**A project Report on**

**on**

**“**"**IPL 2022 Analysis with GUI Interface”**

**Submitted to**

**KIIT Deemed to be University**

**In Partial Fulfillment of the Requirement for the Award of**

**BACHELOR’S DEGREE IN**

**COMPUTER SCIENCE AND COMMUNICATION ENGINEERING**

**MUSKAAN SHARMA** 2129079

**UNDER THE GUIDANCE OF**

**Sreeya Dash**

A green logo with white text

Description automatically generated

**SCHOOL OF COMPUTER ENGINEERING**

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

**BHUBANESWAR, ODISHA - 751024**

**April 2024**

A PROJECT REPORT

on

“**IPL 2022 Analysis with GUI Interface**”

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR’S DEGREE IN

COMPUTER SCIENCE AND COMMUNICATION ENGINEERING

BY

**MUSKAAN SHARMA** 2129079

UNDER THE GUIDANCE OF

Sreeya Dash

A green logo with white text

Description automatically generated

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024

A green logo with white text

Description automatically generated

CERTIFICATE

This is certify that the project entitled

“**IPL 2022 Analysis with GUI Interface**”

Submitted by

**MUSKAAN SHARMA** 2129079

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during the year 2023-2024, under our guidance.

Date: / /

Sreeya Dash

Project Guide

**Acknowledgements**

We are profoundly grateful to **Sreeya Dash** of **Affiliation** for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

MUSKAAN SHARMA

**ABSTRACT**

"IPL 2022 Analysis with GUI Interface" is a comprehensive project designed to explore and analyze data from the 2022 season of the Indian Premier League (IPL) through a user-friendly graphical interface. Leveraging Python programming language and libraries such as Pandas and NumPy for backend data processing and analysis, coupled with a GUI framework like Tkinter or PyQt for frontend interaction, this project offers cricket enthusiasts and data enthusiasts alike an intuitive platform to delve into various facets of the IPL 2022 season. Through interactive visualizations, customizable queries, and dynamic exploration of match outcomes, player performances, venue statistics, and more, users can gain valuable insights and a deeper understanding of one of the world's with frontend GUI enhances user engagement, accessibility, and overall user experience, making IPL 2022 analysis more interactive and enjoyable.

Keywords: IPL 2022, data analysis, graphical user interface, Python, Pandas, NumPy, Tkinter, PyQt, cricket, match outcomes, player performances, venue statistics, interactive visualizations, customizable queries, dynamic exploration, user engagement, accessibility, user experience

Contents

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Chapter Title** | **Page No.** |
|  | Certificate | 4 |
| Acknowledgement | 5 |
| Abstract | 6 |
| Table of Contents | 7 |
| Chapter 1 | Introduction & Overview | 8 |
| Chapter 2 | Scope: Information | 8 |
|  | Important Terminology & Methodology | 9-10 |
| Chapter 3 | Design | 10-11 |
|  | Project Background | 11-12 |
| Chapter 4 | Implementation | 12-18 |
| Chapter 5 | Result &, Conclusion | 19-22 |
| Chapter 6 | Future Scope | 22 |

Chapter 1

**Introduction & Overview**

The Indian Premier League (IPL) stands as one of the most electrifying and widely followed cricket tournaments globally, captivating audiences with its dynamic gameplay and star-studded line-ups. In the backdrop of the IPL 2022 season, this project embarks on a journey to dissect the intricacies of this cricketing extravaganza through meticulous data analysis. With the aid of Python programming language and essential libraries like Pandas and NumPy, coupled with an intuitive graphical user interface (GUI), this endeavor seeks to unravel insights into team performances, player dynamics, and match outcomes. This report presents a structured exploration of IPL 2022 data, aiming to provide cricket enthusiasts and data aficionados with a comprehensive understanding of the tournament's nuances and trends. From examining match statistics to delving into player metrics, this analysis promises to unveil compelling narratives and strategic insights that define the pulse of the IPL 2022 season.

Chapter 2

**Scope and Important Terminology/Methodology**

Scope:

The scope of this project encompasses a comprehensive analysis of the IPL 2022 season, focusing on various aspects such as match outcomes, team performances, player statistics, and venue dynamics. Through data-driven methodologies and advanced analytical techniques, the project aims to extract meaningful insights and trends from the vast repository of IPL data. Key objectives include identifying patterns in match results, uncovering factors influencing team strategies, evaluating player performances, and understanding the impact of different venues on game dynamics. Additionally, the project seeks to provide actionable recommendations for teams, coaches, and stakeholders based on the analysis findings. By delving into the rich tapestry of IPL data, this endeavor aims to contribute to the broader understanding of cricket analytics and facilitate informed decision-making in the context of the IPL ecosystem.

Important Terminology/Methodology:

**Variable:** A named storage location for data.

**Function:** A block of code that performs a specific task and can be reused.

**Class:** A blueprint for creating objects that defines properties and behaviors.

**Method:** A function associated with a class or object.

**Loop:** A control flow statement that repeats a block of code until a condition is met.

**Conditional Statement:** A statement that executes different actions based on whether a condition is true or false.

**List/Array:** A collection of items stored together in a single variable.

**Dictionary/Map:** A collection of key-value pairs used to store data.

**Module:** A file containing Python code that can be imported and used in other Python scripts.

**Library:** A collection of pre-written code or functions that can be reused in different programs.

**Exception Handling: The** process of dealing with errors that occur during program execution.

**Debugging: The** process of finding and fixing errors or bugs in code.

**Algorithm:** A step-by-step procedure or set of rules used to solve a problem.

Methodology refers to the systematic approach or procedure used to conduct research or analysis. In the context of coding and data analysis, methodology involves steps such as:

**Problem Identification:** Clearly defining the problem or objective of the analysis.

**Data Collection:** Gathering relevant data from reliable sources.

**Data Cleaning:** Preprocessing and preparing the data for analysis by handling missing values, removing duplicates, and formatting data types.

**Data Analysis:** Applying statistical techniques, algorithms, or models to derive insights and patterns from the data.

**Interpretation:** Analyzing and interpreting the results to draw meaningful conclusions.

**Validation:** Checking the validity and reliability of the analysis results.

**Documentation:** Documenting the entire process, including methodologies, findings, and conclusions, for reproducibility and transparency.

Chapter 3

**Design and Background**

Block Diagram

**Define Problem/Objective:** Determine the specific objectives of the IPL 2022 analysis, such as understanding team performances, player statistics, and match outcomes, to guide the analysis process.

**Data Collection:** Gather relevant data from reliable sources, including match statistics, player performances, venue details, and toss outcomes, to serve as the basis

**Data Preprocessing:** Cleanse and preprocess the collected data by handling missing values, removing duplicates, and formatting data types to ensure accuracy and consistency.

**A diagram of data analysis

Description automatically generated**

**Data Analysis:** Apply statistical techniques, machine learning algorithms, and data visualization methods to derive insights and patterns from the preprocessed IPL 2022 data.

**Interpretation:** Analyze and interpret the results of the data analysis to extract meaningful insights into team strategies, player dynamics, and trends in match outcomes.

**GUI Development:** Design and develop a graphical user interface (GUI) to interactively visualize and explore the IPL 2022 analysis results, providing users with an intuitive platform for data exploration.

**Validation:** Validate the analysis results by comparing them with known benchmarks, conducting sensitivity analyses, and ensuring the accuracy and reliability of the insights derived from the IPL 2022 data.

**Documentation:** Document the entire IPL 2022 analysis process, including methodologies, findings, and conclusions, to provide transparency, reproducibility, and a reference for future.

Background

The Indian Premier League (IPL) stands as a cornerstone of modern cricket, captivating audiences worldwide with its electrifying matches and star-studded line-ups. As one of the most lucrative and prestigious cricket tournaments, the IPL serves as a melting pot of talent, innovation, and entertainment, attracting top players, coaches, and stakeholders from across the globe. Amidst this backdrop, the project to conduct an IPL 2022 analysis using a GUI holds immense significance. By delving into the vast pool of IPL data, ranging from match statistics to player performances, this project aims to unearth valuable insights and trends that can reshape team strategies, inform player selections, and optimize match tactics. Moreover, by incorporating a graphical user interface, this endeavor seeks to democratize access to IPL data, empowering cricket enthusiasts, analysts, and team management alike to interactively explore and leverage data-driven insights for strategic decision-making and performance enhancement. In essence, this project not only underscores the pivotal role of data analytics in modern cricket but also underscores the potential of technology-driven solutions, such as GUI interfaces, to revolutionize how IPL data is harnessed and utilized for the betterment of the game.

Chapter 4

**Implementation**

1) DATA ANALYSIS

Code:

matches = pd.read\_csv(r"Desktop\IPL\_Matches\_2022.csv")

balls = pd.read\_csv(r"Desktop\IPL\_Ball\_by\_Ball\_2022.csv")

**This part of the code loads the CSV files containing IPL match and ball-by-ball data into pandas DataFrames named matches and balls, respectively.**

balls['batsman\_run'] = pd.to\_numeric(balls['batsman\_run'], errors='coerce') batter\_runs = balls.groupby('batter')['batsman\_run'].sum().reset\_index() batter\_runs\_sorted = batter\_runs.sort\_values(by='batsman\_run', ascending=False) bottom\_batters = batter\_runs\_sorted.tail(40) print("Bottom 12 batters based on total runs scored:") print(bottom\_batters.head)

**These lines convert the "batsman\_run" column to numeric, calculate the total runs scored by each batsman, and extract the bottom 40 batters based on their total runs.**

matches.info()

balls.info()

matches.describe()

balls.describe()

**These lines of code provide detailed information about the data types, non-null counts, and memory usage of each DataFrame, facilitating the understanding of the data's structure and quality. Additionally, they generate descriptive statistics (count, mean, min, max, etc.) for numerical columns in each DataFrame, offering insights into the distribution and summary of the data.**

duo=balls.groupby(['ID','batter','non-striker'])['total\_run'].sum().reset\_index() duo\_50=duo[duo['total\_run']>50 ].sort\_values(by=['total\_run'],ascending=False)

duo\_50

**These lines group the data by unique combinations of match ID, batsman, and non-striker, calculating the total runs scored by each pair. Then, it filters pairs where the total runs exceed 50 and sorts them in descending order based on the total runs.**

toss\_decision = matches['TossDecision'].value\_counts()

toss\_decision.head()  
  
**These lines count the frequency of each type of decision made by teams winning the toss, providing insights into the preferred choice between batting and bowling after winning the toss.**

toss\_won\_count=pd.DataFrame({"Toss Won":matches['TossWinner']}).value\_counts() labels=[x[0]for x in toss\_won\_count.keys()]

toss\_won\_count  
**This code creates a DataFrame "toss\_won\_count" to count the occurrences of each team winning the toss in the matches DataFrame. Then, it extracts the unique team names as labels for visualization. Finally, it displays the count of toss wins for each team.**

batsman\_strike\_rate = balls.groupby('batter').apply(lambda x: (x['batsman\_run'].sum() / len(x)) \* 100).reset\_index()

batsman\_strike\_rate.columns = ['Batsman', 'StrikeRate'] batsman\_strike\_rate\_sorted = batsman\_strike\_rate.sort\_values(by='StrikeRate', ascending=False) print(batsman\_strike\_rate\_sorted)

**These lines calculate the strike rate of each batsman based on the total runs scored and the number of balls faced, providing an indication of their batting efficiency. The results are then sorted in descending order of strike rate and displayed.**

balls["extra\_type"].fillna("Normal Delivery",inplace=True) balls["player\_out"].fillna('No One Out',inplace=True) balls["kind"].fillna(0,inplace=True) balls["fielders\_involved"].fillna(0,inplace=True)

**These lines of code fill missing values in the "extra\_type", "player\_out", "kind", and "fielders\_involved" columns of the DataFrame "balls" with appropriate default values. This ensures that the data remains consistent and ready for analysis, preventing any potential issues caused by missing values in these columns.**

class TeamPerformanceAnalysis:

def \_\_init\_\_(self, matches, balls):

self.matches = matches

self.balls = balls

def get\_team\_stats(self, team):

# Filter matches where the team is either batting or bowling

team\_matches = self.matches[(self.matches['Team1'] == team) | (self.matches['Team2'] == team)]

# Calculate total runs scored by the team

total\_runs = self.balls[self.balls['BattingTeam'] == team]['total\_run'].sum()

# Calculate win-loss ratio

total\_matches = len(team\_matches)

total\_wins = len(team\_matches[team\_matches['WinningTeam'] == team])

win\_loss\_ratio = total\_wins / total\_matches if total\_matches > 0 else 0

return {

'Team': team,

'TotalRuns': total\_runs,

'TotalMatches': total\_matches,

'TotalWins': total\_wins,

'WinLossRatio': win\_loss\_ratio

}

team\_analysis = TeamPerformanceAnalysis(matches, balls)

team\_stats = team\_analysis.get\_team\_stats('Mumbai Indians')

print("Team Stats:", team\_stats)

**The TeamPerformanceAnalysis class takes two parameters, matches and balls, representing match and ball data, respectively. The get\_team\_stats method calculates various statistics for a given team, such as total runs scored, total matches played, total wins, and win-loss ratio.**

**An instance of TeamPerformanceAnalysis is created with match and ball data (matches and balls), and then the get\_team\_stats method is called with the team name 'Mumbai Indians'. The resulting team statistics are printed.**

2) GUI

def analyze\_team\_runs(button):

if len(team\_runs) == 0:

print("No data available for analysis.")

elif len(team\_runs) == 1:

print("Only one team played in the match.")

else:

# Find the team with the maximum runs

max\_runs\_team = team\_runs.idxmax()

max\_runs = team\_runs.max()

# Find the team with the minimum runs

min\_runs\_team = team\_runs.idxmin()

min\_runs = team\_runs.min()

# Display the analysis results

print("Team '{}' scored the most runs: {}".format(max\_runs\_team, max\_runs))

print("Team '{}' scored the least runs: {}".format(min\_runs\_team, min\_runs))

analyze\_button = widgets.Button(description="Analyze Team Runs")

analyze\_button.on\_click(analyze\_team\_runs)

# Display the button

display(analyze\_button)

**The analyze\_team\_runs function takes a button click event (button) as input. It checks if there is data available for analysis. If there is only one team in the match, it prints a message indicating that. Otherwise, it determines the team that scored the most runs and the team that scored the least runs, along with their respective run totals, and prints out the analysis results.**

Chapter 5

**Result and Conclusion**

DATA ANALYSIS

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

GUI

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a graph

Description automatically generated

A screenshot of a computer

Description automatically generated

Conclusion

In conclusion, the provided code segments offer a comprehensive approach to cricket data analysis, covering various aspects such as team performance, individual player statistics, and match insights. By leveraging functions and widgets, users can efficiently analyze team runs, batsman strike rates, and other key metrics, enabling deeper insights into cricket matches and players' performances. Overall, these code segments provide valuable tools for cricket enthusiasts and analysts alike to delve into the intricacies of the game and extract meaningful insights from the data.

Chapter 6

**Future Work**

Furthermore, for future enhancements, integrating machine learning and deep learning models could extend the analysis capabilities to predictive tasks. By leveraging historical match data, such models could forecast outcomes, player performances, and other game-related metrics with greater accuracy. This integration would not only enhance the depth of analysis but also enable proactive insights for strategic decision-making in cricket.

**`**