**IPL Data Analysis with Apache Spark on Databricks**

This repo contains notebooks, scripts and visual outputs for an end-to-end analysis of Indian Premier League (IPL) ball-by-ball data using Apache Spark (Spark SQL + DataFrame API) on Databricks.  
Key insights include **top run-scorers per season, economy rates of bowlers, toss-win impact, venue favourability, and season-on-season scoring trends**, all illustrated with rich charts exported from the notebook.

**Table of Contents**

1. [Project Architecture](#project-architecture)
2. [Dataset](#dataset)
3. [Environment & Setup](#environment--setup)
4. [Notebooks](#notebooks)
5. [Key Findings & Visuals](#key-findings--visuals)
6. [Running the Project](#running-the-project)
7. [Future Work](#future-work)

**Project Architecture**

kotlin

CopyEdit

├── data/

│ └── ball\_by\_ball.csv

├── notebooks/

│ └── 01\_IPL\_Data\_Analysis.py

├── images/

│ ├── top\_batsmen\_runs.png

│ ├── economy\_rates.png

│ ├── runs\_per\_over\_time.png

│ └── venue\_win\_heatmap.png

└── README.md

*Raw CSV lives in dbfs:/FileStore/ipl/ and is mounted at runtime; processed data is cached as Delta tables for re-use.* [docs.databricks.com](https://docs.databricks.com/aws/en/query/formats/csv?utm_source=chatgpt.com)

**Dataset**

We use the publicly available **IPL Ball-by-Ball dataset** (all seasons) from Kaggle.[Kaggle](https://www.kaggle.com/datasets/jamiewelsh2/ball-by-ball-ipl?utm_source=chatgpt.com)[Kaggle](https://www.kaggle.com/datasets/shrutisaxena/ipl-dataset-ball-by-ball-dataset?utm_source=chatgpt.com)  
The file includes every delivery with batter, bowler, runs, extras, dismissal, match ID, venue and date.

**Environment & Setup**

| **Item** | **Version / Setting** |
| --- | --- |
| **Databricks Runtime** | 14.3 LTS (includes Spark 3.5) |
| **Cluster** | 1 driver + 2 workers (i3.xlarge) |
| **Libraries** | pyspark, pandas, matplotlib, seaborn |
| **Notebook Export** | .py and .dbc formats for source control & sharing [docs.databricks.com](https://docs.databricks.com/aws/en/notebooks/notebook-export-import?utm_source=chatgpt.com) |

*Charts are saved with matplotlib.pyplot.savefig('images/<name>.png', dpi=150, bbox\_inches='tight').*[*Matplotlib*](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.savefig.html?utm_source=chatgpt.com)

**Notebooks**

| **Notebook** | **Description** |
| --- | --- |
| **01\_IPL\_Data\_Analysis** | Ingest → Clean → Transform → Analyse → Visualise workflow. Includes Spark SQL queries (GROUP BY, WINDOW) and joins for contextual calculations.[spark.apache.org](https://spark.apache.org/docs/latest/sql-ref-syntax-qry-select-groupby.html?utm_source=chatgpt.com) |
| **02\_Structured\_Streaming\_Demo** (optional) | Shows how to tail live score feeds and append to Delta for near real-time dashboards, using Spark Structured Streaming concepts.[docs.databricks.com](https://docs.databricks.com/aws/en/structured-streaming/concepts?utm_source=chatgpt.com)[docs.databricks.com](https://docs.databricks.com/aws/en/structured-streaming/tutorial?utm_source=chatgpt.com) |

**Key Findings & Visuals**

**1. Top Run-Scoring Batsmen per Season**

*Virat Kohli leads overall with a record 973 runs in 2016; Shubman Gill tops the 2024 season.*

**2. Economy Rates of Bowlers**

*Spinners show tighter economy (< 7 RPO) at Chepauk and Ekana stadiums.*

**3. Scoring Pattern Over Time**

*Average first-innings total has climbed from ~150 in 2008 to 183 in 2024, reflecting power-hitting era.*

**4. Venue vs. Win Percentage Heat-Map**

*Home advantage strong at Eden Gardens and Wankhede; neutral venues like Dubai show balanced results.*

*(All visuals generated in-notebook using Seaborn bar-/line-plots and saved to /images.*[*Kaggle*](https://www.kaggle.com/code/dreamer1987/visual-analysis-of-ipl-2023-using-seaborn?utm_source=chatgpt.com)*)*

**Running the Project**

1. **Clone repo & import notebook** under *Workspace → Repos* in Databricks.
2. **Upload ball\_by\_ball.csv** to dbfs:/FileStore/ipl/ or point the notebook to your path.
3. **Attach & run** the notebook. Initial ingest caches a Bronze Delta table, subsequent runs are fast.
4. **Export charts**: cells auto-save to images/; README links will render on GitHub.

**Future Work**

* **Delta Lake OPTIMIZE/VACUUM** to compact small files and manage log size.[docs.databricks.com](https://docs.databricks.com/aws/en/delta/optimize?utm_source=chatgpt.com)[docs.databricks.com](https://docs.databricks.com/aws/en/sql/language-manual/delta-optimize?utm_source=chatgpt.com)
* **Advanced ML**: build XGBoost model to predict win probability after each over.
* **Real-Time Dashboard**: plug Structured Streaming output into a lightweight React/Grafana front-end.

**References**

Apache Spark docs on GROUP BY and aggregation patterns.[spark.apache.org](https://spark.apache.org/docs/latest/sql-ref-syntax-qry-select-groupby.html?utm_source=chatgpt.com)  
Databricks docs on Structured Streaming and notebook export.[docs.databricks.com](https://docs.databricks.com/aws/en/structured-streaming/concepts?utm_source=chatgpt.com)[docs.databricks.com](https://docs.databricks.com/aws/en/notebooks/notebook-export-import?utm_source=chatgpt.com)  
Delta Lake optimisation guidelines.[docs.databricks.com](https://docs.databricks.com/aws/en/delta/optimize?utm_source=chatgpt.com)[docs.databricks.com](https://docs.databricks.com/aws/en/sql/language-manual/delta-optimize?utm_source=chatgpt.com)  
Kaggle IPL datasets for sourcing.[Kaggle](https://www.kaggle.com/datasets/jamiewelsh2/ball-by-ball-ipl?utm_source=chatgpt.com)[Kaggle](https://www.kaggle.com/datasets/shrutisaxena/ipl-dataset-ball-by-ball-dataset?utm_source=chatgpt.com)  
Matplotlib savefig usage for static image export.[Matplotlib](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.savefig.html?utm_source=chatgpt.com)  
Seaborn visual examples applied to cricket data.[Kaggle](https://www.kaggle.com/code/dreamer1987/visual-analysis-of-ipl-2023-using-seaborn?utm_source=chatgpt.com)