IPLitics

Project-2 Report

SUBMITTED IN PARTIAL FULFILLMENT REQUIREMENT FOR

THE AWARD OF DEGREE OF

Bachelor of Technology
(COMPUTER SCIENCE & ENGINEERING)

BML MUNJAL
UNIVERSITY™
FROM HERE TO THE WORLD

SCHOOL OF ENGINEERING AND TECHNOLOGY

BML MUNJAL UNIVERSITY Gurugram,
Haryana - 122413

May 2023

IPLitics

Group Members:

Ridit Jain	***CSE-3***	***210C2030184***
Lavanya Anand	*CSE-2*	*210C2030072*
Akshay Kumar Jain	*CSE-1*	*210C2030006*
Ashi Jain	*CSE-3*	*210C2030165*
Priyansh Jain	*CSE-3*	*210C2030138*
Vishav Garg	*CSE-3*	*210C2030213*
Akshat Manohar	*CSE-3*	*210C2030162*
Aditi Singh	*CSE-1*	*210C2030120*
Runishka Rao	*CSE-3*	*210C2030134*
Prashant Sharma	*CSE-3*	*210C2030192*

1. Introduction

1.1 **Motivation: -**

IPLitics is a one of a kind app which has been developed by our team with the help of a unique and distinctive approach to formulate the various outcomes of a certain IPL match, several major features concerning a match such as runs scored by a particular batsman, strike rate, etc. One of the main reasons behind its creation is giving the users a reliable platform to analyse their team's detailed and in-depth performance. It helps the users stay ahead of everyone else by having a user-friendly interface as well as a highly reliable one as well.

IPLitics is not just a prediction app; it is a combination of various features for cricket fans who always want to stay on the top of their game. It provides in depth information and data on team and player performance, allowing enthusiasts to have a brief knowledge of the game. Our team has collectively worked putting all our effort in developing the app. The app uses sophisticated machine learning algorithms to predict the score of upcoming matches. We are committed to providing the best possible experience to our users.

1.2 What is our product: -

IPLitics is an app made for cricket fans that will provide accurate prediction for different IPL matches results .The software uses different algorithms and models to generate exact results for the upcoming matches based on the current team and individual statistics .It uses a variety of fields to make predictions including strike rate, runs conceded ,balls faces ,wicket taken and many more. Adding to it ,the app provides details on current form ,all the previous records against the opposition team and other important factors that may affect the outcome of the match. The platform also has a real time score tracker that keeps the user up to the mark of the on going cricket matches progress.

1.3 Why do we need this product: -

IPLitics is a must-have for cricket fans since it delivers a dependable and accurate Indian Premier League prediction. The application evaluates team and individual player statistics using sophisticated algorithms and methods based on machine learning, providing users with reliable data to make informed predictions. The app's precise forecasts assist users in increasing their odds of winning and making profitable selections. Furthermore, IPLitics can assist cricket analysts and commentators in providing insights and predictions for upcoming matches, improving their accuracy and credibility. The app can also improve cricket fans' entire viewing experience by giving live scores and statistics, keeping them interested and informed throughout the game.

1.4 Real World Implications

All IPLitics really means is that it assists consumers in making educated choices while betting on IPL games. The app's precise forecasts boost consumers' odds of victory and assist them in making successful selections, helping teams in their statistical analysis and average performance of any player.

1.5 Survey of existing solutions

There are several prediction applications for IPL, but their reliability and precision could differ. Some of these apps might utilize simple mathematical computation, while others may use more sophisticated algorithms and machine learning techniques. The practicality of such apps will ultimately be determined by the standards of the data implemented to produce predictions and the sophistication of the algorithms deployed.

- **Cricadium:** Cricadium is an IPL prediction app that provides users with accurate predictions for upcoming matches based on team and player performances. The app uses a combination of statistical analysis and machine learning techniques to make predictions. The dataset used by Cricadium is collected from various sources, including player and team statistics, historical match data, and expert analysis. The app evaluates the usability of the product by providing users with an easy-to-use interface and real-time updates during the match.

- **Cricket** Mazza: Cricket Mazza is another IPL prediction app that provides users with accurate predictions for upcoming matches. The app uses statistical analysis to predict the winner of the match, the player of the match, and other outcomes. The dataset used by Cricket Mazza is collected from various sources, including player and team statistics, historical match data, and expert analysis. The app evaluates the usability of the product by providing users with an easy-to-use interface and real-time updates during the match.

- **Dream11**: Dream11 is a fantasy cricket app that allows users to create their own virtual teams and compete with other users. The app uses statistical analysis to predict the performance of players in upcoming matches, allowing users to make informed decisions when creating their teams. The dataset used by Dream11 is collected from various sources, including player and team statistics, historical match data, and expert analysis. The app evaluates the usability of the product by providing users with an easy-to-use interface and real-time updates during the match.

Index	App Name	Methodology	Dataset Used	Evaluation	Limitations
1	Cricadium	Statistical Analysis + Machine Learning	Player and team statistics, expert analysis	Usability evaluation, real-time updates	Poor User Interface, confusing
2	Cricket Mazza	Statistical Analysis	Player and team statistics, historical match data, expert analysis	Usability, real-time updates	User engagement, Slow connectivity
3	Dream11	Statistical Analysis	Player and team statistics, historical match data, expert analysis	Usability, real-time updates	Does not include expert commentary, inaccurate predictions

Table 1.1: Analysis of existing solutions in the market.

2. System

2.1 System Architecture

The database servers supply data to the processing servers, where machine learning algorithms are used to process the data. Initially, EDA is performed where we inspect the data to identify any patterns or anomalies. Secondly, we perform PCA to filter out the most corelated variables. This helps us in choosing the best features we could use for predictions. Once we have selected the most suitable model, we deploy it to start serving results. The deployment process involves integrating the model with the existing system and ensuring that it works seamlessly with the data processing servers. We also perform thorough testing to ensure that the model is accurate and reliable. The final step is to continuously monitor the model's performance and make any necessary adjustments. This includes monitoring the data quality, identifying any drift in the data, and retraining the model as needed.

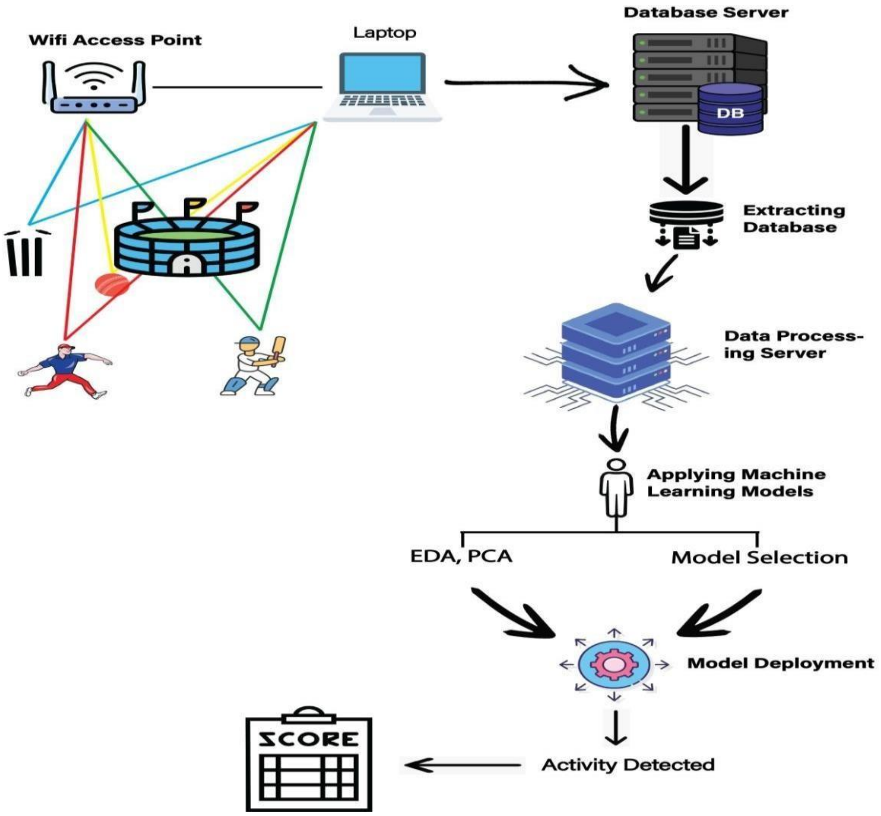


Figure 1.1

Feature within the application	**Associated Design Principle**	**Significance of the feature**
Selection of values instead putting them in	Consistency, constraints	Gives the user a selection of options to choose from instead of them having to put the data in
Live API	Visibility, Consistency	Providing real-time scoring, detailed match statistics, and a variety of additional information for sports enthusiasts
Information about Top Players	Visibility	Crucial for understanding and improving the performance of players.
Information about Teams	Visibility	Assist the user in knowing more about the different teams and their players
Navigation bar instead of using a hamburger menu	Visibility, Affordance	Makes the application straightforward to browse

Description:

- **Selection of values instead of putting them in** - Instead of manually inputting data into a system, this feature lets users select from a predefined list of possibilities. When filling out forms this tool may considerably enhance efficiency and accuracy. By restricting the alternatives to a narrow set, the risk of human mistakes, such as typos or erroneous formatting, is eliminated. Additionally, it saves time by reducing the need for manual input and can improve data analysis by ensuring consistency in the selected values. Overall, the feature of the selection of values streamlines data entry and improves the quality of data collected.
- **Constraints** - The feature of choosing values rather than entering them is connected to the design fundamental of constraints since it restricts the

available possibilities to those that are valid and practicable within the current context. By limiting the accessible alternatives, designers may guarantee that consumers select choices that match the design's intended goal, resulting in an enhanced user experience. Furthermore, limitations can help to simplify the design process by lowering the number of alternative configurations and allowing designers to concentrate on improving a smaller set of possibilities. This method can increase the design's usability and efficiency, resulting in a more effective product or system.

- **Consistency** - By fostering consistency across the system, the attribute of selecting values rather than entering them directly connects to the design concept of consistency. When consumers are given a consistent manner of picking values, the possibility of mistakes and confusion is reduced.

  We have also made sure that the options up for selection throughout the application are consistent and that the order of flow of usage is consistent. Consistency guarantees that the system's architecture is foreseeable and that consumers can rely on it to act predictably. The user interaction becomes more simplified and efficient by providing a uniform way to choose values. Overall, consistency is crucial to a good user interface design since it helps the user create confidence and reliability.

- **Live API** - The user can enjoy the ongoing IPL match and make predictions on the player's performance simultaneously with real-time updates provided by the application.

- **Visibility** - The live API should provide real-time updates on the match, including scores, player statistics, and other relevant information. This information should be easily accessible to the user, with clear and concise visualizations that allow for quick and accurate predictions. By prioritizing visibility in the design process, the user can make informed decisions based on the most up-to-date information available. Additionally, a live API can enhance the user experience by providing a more engaging and interactive platform. Overall, integrating a live API in an IPL prediction application can improve accuracy and increase user engagement.

- **Consistency** - The live API is quite similar in appearance to many popular platforms such as Cricbuzz, cricket scores/stats shown online and the way a live match score is presented on screens. Consistency in design is crucial for a seamless user experience. The live API's similarity in appearance to popular sports platforms fosters familiarity and reduces confusion. This familiarity, in turn, helps to reduce any potential confusion or mistakes that may occur while using the application, which builds trust and credibility with users. Consistency also improves the overall efficiency of the application by reducing potential mistakes. The design concept of consistency is critical for the live API to provide users with quick and efficient access to information, building trust and credibility over time.

- **Information about top players -** The front page of the programme offers a carousel of photographs of the best players; clicking on one of the images takes the user to the player's page, which gives information about the player such as the number of games played, total runs scored, strike rate, balls faced, and highest score. This is provided to assist people who are inexperienced with the event in better comprehending it.

- **Visibility** - The function of displaying prominent players' statistics in an IPL predictions app is an illustration of the design concept of visibility. The concept behind this approach is that significant information should be clearly visible and stand out from other items on the screen. The data about the best players in an IPL prediction software is crucial for creating good predictions. The top players' information has been presented prominently on the display, making it easier for people to locate. Users may then easily discover the information they want and make accurate predictions.

  Overall, the function of displaying information about the best players throughout an IPL prediction application effectively exhibits the visibility design concept. By making critical information highly visible, the app enhances the user experience and helps users make informed predictions.

- **Information about teams -** The application's main page has a carousel of team logos; clicking on one of the logos takes the user to the team's page, which provides information about the team, such as the identity of the coach, the team's history, and the names of the players. This is provided to assist people who are inexperienced with the event in better comprehending it.

- **Visibility** - The visibility design concept emphasizes the necessity of making information and actions apparent to users. This idea is being utilised by displaying information about teams since it makes the information about the teams appear to users. This tool enables users to quickly access information about the teams in which they are interested, making it easier for them to explore and discover what they are searching for. This feature provides consistency and makes it easier for users to comprehend and engage with the system by utilising recognisable symbols, colours, and menu hierarchy.

An implementation of the design principle of visibility is the feature of displaying team information in an IPL prediction app. Users may effortlessly and quickly learn the information they need to make accurate predictions by showing information about teams prominently on the app screen. To successfully apply the visibility design concept, team information can be shown in a prominent area on the screen, such as at the top of the app's main page. The team names and logos can be shown in high-contrast colours and huge fonts to focus the user's attention to these teams.

Users may also make educated predictions about each team's performance by providing vital statistics and information about each squad, such as their win-loss record or current league ranking. This information may be visually highlighted by employing bigger typefaces, strong writing, or colour contrast, allowing users to rapidly identify the critical information they require. entire, the feature of displaying team information is a good use of the design concept of visibility, enhancing the entire user experience of an IPL prediction software.

- **Using a navigation bar instead of a hamburger menu** - It is a user interface component that provides buttons or links to other menus or pages on the application. It includes features such as drop-down menus, submenus, and other navigational aids such as search boxes and icons.

- **Visibility** - The use of a navigation bar instead of a hamburger menu shows the design principle of visibility. A navigation bar is an element that is constantly displayed on the screen, making it easier for users to locate and access the site's various sections. It gives a clear idea of the site's structure and available options which helps users to understand where to navigate. On the other hand, a hamburger menu is often hidden and requires an extra step for users to access it, making it less visible and confusing. Thus, by using a navigation bar instead of a hamburger menu, designers give priority to visibility. The are making sure that users have a clear understanding of the site's layout and available options.

- **Affordance** - The design principle of affordance is demonstrated by the feature of a navigation bar rather than a hamburger menu because it gives the user a clear and obvious indicator of the activities that are possible. A design idea known as affordability refers to an object's instinctive and obvious characteristics that imply how it should be used. In this instance, the user can engage with a variety of alternatives that are obvious and simple to identify thanks to the navigation bar. The navigation bar's functionality is obvious because it is made to resemble a row of clickable buttons. The user will easily comprehend how to use the navigation options and get what they're seeking for thanks to this design choice.

2.2 Technology Stack:

1. **Figma:** Figma is a browser-based design and prototyping tool. We designed our low fidelity prototype on Figma. It gave our team the opportunity to work collectively and refine the design of the app before development even started. Creating a prototype made available a visual representation of our application's practicality, layout and user interface which gave us the opportunity to test and ratify the design and make any required alterations. Figma played a very important role in the app design process.

2. **Flutter:** Flutter is an open-source UI software development kit (SDK) created by Google. It allowed us to build a high-quality mobile based application. Dart programming language allowed us steady development and easy maintenance. We were provided with a wide range of pre-built widgets and tools that enabled us to create visually attractive and alterable user interfaces.

3. **Python:** We developed the backend of the app using Python. Python is a very adaptable coding language which provides wide-range of libraries and frameworks. With the help of python, we were able to develop an efficient and saleable backend. With python's strong assimilation and capability to handle large amounts of data made it an ideal choice for the backend of the app.

4. **ML Models**: Machine learning algorithms are approaches that may be used to create develop prediction-based apps. They use mathematical algorithms and pattern recognition to enable computers to take decisions and make predictions on their own. They can server in many ways including DIP, NLP, and prediction tasks. To make our predictions, we have used the Linear Regression model. Linear regression is a method for predicting the value of one variable based on the value of another. In linear regression, one variable is dependent on the other variable.

5. **Flask:** Flask is a web framework written in Python that is used to create web applications. It is a micro-framework, which implies that it just provides just the necessary features required to develop an application, such as routing, request processing, and response generation. It enabled us to tailor our application to our specific requirements. Flask was used to link the backend with flutter. The backend oversees managing front-end requests and answers. When a user interacts with an app, the front end sends a request to the backend, which processes it and returns a response. When the backend has processed the request and generated a response, it sends it to the frontend, which then updates the user.

6. **Cloud:** Machine learning algorithms are approaches that may be used to create develop prediction-based apps. They use mathematical algorithms and pattern recognition to enable computers to take decisions and make predictions on their own. They can server in many ways including DIP, NLP, and prediction tasks. To make our predictions, we have used the Linear Regression model. Linear regression is a method for predicting the value of one variable based on the value of another. In linear regression, one variable is dependent on the other variable.

3. Methodology

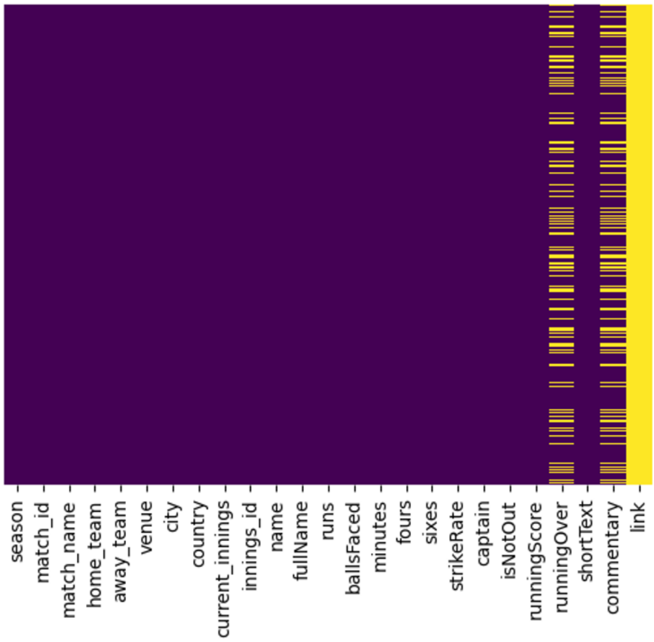
3.1 ML: CRISP-DM

3.1.1 EDA (Exploratory Data Analysis)

Batsman

```
        import pandas as pd
import numpy as np import seaborn as sns # Load the dataset df =
pd.read_csv("C:/Users/ridit/Downloads/all_season_batting_card.csv")
```

Graph showing the null values in the dataset



Dropping features that we don't need

```
df.drop("commentary",axis=1,inplace=True)
df.drop("link",axis=1,inplace=True)
```

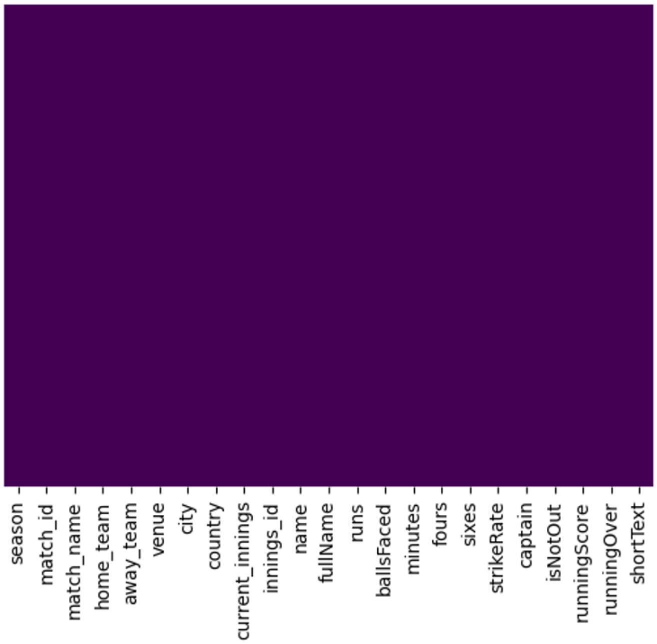
Here, we are replacing the null values with mean

```
df['runningOver'].fillna(int(df['runningOver'].mean()),inplace=True)
```

Graph representing that there are no null values present now

```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<AxesSubplot:>
```



```
df['runs'].isna().sum() np.where(df['runs'].isnull() ==
True)[0] array([ 3038, 7642, 9490, 11942, 12985, 13068],
dtype=int64) df['ballsFaced'].isna().sum()
np.where(df['ballsFaced'].isnull() == True)[0] array([ 3038,
7642, 9490, 11942, 12985, 13068], dtype=int64)
df['fours'].isna().sum() np.where(df['fours'].isnull() == True)[0]
array([ 3038, 7642, 9490, 11942, 12985, 13068], dtype=int64)
df['sixes'].isna().sum()
np.where(df['sixes'].isnull() == True)[0] array([ 3038, 7642,
9490, 11942, 12985, 13068], dtype=int64) df['strikeRate'].isna().sum()
np.where(df['strikeRate'].isnull() == True)[0] array([ 3038, 7642,
9490, 11942, 12985, 13068], dtype=int64) df.drop([3038, 7642, 9490,
11942, 12985, 13068],axis=0,inplace=True) df['minutes'].isna().sum()
np.where(df['minutes'].isnull() == True)[0] array([], dtype=int64)
df["captain"] = df["captain"].astype(int) df["isNotOut"] =
df["isNotOut"].astype(int)
```

Different data types

```
df.dtypes
season      int64 match_id
   int64 match_name
   object home_team
   object away_team
   object venue     object
   city object country object
current_innings          object
innings_id          int64
name    object fullName object
runs float64
ballsFaced          float64
minutes object fours float64
sixes float64 strikeRate
   object captain int32
isNotOut int32
runningScore        object
runningOver float64 shortText
object dtype:
object
```

Replacing a negative value with 0

```
df = df.replace(['-'], 0)
```

Conversion from float to integer

```
df['runs'] = df['runs'].astype(int)

df['ballsFaced'] = df['ballsFaced'].astype(int)

df['fours'] = df['fours'].astype(int) df['sixes']

= df['sixes'].astype(int)



df1=df[['runs','ballsFaced']]
```

Bowler

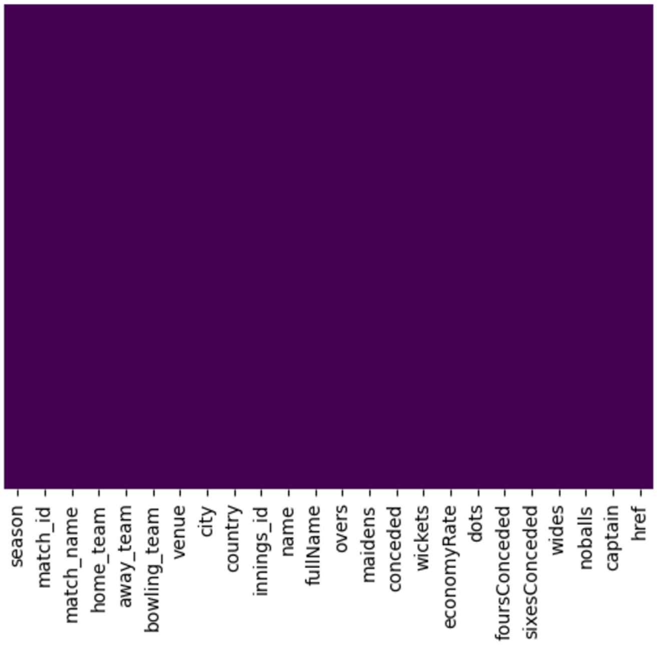
```
import pandas as pd
import numpy as np
import   seaborn   as
sns
```

```
df = pd.read_csv("C:/Users/ridit/Downloads/all_season_bowling_card.csv")
```

Graph showing the null values in the dataset

```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<AxesSubplot:>
```



Dropping features that we don’t need

```
df.drop("country",axis=1,inplace=True)
df.drop("city",axis=1,inplace=True) df.drop("name",axis=1,inplace=True)
df.drop("venue",axis=1,inplace=True)
```

```
df['overs'].isna().sum()

np.where(df['overs'].isnull() == True)[0]

array([], dtype=int64) df['maidens'].isna().sum()

np.where(df['maidens'].isnull() == True)[0]

array([], dtype=int64)

df['conceded'].isna().sum()

np.where(df['conceded'].isnull() == True)[0]

array([], dtype=int64)

df['innings_id'].isna().sum()
np.where(df['innings_id'].isnull() == True)[0] array([],
dtype=int64)
```

```
df['wickets'].isna().sum()

np.where(df['wickets'].isnull() == True)[0]

array([], dtype=int64)

df['economyRate'].isna().sum()

np.where(df['economyRate'].isnull() == True)[0]

array([], dtype=int64) df['dots'].isna().sum()

np.where(df['dots'].isnull() == True)[0] array([],

dtype=int64) df['foursConceded'].isna().sum()

np.where(df['foursConceded'].isnull() == True)[0]

array([], dtype=int64)

df['sixesConceded'].isna().sum()

np.where(df['sixesConceded'].isnull() == True)[0]

array([], dtype=int64) df['wides'].isna().sum()

np.where(df['wides'].isnull() == True)[0] array([],

dtype=int64) df['noballs'].isna().sum()

np.where(df['noballs'].isnull() == True)[0]

array([], dtype=int64)
```

Different data types

```
df.dtypes
```

```
season      int64 match_id
   int64 match_name
   object home_team
   object away_team
   object bowling_team
   object innings_id
   int64 fullName object
overs float64 maidens int64
conceded int64 wickets int64
   economyRate object
dots              int64
foursConceded int64 sixesConceded
int64
wides  int64  noballs
int64  captain  int32
href  object  dtype:
object
```

Replacing a negative value with 0

```
df = df.replace(['-'], 0) # df['runs']
= df['runs'].astype(int)
```

Conversion from float to integer

```
df["captain"] = df["captain"].astype(int)
```

3.1.2 PCA (Principal Component Analysis)

Batsman

```
from scipy.stats import pearsonr
corr1,_=pearsonr(df['runs'],df['ballsFaced'])
corr2,_=pearsonr(df['runs'],df['fours']) corr3,_=pearsonr(df['runs'],df['sixes'])
corr4,_=pearsonr(df['runs'],df['strikeRate'])
corr5,_=pearsonr(df['runs'],df['minutes'])
corr6,_=pearsonr(df['runs'],df['captain'])
corr7,_=pearsonr(df['runs'],df['isNotOut'])
corr8,_=pearsonr(df['runs'],df['runningOver'])

print(corr1) print(corr2)
print(corr3) print(corr4)
print(corr5) print(corr6)
print(corr7) print(corr8)
```

```
0.9315877805853926
0.8641393500176645
0.757760267146123
0.503465259316379
0.5885765095899526
0.12920257200382604
0.0628862828717322 0.19376809259357786 df['opponentteam'] = np.nan
```

```
df.loc[df['current_innings'] == df['away_team'], 'opponentteam'] = df['home_team']

df.loc[df['current_innings'] == df['home_team'], 'opponentteam'] = df['away_team']

df1.drop(df.columns[[2,3,5,6,7,8,9,10]], axis=1, inplace=True)
```

Bowler

```
from scipy.stats import pearsonr
corr1,_=pearsonr(df['conceded'],df['overs'])
corr2,_=pearsonr(df['conceded'],df['maidens'])
corr3,_=pearsonr(df['conceded'],df['wickets'])
corr5,_=pearsonr(df['conceded'],df['economyRate'])
corr6,_=pearsonr(df['conceded'],df['dots'])
corr7,_=pearsonr(df['conceded'],df['foursConceded'])
corr8,_=pearsonr(df['conceded'],df['sixesConceded'])
corr9,_=pearsonr(df['conceded'],df['wides'])
corr10,_=pearsonr(df['conceded'],df['noballs'])

print(corr1) print(corr2)
print(corr3) print(corr5)
print(corr6) print(corr8)
print(corr7) print(corr9)
print(corr10)
0.593037778502806
-0.09139672493748126
0.031138667574488514
0.49196265458652966
0.08998784561370153
0.624253459233199
0.641321316221021
0.26355211400679246 0.1667386014056911 df=

df.drop(['match_name','innings_id','economyRate','dots','wides','noballs','captain','href','maidens'

], axis=1)



df['overs']=df['overs']*6 df["overs"]

= df["overs"].astype(int) df['opponentteam'] = np.nan df.loc[df['bowling_team']

== df['away_team'], 'opponentteam'] = df['home_team']

df.loc[df['bowling_team'] == df['home_team'], 'opponentteam'] = df['away_team']
df= df.drop(['home_team','away_team','bowling_team'], axis=1)
```

3.1.3 Business Understanding

Business Understanding in CRISP-DM for IPLitics involves understanding the objectives, requirements, and constraints of the project. The main objective of IPLitics is to provide cricket enthusiasts with a reliable and accurate prediction app for the Indian Premier League. The app's primary function is to provide precise forecasts for upcoming matches based on recent team and player performances, using advanced algorithms and machine learning techniques.

To achieve this objective, IPLitics considers various factors such as strike rate, runs conceded, balls faced, wickets taken, and more to make predictions. The app also provides comprehensive statistics on recent form, past records against the opponent, and other significant metrics that could impact the match outcome. Additionally, the app features a live score tracker to keep users updated on the match's progress.

The stakeholders involved in IPLitics includes cricket fans, betting companies, cricket analysts and commentators, and the IPL itself. The app must provide accurate predictions as it plays an essential role in place bets. It must also provide valuable insights and predictions for cricket analysts and commentators to improve their accuracy and credibility.

The constraints of IPLitics include data availability, data quality, and time constraints. The app must rely on data sources that are reliable and up-to-date to provide accurate predictions. It must also ensure that the data used is of high quality to avoid inaccurate predictions. Finally, IPLitics must operate within the time constraints of the IPL season to provide users with timely and relevant information.

3.2 HCI: Experiment Design

We conducted usability evaluation by the method of discount usability to get an idea of the usability issues in the user interface of the prototype.

3.2.1 Learnability -
The term "learnability" describes how simple a system is to learn to use. It is common knowledge that individuals dislike taking a lot of time to learn how to utilize a system. They want to familiarise themselves with the application as quickly as possible. Therefore, it is vital that consumers spend as little time as possible learning the functionality.

- To make the programme easier to learn, we've incorporated a navigation bar. The navbar can aid in learnability by giving users a simple and dependable way to move about an application. When users access the application, they frequently want to locate the data or pages they require fast, and the navbar can offer them a simple navigation method. The most crucial pages of an application, such as the home page, live page, and other features, are often linked to the navbar. Users may access the information they need quickly and simply by posting these links in a visible area at the top of the page.

- Additionally, users may learn the application's layout and where to access particular pages or content if the navbar is consistently placed throughout the entire application. Users rapidly pick up on the fact that the navigation bar is constantly at the bottom of the page and that certain sites always appear in the same spot within the navbar.

- Overall, by offering a simple and consistent navigation system that makes it simple for visitors to discover the information they need and become familiar with the application's structure, the navbar can aid in learnability.

- To aid with learnability, we have also ensured that the application's overall layout is consistent. By ensuring consistency and predictability in design, layout, and interaction, uniformity promotes learnability. It eliminates the need for exploration and trial-and-error by enabling consumers to quickly learn and comprehend how to use an application or programme.

- Additionally, we have made sure that the user can choose an option rather than entering data anywhere we need it. We can improve the learnability of the material by choosing specific values to include rather than providing learners with an abundance of possibilities. We can lessen the cognitive load and make it simpler for learners to assimilate and retain the information by carefully curating the content and giving priority to the most crucial and pertinent information. Due to the more focused and meaningful presentation of the principles being taught, this can also aid students in understanding their relevance and context. Ultimately, by being deliberate in our selection of values, we can improve the efficacy and efficiency of the learning process.

We used discount usability to evaluate usability. We asked a group of 10 people to rate the learnability of the application out of 5, the average rating was 4.1.

3.2.2 Efficiency -

Efficiency is the ability of a user interface to enable quick and simple task completion. Users may execute jobs with the least amount of effort and in the least amount of time thanks to an effective interface.

- Users can learn and recall tasks more quickly if an interface's layout, design, and functionality are all consistent. Users may find it easier to understand the application's structure and where to find particular pages or content if the navbar is consistently placed throughout the entire application. Users rapidly pick up on the fact that the navigation bar is constantly at the bottom of the page and that certain sites always appear in the same spot within the navbar.

- Giving users keyboard shortcuts or other expedient workarounds can save waiting time and boost productivity. The navigation bar we've given users allows them to quickly and simply access the features they want to use in our programme.

- Choosing values rather than typing them in can help with efficiency in a variety of ways. For starters, by picking only the relevant information, the quantity of data that must be processed is minimised, thereby saving time and resources. Second, choosing values can aid in the elimination of redundancies and inconsistencies in data, which can increase accuracy and prevent mistakes. Third, choosing values can aid in prioritising the most significant information, making it simpler to see patterns and trends. Fourth, by minimising the amount of noise in the data, picking values can assist to enhance the effectiveness of algorithms and models. Overall, value selection can result in greater effectiveness and efficiency in data processing and analysis.

To assess usability, we used discount usability. We asked a group of 10 people to rank the application's effectiveness on a scale of 1 to 5, and the mean score was 4.08.

3.2.3 Memorability -

- The term "memorability" describes how simple it is to recall how to utilise a product once you have it. This is crucial for interactive items that aren't utilised very often. Users should be able to quickly be reminded of how to use an operation if they haven't used it in a few months or longer. Users shouldn't always need to learn how to complete tasks.

- The navigation bar is an essential feature in an app design that helps users to navigate through different sections or pages of a digital product. This feature enhances the design principle of memorability by making it easier for users to remember how to move around the product. With a consistent and well-designed navigation bar, users can easily recall the layout and structure of the product, making it easier for them to find what they need and to return to important sections or pages. Additionally, a navigation bar that is easy to use and understand also makes the product more memorable and enjoyable to use, creating a positive user experience that encourages return visits.

- The feature of selecting options instead of inputting text can help with the design principle of memorability by reducing the cognitive load required to recall information. When users are presented with a set of predetermined options, they are more likely to remember their choices compared to when they have to recall information from memory or type it out themselves. This is because selecting from a list of options is easier than generating information from scratch, and it can also help users quickly associate their choices with the available options. As a result, designing interfaces that allow users to select options instead of inputting text can help improve the memorability of the user experience, leading to more efficient and effective interactions.

- We have also made sure that the options up for selection throughout the application are consistent and that the order of flow of usage is consistent which helps with memorability.

To assess usability, discount usability was used. The application was used by a group of people, who were then invited to use it again after a week had passed. The participants from that group were then given the task of rating the application's memorability on a scale of 1 to 5, with the mean result being 4.04..

3.2.4 Error Protection -

Error protection involves defending the user against hazardous circumstances and uncomfortable circumstances. It refers to assisting any type of user in any situation to avoid the risks of unintentionally performing undesired activities. It also alludes to the alleged anxieties users may have regarding the results of making mistakes and how this influences their behaviour. Preventing users from making critical mistakes is a key component of making interactive products safer.

- There are various ways in which selecting values as opposed to just entering them might aid with error protection. First off, it can prohibit the entry of false or inaccurate data into a system or application by carefully selecting the values that are permitted. This can aid in ensuring that the result is exact and correct. Second, choosing values can aid in avoiding security flaws like buffer overflows or other potential attack vectors. The system can be made more secure and less susceptible to assaults by restricting the possible range of values. Last but not least, choosing values can help to ensure that the system is more dependable and robust because it lowers the likelihood that faults and errors will emerge due to improper or unexpected input.

We utilised discount usability to rate usability. On a scale of 1 to 5, we asked a group of 10 people to rate the application's error prevention; the mean result was 3.97.

3.2.5 Satisfaction -

Human-computer interaction (HCI) relies heavily on user pleasure. It relates to the consumer's perception of the computer system that they utilise in their workplace. Satisfaction among users is an indication of how well a system satisfies the requirements and expectations of the user. It is frequently examined through questionnaires and surveys that gauge the user's perception of a particular computer programme or system. The University of Maryland's Human-Computer Interaction Laboratory (HCIL) has created a questionnaire for user interaction satisfaction (QUIS) that includes a demographic questionnaire as well as a satisfaction questionnaire. User satisfaction is inextricably linked to the idea of user experience (UX), which is also frequently debated in the HCI community. User experience (UX) is a larger notion that includes the user's emotions, attitudes, and perceptions of the system, whereas user satisfaction is a more precise assessment of the user's happiness with the system.

User Satisfaction can be achieved in various ways:-

Involving users in the design phase is one technique to guarantee that the system satisfies their requirements and expectations.

Another approach is to give users with clear and simple instructions and feedback to assist them in understanding how to utilise the system efficiently. User happiness may be increased by providing a user-friendly interface that is simple to browse and comprehend. Conducting usability testing and getting user input may also aid in identifying areas for improvement and increasing user happiness.

Standard questionnaires and surveys can also be useful in identifying areas for improvement and monitoring the impact of design modifications. Overall, increasing user happiness necessitates a user-centred strategy that focuses on satisfying the consumers' needs and expectations.

To assess satisfaction we conducted several surveys and asked the respondents to rate their satisfaction with the application on a scale of 1-5, and received an average rating of 4.1.

3.2.6 Utility -

Utility in HCI (Human-Computer Interaction) refers to the degree to which a computer system or interface is perceived to be useful and usable by its users. Utility is an essential factor in the design of user interfaces, as it affects the user's satisfaction and productivity while using the system.

- The use of Live APIs in an IPL prediction application reflects the design principle of utility by providing users with real-time and relevant data that is essential for making informed predictions. By integrating Live APIs into the application, users can receive up-to-date information about team standings, player stats, and match schedules. This information serves as a utility to the user, as it assists in making accurate predictions, ultimately enhancing their overall experience. The principle of utility emphasises the importance of creating products that provide value and usefulness to the user. In the case of an IPL prediction application, the use of Live APIs is a prime example of this principle in action

- The design principle of utility refers to the degree to which a product or application is useful and efficient for achieving its intended purpose. Selecting options instead of inputting them in an IPL prediction application is a design feature that exemplifies this principle. By providing users with predetermined options, the application streamlines the prediction process, making it faster and more user-friendly. Users can easily select from a set of options rather than having to type out their predictions manually. This design feature increases the overall utility of the application, as it allows users to make predictions quickly and easily, ultimately enhancing their overall experience. Overall, selecting options over manual input is a design choice that reflects the importance of utility and usability in creating effective products and applications.

We utilised discount usability to rate usability. On a scale of 1 to 5, we asked a group of 10 people to rate the application's utility; the mean result was 4.18.

3.3 MAD (Mobile Application Development)

We have used plenty of features for the development of our app like:

1: Carousel_pro: It is a Flutter package that provides a carousel slider widget that supports both network and asset images. can modify the UI according to your design easily. It is a customizable carousel slider widget in Flutter which supports infinite scrolling, autoscrolling, custom child widget, custom animations and pre-built indicators. We have used it to make the slide show on the homepage.

2: Url_launcher: Url Launcher is a Flutter plugin that allows you to launch any URL (Uniform Resource Locator) in the user's default browser, email or phone app from your Flutter app. Seamlessly integrate external content such as websites, phone numbers, email addresses and maps into your app. Url Launcher also allows you to handle the response from the launched URL. B. Recognize if the user closed the browser or successfully sent the email. Url Launcher is a must-have plugin for apps that require the integration of external content, improving the user experience by enabling seamless navigation between your app and other apps. We have used it in our Terms & Conditions page which when clicked redirects us to a pdf.

3: Stateful_widget: A StatefulWidget is a Flutter widget that allows you to create dynamic and interactive UI components that can change state over time. It is used when depending on user actions or other events he needs to update the UI.

4: Bottom_navigation_bar: BottomNavigationBar is a Flutter widget that displays a series of buttons or tabs at the bottom of the screen. We typically use it to provide navigation between different views or screens within our app. Each button in the

BottomNavigationBar is represented by a BottomNavigationBarItem widget that contains an icon and label. We can customise the BottomNavigationBar by setting its appearance, such as background colour and selected item colour. Additionally, we can handle the user tapping an item by providing a callback function that executes when the item is selected.

5: Future: By using the ‘Future’ feature, we can represent an asynchronous operation that may complete with a value or with an error. We use Future to perform operations that may take some time to complete, such as network requests, file I/O, or any other long-running task.

6: Async and Await: To make a function asynchronous, use the async keyword. Even if the function has not completed its execution, an async function returns a Future object immediately. This allows the caller code to continue executing without having to wait for the async function to finish. The await keyword instructs Dart to wait for a Future to complete before proceeding with the next command. When the await keyword is met, the function comes to a standstill until the expected Future completes. When the Future is finished, the execution resumes. Async and await, when used together, allow us to design code that looks to operate synchronously but actually executes asynchronously beneath the hood.

7: ListView_builder: ListView.builder is a Flutter widget that produces a scrollable list of things quickly. It is utilised when there are a big or dynamic number of objects that cannot all fit on the screen at the same time. The ListView.builder only generates the items that are now visible on the screen, and the remainder is created lazily as the user scrolls. You must supply the itemBuilder function, which is used to generate each item in the list. The BuildContext and the index of the item to be constructed are provided to the itemBuilder function. You can use the index to acquire the data for that item and make a widget out of it. We have used this widget to create lists for selecting the teams and players.

8: Cupertino: Cupertino is a Flutter design language that mimics the look of Apple's iOS. Cupertino widgets use iOS-style design and give user interface features familiar to iOS users, such as buttons, selectors, and sliders. Cupertino widgets ensure a uniform appearance and feel across iOS devices, which is critical for delivering a unified user experience. Cupertino widgets are straightforward to use and connect with Flutter widgets, having a clean and basic design that is simple to grasp. We have used this to design the ball by ball analysis selection.

9: **Http_overrides**: It is a Flutter framework feature that allows us to override the HTTP client's default actions. With httpOverrides, we can modify the HTTP client's behaviour for specific URLs, headers, or response handling. httpOverrides can be used to add custom headers to HTTP requests, adjust response processing, or even completely replace the HTTP client implementation. The httpOverrides property is useful for developers who need to fine-tune their HTTP requests and replies for certain contexts, or who want to interact with third-party services that require special headers or behaviour. Overall, httpOverrides is a strong utility for increasing the flexibility and customisation of Flutter's HTTP client. We have used this framework in development of live API.

10.**Animation_controller**: Animation_controller class lets you control animation. It lets the user decide the length and direction of an animation and control its present value.

This class has a value that defines the animation’s ongoing position, which can further be used to control the appearance of the animation. It can also be used to add listeners that acknowledge the changes in the animation’s state, similarly when it starts or ends. This class is necessary for developing dynamic and associated animations in Flutter, and it delivers an untroubled and precise path to process animations. We have used it in the more button in the navigation bar.

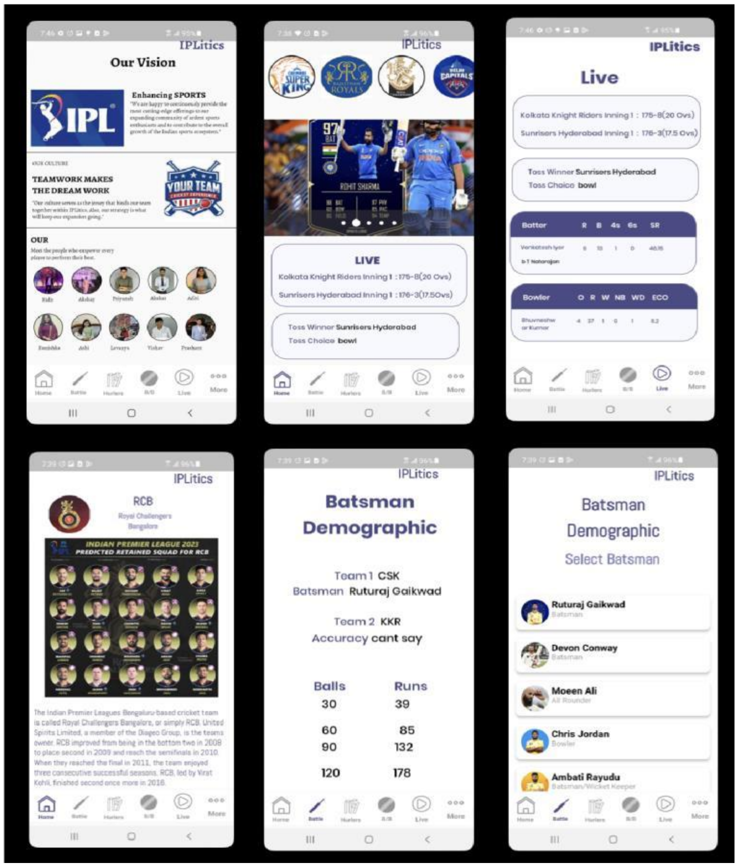
4. Results

Usability Evaluation

Evaluation Criteria	**Rating (0-5)**
Learnability	4.1
Efficiency	4.08
Memorability	4.04
Error Protection	3.97
Satisfaction	4
Utility	4.18

We gave 75 individuals access to our app so they could use it, and we gave them a form to review it. They were asked to rate the app on a scale of 1 to 5, with 1 denoting that it does not at all meet the usability evaluation criteria and 5 denoting that it does so entirely. After a few days, we offered the same individuals another chance to use the application before allowing them to modify their memorability rating as they saw fit.

4.1 User Interface Screenshots:



IPLitics

Our Vision

Enhancing SPORTS

"We are happy to continuously provide the most cutting-edge offerings to our expanding community of ardent sports enthusiasts and to contribute to the overall growth of the Indian sports ecosystem."

OUR CULTURE

TEAMWORK MAKES THE DREAM WORK

"Our culture serves as the jersey that binds our team together within IPLitics. also, our strategy is what will keep our expansion going."

OUR

Meet the people who empower every player to perform their best.

Rishi, Akshay, Priyansh, Alankar, Aditi, Ranishka, Ashi, Lavanya, Vishav, Prashant

IPLitics

LIVE

Kolkata Knight Riders Inning 1 : 175-8(20 Ovs)

Sunrisers Hyderabad Inning 1 : 176-3(17.5Ovs)

Toss Winner Sunrisers Hyderabad

Toss Choice bowl

IPLitics

Live

Kolkata Knight Riders Inning 1 : 175-8(20 Ovs)

Sunrisers Hyderabad Inning 1 : 176-3(17.5 Ovs)

Toss Winner Sunrisers Hyderabad

Toss Choice bowl

Batter	R	B	4s	6s	SR
Venkatesh Iyer b T Natarajan	6	13	1	0	46.15

Bowler	O	R	W	NB	WD	ECO
Bhuvneshwar Kumar	4	37	1	0	1	8.2

IPLitics

RCB

Royal Challengers Bangalore

INDIAN PREMIER LEAGUE 2023 PREDICTED RETAINED SQUAD FOR RCB

The Indian Premier Leagues Bengaluru-based cricket team is called Royal Challengers Bangalore, or simply RCB. United Spirits Limited, a member of the Diageo Group, is the teams owner. RCB improved from being in the bottom two in 2008 to place second in 2009 and reach the semifinals in 2010. When they reached the final in 2011, the team enjoyed three consecutive successful seasons. RCB, led by Virat Kohli, finished second once more in 2016.

IPLitics

Batsman Demographic

Team 1 CSK

Batsman Ruturaj Gaikwad

Team 2 KKR

Accuracy cant say

Balls	Runs
30	39
60	85
90	132
120	178

IPLitics

Batsman Demographic

Select Batsman

Ruturaj Gaikwad	Batsman
Devon Conway	Batsman
Moeen Ali	All Rounder
Chris Jordan	Bowler
Ambati Rayudu	Batsman/Wicket Keeper

4.2 ML Results

Batsman Demographic Model

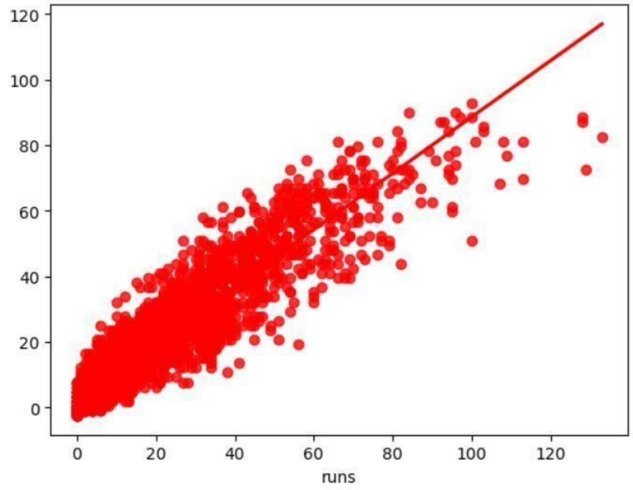
Accuracy:

KNN: 0.9729499038806635

Linear Regression: 0.9806423475142928 Random

Forest: 0.9756855112132546

Linear Regression



Bowler Demographic

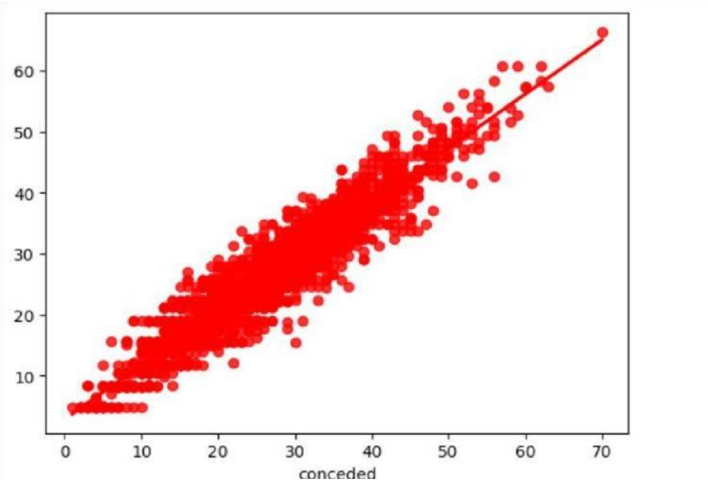
Model Accuracy:

KNN: 0.22417658824740522

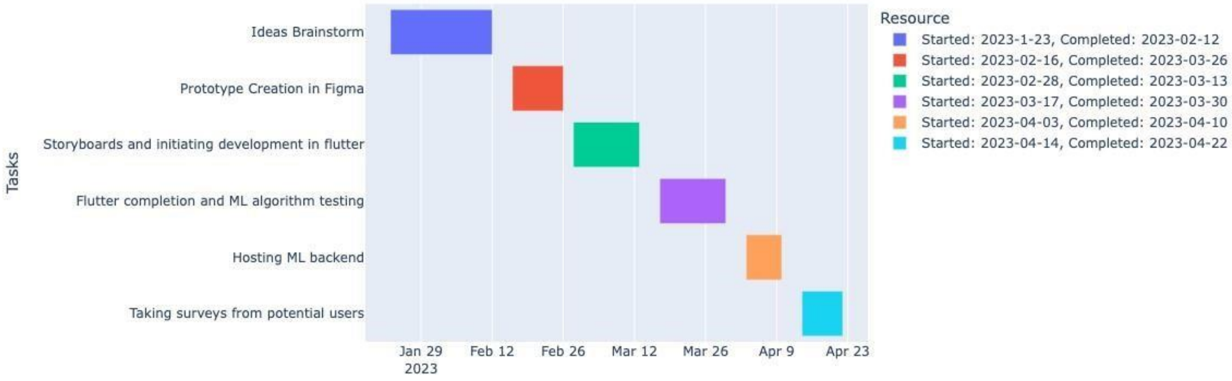
Linear Regression: 0.35066183323476474 Random

Forest: 0.36401675911304726

Linear Regression



5. Project Management



A Gantt chart was used to monitor progress over several months. The chart provided a clear visual representation of the project timeline and made it easy to track the completion of each task. First, we focused on brainstorming and researching what the first bar of the chart represented. Our brainstorming session started on January 23rd and ended on February 12th. We then proceeded to the next phase of the project. This involved creating low-quality prototypes in Figma. We ran this step between Feb 16th and Feb 26th and updated the Gantt chart accordingly. The Figma prototype helped develop a major application. The next step was to start storyboarding and Flutter development, which we worked on from February 28th to March 13th. After that, I focused on finalising Flutter development and testing all the machine learning algorithms. This phase lasted from March 17th to March 30th. From April 3rd to April 22nd, nearing the end of the project, we focused on hosting the application and conducting customer surveys. Ultimately, the Gantt chart served as a guide that kept the project on track and completed on time. This process taught me the value of careful planning and the importance of using tools like the Gantt chart to track progress.

How did we split up the project amongst ourselves? How were the tasks distributed?

After we went through ideation and came to the decision to make a cricket prediction application, we did a SWOT analysis of all the members of the team. We looked at that to gain a sense of their knowledge bases, strengths, and weaknesses. We initially assigned tasks based on each person's talents and weaknesses. But as time passed, we realised that if we carried on working in this manner, nobody would venture outside of their comfort zone and, as a result, would not acquire any new knowledge. So we decided to do reverse team ups for each task. We put people in small groups where each team had a task that suited one person's strengths and the weaknesses of the other. This resulted in the team learning from the experience and the work being completed in an efficient manner.

5.1 Peer Evaluation

Name	***Grade***	***Justification***
Ridit Jain	*10*	*SPOC of the group and actively managed and lead the technical aspects of the project*
Lavanya Anand	*9.5*	*Essential team member of the low-fidelity prototype team and the content team.*
Akshay Kumar Jain	*9.35*	*A key member of the content team and a part of the design team*
Ashi Jain	*9.1*	*Worked in the designing and development of the UX/UI*
Priyansh Jain	*9.0*	*A vital team player on the content team who assisted with survey administration.*
Vishav Garg	*8.25*	*He played a significant role on the high-fidelity prototype team.*
Akshat Manohar	*8.2*	*He assisted with the project's coding and played a significant role in the API connectivity.*
Aditi Singh	*8.4*	*She helped in the UI/UX design*
Runishka Rao	*8.3*	*She participated in the UI/UX team and contributed to the creation of the low-quality prototype.*
Prashant Sharma	*7.7*	*He assisted in the execution of the surveys and worked as a member of the UI/UX team.*

6. Discussion Section

- **Learnings**

  We were rather dissatisfied at the beginning of our project owing to its size and complexity. It was a challenging task that took a team of about ten individuals to execute and comprised several components. However, as we worked on the project over the next few months, we came to appreciate the challenges it presented.

- **Challenges and Limitations**

  Unfamiliarity with cricket was one of the main issues we faced. But as a team, we were able to understand it in a better way which enabled us to start the project with more confidence. Lack of data was another challenge for us. We worked collectively during the whole project and learned from our team members skills and experiences. We were able to apply various concepts that we had learned by attending lectures, which helped us to gain a better understanding of these concepts in a more practical setting.

  Of course, as with any team, there were clashes and disagreements that arose. However, we learned the value of teamwork, mediation, and diplomacy. Initially, we assigned tasks based on each person's strengths and weaknesses. However, we soon realised that if we continued to work in this manner, we would never venture outside of our comfort zones and would not acquire any new knowledge.

  To overcome this challenge, we decided to implement a reverse team-up approach for each task. We formed small groups, where each team had a task that played to one person's strengths and the weaknesses of another. This approach allowed us to learn from each other's experiences and complete the project in an efficient manner.

- **Future**

Updating the dataset for a better result and procuring more information about the teams and existing players :

Our model makes predictions by using the old matches dataset. It provides feasible predictions for upcoming matches making us stay primed to make the right choices. Precision is crucial when it comes to predicting cricket scores. Since the data changes every year, we plan to update our dataset accordingly. This will allow us to gain more information about the players and teams. By doing so, we will be able to provide users with the most up-to-date information and make even more accurate predictions.

Adding more features like predicting the price of players in a future auction:

We aim to implement features such as player price prediction. This will let us choose the top players in the forthcoming auction. Being that the IPL is one of the most significant events in India, involving a feature that anticipates the selling price of players in a future auction could be highly useful. It could prove to be very beneficial for team owners from investment point of view.

Predicting the win/loss %

It may add features such as predicting player prices in future auctions. The Indian Premier League (IPL) is one of the highly significant events in the world of cricket, where players from all over the world are auctioned off for the highest bidder. Adding the capability to predict player prices at future auctions would be a huge benefit to cricket enthusiasts and team owners looking to invest in players. Use last year's game data and player performance to predict future value at auction. This feature helps users make informed decisions when bidding on players in auctions.

Predicting your perfect 11(team) for a particular game.

We also aim to add a feature that will allow people to foresee the most suitable 11 for a specific cricket match. Users can provide their preferences, such as favourite players, and the app will build their lineup of recommended teams based on this information. This function gives players vital information into the best potential squad composition, which may improve their chances of winning in fantasy cricket leagues or live matches.

In summary, while the project initially appeared daunting, we were able to work collaboratively as a team and overcome the challenges it presented. We learnt the significance of cooperation we gained deeper understanding of the concepts.

7. Abstract

IPLitics is an app designed to cherish the needs of cricket fans by giving them the to the point predictions for a variety of outcomes of cricket matches in the Indian Premier League (IPL). The app aims to deliver a reliable platform for cricket enthusiasts to stay updated on match progress and make wise decisions, especially when they are betting on their favourite teams.

The methods used by IPLitics are latest algorithms and statistical analysis which makes accurate predictions which depends on several factors like strike rate, run rate, balls faced, wickets and many more. It offers a wide range of mathematical arguments on present forms, previous records against the opponent and other important factors that could affect the match outcome. The platform also includes a live score tracker that updates the ongoing progress of the match.

To find out the significance of IPLitics for finding out the match outcomes, the results provide by the app were compared with actual match with multiple matches. The results were 93% accurate which shows that the app is capable of giving reliable predictions which is really beneficial for cricket Fans.

In conclusion, IPLitics is a valuable platform for cricket fans, offering them precise predictions using sophisticated algorithms and mathematical analysis. The app's live score tracker fetches real-time updates on match progress, its wide-ranging statistics on an array of metrics enables users to stay updated on the latest developments in the IPL and make educated decisions. The evaluation results validate the app's efficiency in predicting match outcomes, making it a consistent tool for users.

8. Bibliography

Chen, J. d. (2019). A comparison of linear regression, regularization, and machine learning algorithms to develop Europe-wide spatial models of fine particles and nitrogen d. 70-100.

Maulud, D. &. (2020). A review on linear regression comprehensive in machine learning. . *Journal of Applied Science and Technology Trends,* , 1(4), 140-147.

P. K., Q. W. (2021). Sports result prediction using data mining techniques in comparison with base line model. *Opsearch*, 54-70.

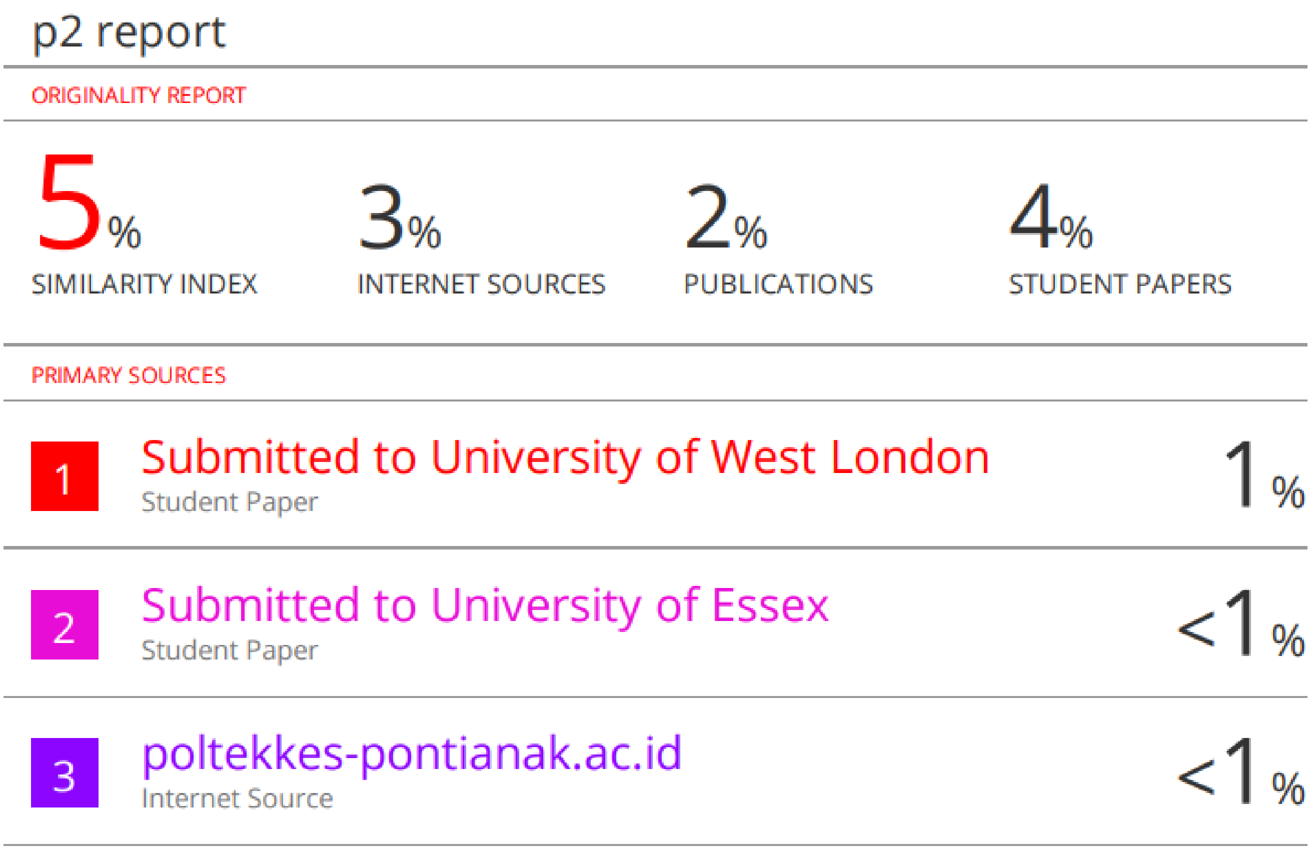
Salem, A. Z. (2018). Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. . *arXiv preprint arXiv*, 186-216.

Santra, A. M. (2020). Prediction of Most Valuable Bowlers of Indian Premier League (IPL). In Data Management, Analytics and Innovation. *ICDMAI 2020, Volume 2 Springer Singapore.*, 211-223.

Singh, S. &. (2017). IPL visualization and prediction using HBase. *Procedia computer science*, 910-915.

Tripathi, A. I. (2020). Prediction of IPL matches using Machine Learning while tackling ambiguity in results. *Indian J. Sci. Technol,* , 4013-4035.

Plagiarism



p2 report

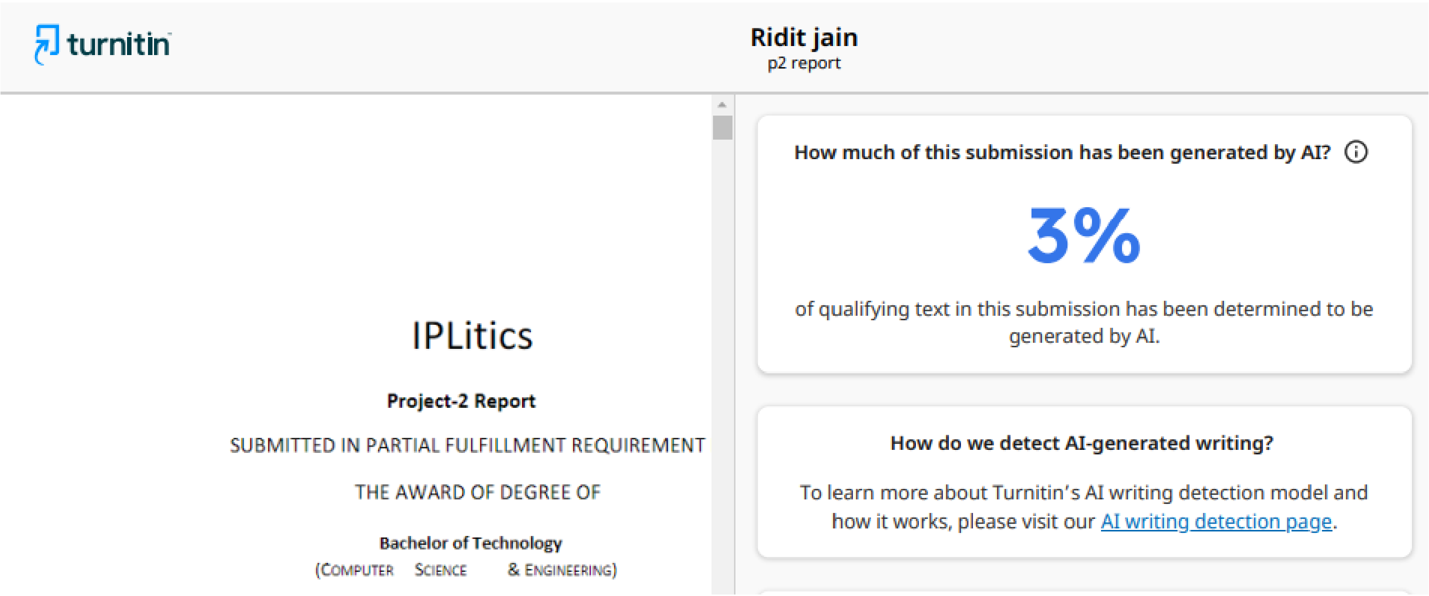
ORIGINALITY REPORT

5%	3%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of West London Student Paper	1 %
2	Submitted to University of Essex Student Paper	<1 %
3	poltekkes-pontianak.ac.id Internet Source	<1 %

AI Report



Ridit jain
p2 report

IPLitics

Project-2 Report

SUBMITTED IN PARTIAL FULFILLMENT REQUIREMENT

THE AWARD OF DEGREE OF

Bachelor of Technology

(COMPUTER SCIENCE & ENGINEERING)

How much of this submission has been generated by AI?

3%

of qualifying text in this submission has been determined to be generated by AI.

How do we detect AI-generated writing?

To learn more about Turnitin's AI writing detection model and how it works, please visit our AI writing detection page.