available at: https://www.espncricinfo.com/story/a-glossary-of-cricket-terms-239756.

Kumar, V., Lalotra, G.S. and Kumar, R.K. (2022). Improving performance of classifiers for diagnosis of critical diseases to prevent COVID risk.

Computers and Electrical Engineering, [online] 102, p.108236. doi:https://doi.org/10.1016/j.compeleceng.2022.108236.

Shadiqin Thirafi, Moch.F. and Rahutomo, F. (2018). *Implementation of Naïve Bayes Classifier Algorithm to Categorize Indonesian Song Lyrics Based on Age*. [online] IEEE Xplore. doi:https://doi.org/10.1109/SIET.2018.8693201.

Sharma, V. (2022). *Pandas Profiling - A Visual Analytics Wonder*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2022/08/pandas-profiling-a-visual-analytics-wonder/#:~:text=The%20Pandas%20Profiling%20Li [Accessed 29 Aug. 2023].

Srikantaiah, K.C., Aryan Khetan, Kumar, B., Tolani, D. and Patel, H. (2021). Prediction of IPL Match Outcome Using Machine Learning Techniques. *arXiv (Cornell University)*. doi:https://doi.org/10.2991/ahis.k.210913.049.

Vinutha, H.P., Poornima, B. and Sagar, B.M. (2018). Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. *Advances in Intelligent Systems and Computing*, 701, pp.511–518. doi:https://doi.org/10.1007/978-981-10-7563-6_53.

www.iplt20.com. (n.d.). *Indian Premier League Official Website*. [online] Available at: https://www.iplt20.com/matches/results [Accessed 27 Aug. 2023].

www.sciencedirect.com. (n.d.). *Weighted Sum Method - an overview | ScienceDirect Topics*. [online] Available at: https://www.sciencedirect.com/topics/computer-science/weighted-sum-method#:~:text=In%20the%20weighted%20sum%20method [Accessed 29 Aug. 2023].