JAAFR INTERNATIONAL RESEARCH JOURNAL
JOURNAL OF ADVANCE AND FUTURE RESEARCH
JAAFR.ORG | ISSN : 2984-889X
An International Open Access, Peer-reviewed, Refereed Journal

Indian Premier League (IPL) Data Analysis

[Tanishka sandip joshi][1], [Prof. Jadhav A.E][2]

[1]Student, Department of Computer Engineering, [Sau. Sundarabai Manik Adsul Poly.], Maharashtra, India

[2]Guide, Department of Computer Engineering, [Sau. Sundarabai Manik Adsul Poly], Maharashtra, India

Abstract

The Indian Premier League (IPL) generates a large volume of ball-by-ball and match-level data every season, offering rich potential for data-driven insight into player and team performance. This paper presents an end-to-end sports-analytics pipeline that transforms raw batting, bowling, match and player data from three IPL seasons (2021–2023) into actionable insights using Structured Query Language (SQL), Python, and Microsoft Power BI. The dataset comprises 206 matches, 3,268 batting entries and 2,430 bowling entries across ten franchise teams. After data cleaning — correcting data types and handling missing values — SQL queries were used for aggregation-based analysis, while Python (pandas, Matplotlib, Seaborn) was used for exploratory data analysis (EDA) and visualization. An interactive Power BI dashboard was then built to present team and player Key Performance Indicators (KPIs). Results show Shubman Gill as the leading run-scorer (1,851 runs), Mohammed Shami as the leading wicket-taker (67 wickets), and Chennai Super Kings and Royal Challengers Bengaluru tied as the most successful teams by match wins (25 each). The study demonstrates how combining relational-database querying, code-based EDA and business-intelligence dashboarding provides a robust, cross-validated and easily interpretable approach to sports-performance analysis.

Keywords — IPL, Data Analytics, Exploratory Data Analysis (EDA), SQL, Python, Pandas, Power BI, Sports Analytics, Data Visualization.

I. INTRODUCTION

The Indian Premier League (IPL), launched by the Board of Control for Cricket in India (BCCI) in 2008, is one of the most-watched Twenty20 (T20) cricket leagues in the world. Every season generates a large volume of structured data — ball-by-ball events, batting and bowling summaries, match results, and player profiles — that is rich in analytical potential but difficult to interpret in its raw form. With the growing adoption of data-driven decision-making in professional sport, franchises, broadcasters and fantasy-sports platforms are increasingly interested in extracting actionable insights from match data.

This paper is motivated by the opportunity to apply a complete data-analytics stack — SQL for data querying, Python for exploratory data analysis (EDA), and Microsoft Power BI for interactive dashboarding — to three consecutive IPL

seasons (2021–2023). The specific objectives of this work are to: (i) analyse player performance metrics such as batting average, bowling economy and strike rate using SQL; (ii) evaluate team performance based on win–loss ratios, total runs and total wickets; (iii) perform data cleaning and EDA in Python to identify trends and relationships in the data; and (iv) visualize the key findings through static charts and an interactive Power BI dashboard.

II. LITERATURE REVIEW

Sports analytics is the application of statistical and computational methods to sports data to support decision-making by coaches, team management and fans. In cricket, analytics has evolved from simple scorecards to advanced metrics such as strike rate, economy rate and win-probability models [1]. Exploratory Data Analysis (EDA), an approach popularised by Tukey [2], emphasises summarizing a dataset's main characteristics through visual and statistical methods prior to formal modelling; this forms the analytical core of the present study.

Several public dashboards analyse IPL data using Power BI, Tableau or Python, typically focusing on isolated run-scoring or wicket-taking charts [3]. However, most publicly available work presents disconnected visualizations rather than a reproducible pipeline that spans raw CSV data, SQL-based querying, Python-based EDA and an interactive dashboard. This paper addresses that gap by combining all three tools — SQL, Python and Power BI — in a single, cross-validated pipeline, reflecting the toolchain typically used in real-world data-analyst roles.

III. METHODOLOGY

The proposed system follows a five-stage workflow: (1) data collection, (2) data cleaning and preparation, (3) SQL-based structured analysis, (4) Python-based exploratory data analysis and visualization, and (5) interactive dashboard design in Power BI, as illustrated in Fig. 1.

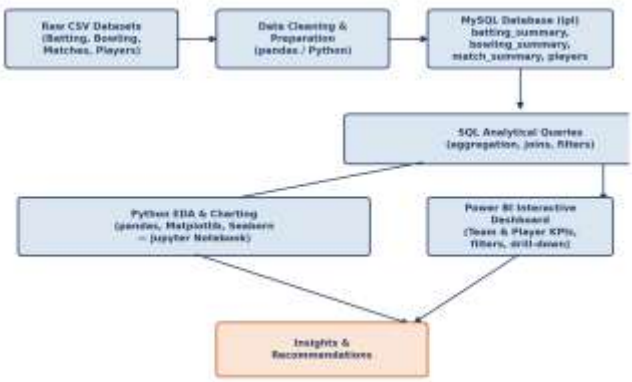

Fig. 1. System architecture and data flow of the proposed pipeline

A. Dataset

The dataset, sourced from the codebasics.io Resume Project Challenge, consists of four CSV files covering IPL seasons 2021–2023: Batting_summary.csv (3,268 rows), Bowling_Summary.csv (2,430 rows), MATCH SUMMARY.csv (206 rows) and Players.csv (292 rows). The four files were imported into a MySQL database for relational, SQL-based querying, and also loaded directly into pandas DataFrames for Python-based analysis.

B. Data Cleaning

The Strike Rate (SR) column in the batting dataset was stored as text and was converted to a numeric type using pandas.to_numeric() with invalid entries coerced to NaN. This revealed 54 rows (1.7%) with missing SR values, corresponding to batsmen who faced zero balls; these were excluded from rate-based calculations only. No duplicate records were found in any of the four files.

C. SQL-Based Analysis

Analytical SQL queries were written to compute total runs and wickets, strike rate, batting average and boundary percentage per player, applying a minimum-balls-faced threshold (60 balls) to avoid short, unrepresentative innings skewing the rankings. A representative query is shown below.

```
SELECT batsmanName, (SUM(runs)/SUM(balls))*100 AS Strike_rate
FROM batting_summary INNER JOIN players
ON batting_summary.batsmanName = players.name
GROUP BY batsmanName HAVING SUM(balls) >= 60
ORDER BY Strike_rate DESC LIMIT 10;
```

D. Exploratory Data Analysis and Dashboarding

The cleaned data was analysed in a Jupyter Notebook using pandas for aggregation, and Matplotlib/Seaborn for univariate analysis (e.g., top run-scorers), bivariate analysis (e.g., runs vs. strike rate) and trend analysis (matches played per year). Finally, the aggregated metrics were connected to Power BI to design an interactive dashboard with KPI cards, ranked leaderboards and season/team-level filters.

IV. RESULTS AND DISCUSSION

Fig. 2 shows the top 10 run-scorers across the three seasons studied. Shubman Gill leads with 1,851 runs, followed by Faf du Plessis (1,831) and Ruturaj Gaikwad (1,593), confirming that consistent top-order batting was a decisive factor over this period.

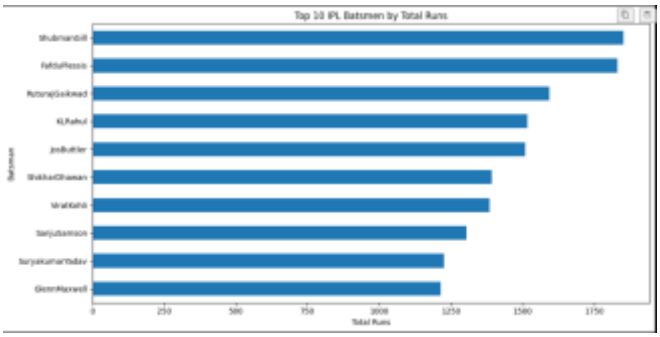

Fig. 2. Top 10 IPL batsmen by total runs (2021–2023)

Fig. 3 presents the leading wicket-takers. Mohammed Shami tops the list with 67 wickets, ahead of Yuzvendra Chahal (66) and Harshal Patel (65), indicating that wicket-taking success was not confined to a single bowling style.

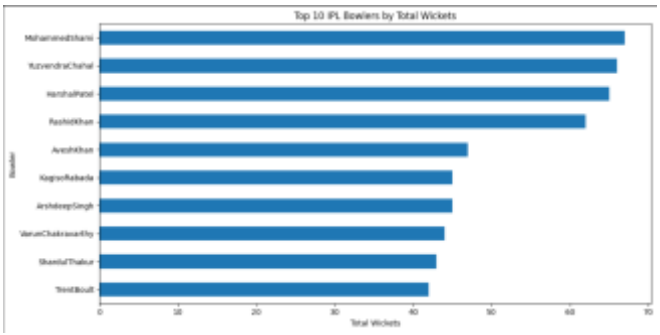

Fig. 3. Top 10 IPL bowlers by total wickets (2021–2023)

Table I summarizes team-level performance. Chennai Super Kings (CSK) and Royal Challengers Bengaluru (RCB) share the highest win count (25 wins each); CSK also leads in total runs (7,372) and total wickets (271), suggesting that its strong record was supported by balanced batting and bowling performance.

Team	Wins	Total Runs	Total Wickets
CSK	25	7,372	271
RCB	25	7,028	266
GT	23	5,423	207
RR	22	7,181	237
KKR	21	6,870	253
MI	20	6,937	245

TABLE I. Team-wise Wins, Runs and Wickets (2021–2023)

A scatter analysis of total runs against strike rate (not shown here for brevity) revealed no strong linear relationship, indicating that run-scoring consistency and strike rate are largely independent skills among IPL batsmen. The number of matches played per season rose from 59 in 2021 to 74 in 2022 following the league's expansion to ten teams, and remained stable at 73 in 2023.

The Power BI dashboard cross-validated all of the above findings and additionally surfaced KPIs such as the best team batting average (GT, 32.28), the best individual batting average (KL Rahul, 51.00) and the best bowling economy (Sunil Narine, 6.60), all of which matched the corresponding SQL and pandas computations, confirming the correctness and consistency of the analytical pipeline across all three tools.

V. CONCLUSION AND FUTURE SCOPE

This paper presented an end-to-end sports-analytics pipeline for IPL data covering the 2021–2023 seasons, combining SQL, Python and Power BI. The pipeline was validated by cross-checking results across all three tools, and identified Shubman Gill, Mohammed Shami, and CSK/RCB as the standout performers of the period. The work demonstrates that combining relational-database querying with code-based EDA and BI dashboarding provides a reliable and easily interpretable approach to sports-performance analysis.

Future work may extend this pipeline to ball-by-ball commentary data for phase-wise performance analysis, incorporate machine-learning models for predictive analytics such as match-outcome forecasting, and add natural-language query features to the Power BI dashboard for broader accessibility.

ACKNOWLEDGMENT

The authors would like to thank the Department of Computer Engineering, [Name of College / Institute], for providing the resources and guidance necessary to carry out this work.

REFERENCES

[1] S. Chetan and R. Kumar, "A survey on cricket analytics using machine learning," Int. J. Comput. Appl., vol. 178, no. 5, 2021.

[2] J. W. Tukey, Exploratory Data Analysis. Reading, MA: Addison-Wesley, 1977.

[3] S. Sett, "Indian-Premier-league," GitHub repository, 2026. [Online]. Available: https://github.com/Github-sanket07sett/Indian-Premier-league

[4] codebasics.io, "Resume Project Challenge #10 – IPL Dataset," 2024. [Online]. Available: https://codebasics.io/challenge/codebasics-resume-project-challenge

[5] pandas Development Team, "pandas Documentation." [Online]. Available: https://pandas.pydata.org/docs/

[6] Matplotlib Development Team, "Matplotlib Documentation." [Online]. Available: https://matplotlib.org/stable/

[7] M. Waskom, "seaborn: statistical data visualization," J. Open Source Softw., vol. 6, no. 60, p. 3021, 2021.

[8] Oracle Corp., "MySQL 8.0 Reference Manual." [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/

[9] Microsoft Corp., "Power BI Documentation." [Online]. Available: https://learn.microsoft.com/en-us/power-bi/