Subject: Big Data Analytics (CSC702)

AY: 2024-25

Experiment 10(Mini Project)

Aim: Design the infrastructure of a Big Data Application.

Tasks to be completed by the students:
Task 1: Choose a problem definition which requires handling Big Data.Task 2: Design the data pipeline for your application.
Task 3: Deploy your project on suitable platform.
Task 4: Test your application with different volume, variety and velocity of data.

Report on Mini Project

Subject: Big Data Analytics (CSC702)

AY: 2024-25

IPL DATA ANALYSIS

Raghav Rathi : 2113208

Ronak Singh : 2103087

Parth Kadam : 2103074

Aaryan Jha : 2103071

Guided By

(Dr. Arti Deshpande)

CHAPTER 1: INTRODUCTION

In the era of modern data-driven decision-making and large-scale event-driven architectures, analyzing vast amounts of data in real-time has become critical for driving insights and optimizing strategies. The **"IPL Data Analysis"** project is designed to tackle the challenges of handling large volumes of structured and semi-structured data, specifically from the Indian Premier League (IPL), by creating an end-to-end data engineering pipeline using **Apache Spark** on **Databricks**.

In a typical IPL data ecosystem, data is generated from various sources like match statistics, player performance, stadium data, and audience engagement. This data, once processed and analyzed, can provide crucial insights into team performance, player trends, and match outcomes. The project leverages **Apache Spark**, a powerful distributed data processing engine, to ingest, process, and analyze this data efficiently.

The architecture incorporates Spark's distributed computing capabilities to handle large-scale IPL data in real-time and batch modes. **Databricks** is used to orchestrate and optimize the processing pipeline, enabling easy management of data flows from ingestion to analysis. The data is collected from multiple sources such as CSV files, APIs, or real-time streams and is processed using **Spark's DataFrames** and **SQL** for efficient analysis.

Storage of processed data is handled by **Delta Lake**, which provides ACID transactions and scalable metadata management on top of data lakes. This ensures that IPL data is ingested, transformed, and stored efficiently for real-time analytics. Additionally, **Databricks SQL** is employed for interactive queries, and **visual dashboards** are created to provide real-time insights into various aspects like top-performing players, match statistics, and trends across seasons. These insights help teams, analysts, and enthusiasts make data-driven decisions.

By offering both structured data analysis through **Spark SQL** and **Delta Lake**, and advanced querying via **Databricks** dashboards, the **"IPL Data Analysis"** project ensures that IPL data can be easily processed, analyzed, and visualized for immediate and long-term insights. This architecture provides a robust solution to IPL data management, making it easier to explore and derive insights from large volumes of complex data.

CHAPTER 2: DATA DESCRIPTION AND ANALYSIS

In the **"IPL Data Analysis"** project, data is collected from various sources related to the Indian Premier League (IPL), such as match statistics, player performance records, team rankings, and historical match outcomes. This data is rich in information and critical for generating insights into player trends, match strategies, and team performance throughout different seasons.

The data is stored in various formats such as **CSV files**, APIs, and potentially real-time data streams. The challenge is to process and analyze this massive volume of IPL data efficiently. To handle this, **Apache Spark** on **Databricks** is employed, enabling distributed processing of structured and semi-structured data at scale.

Data Ingestion and Processing

Data is ingested into the system using **Apache Spark** in **Databricks**, which supports seamless integration with various data sources. Data can be ingested either in batch mode or as real-time streams from APIs that provide live updates on IPL matches. Once ingested, the data undergoes preprocessing, which involves cleaning, transformation, and integration of data from different sources to create a unified dataset for analysis.

The **structured and semi-structured IPL data** contains several key attributes such as:

- **Match Details**: Includes match date, venue, teams involved, and match outcomes.
- **Player Performance**: Tracks runs scored, wickets taken, strike rates, economy rates, and fielding statistics.
- **Team Performance**: Includes team scores, win/loss margins, and rankings across seasons.
- **Audience and Engagement Metrics**: Tracks fan engagement, viewership statistics, and social media trends.

Parallel Processing with Spark

The power of **Apache Spark** lies in its ability to perform distributed and parallel processing of large datasets. Spark efficiently processes IPL data in parallel, ensuring that even large volumes of match and player statistics are ingested and analyzed quickly. Databricks optimizes this

pipeline, ensuring the system remains responsive and scalable as the data volume increases.

Data Storage: Delta Lake and Spark SQL

Once the data is processed, it is stored in **Delta Lake**, a storage layer built on top of data lakes that supports ACID transactions and schema enforcement. Delta Lake ensures that the IPL data is reliable, consistent, and easily accessible for querying.

- **Delta Lake**: Used to store the structured IPL data such as player and team statistics. The data is stored in a structured format for efficient querying and analysis.
- **Spark SQL**: Used to run SQL queries on the processed data to generate insights, such as identifying top-performing players in a season, calculating player averages, or tracking team win rates across multiple seasons.

Data Analysis with Spark and Databricks

The data analysis in this project occurs on two fronts:

1. **Structured Data Analysis**: Using **Spark SQL**, we can perform in-depth queries and analysis on structured data from Delta Lake. For example:
    - Identifying the highest run-scorer in a season.
    - Calculating player strike rates and bowling economy rates.
    - Analyzing the win/loss patterns of teams across seasons.

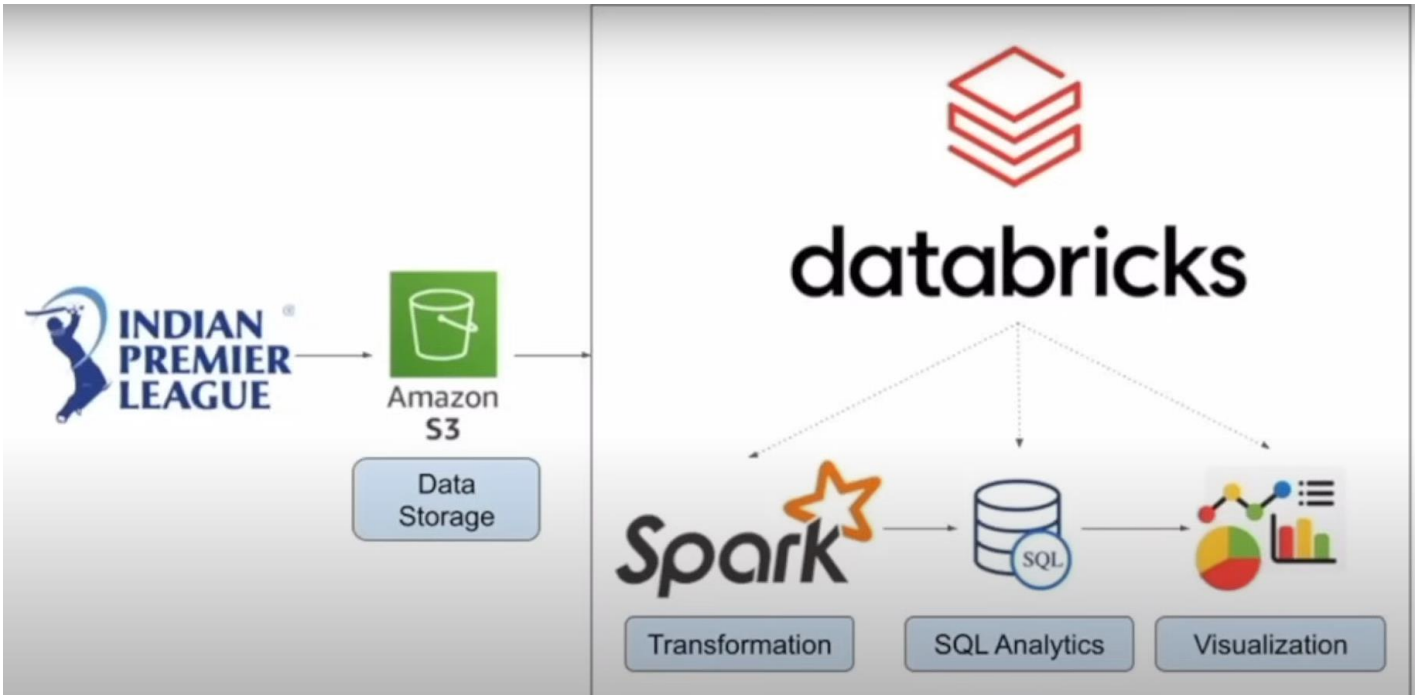
These insights are useful for understanding player and team performance, strategy evaluation, and predicting match outcomes.

2. **Advanced Analysis and Dashboards**: **Databricks** offers rich visualization tools that enable users to create interactive dashboards for real-time insights into IPL data. Analysts and stakeholders can visualize player performance trends, match statistics, and team standings over time. The visual insights provided by Databricks allow for interactive exploration of IPL data, facilitating better decision-making and strategy formulation.

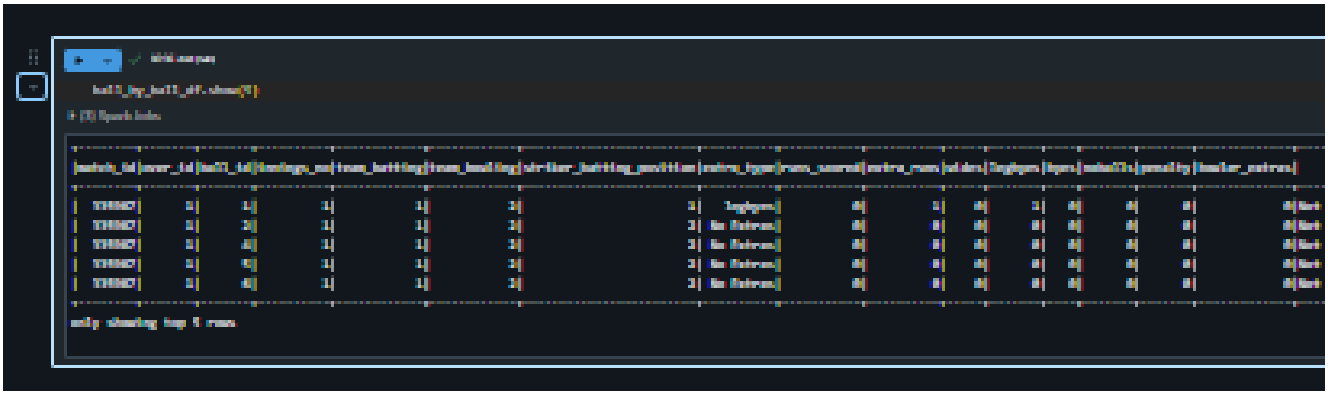
By leveraging **Apache Spark** for distributed processing and **Databricks** for real-time data

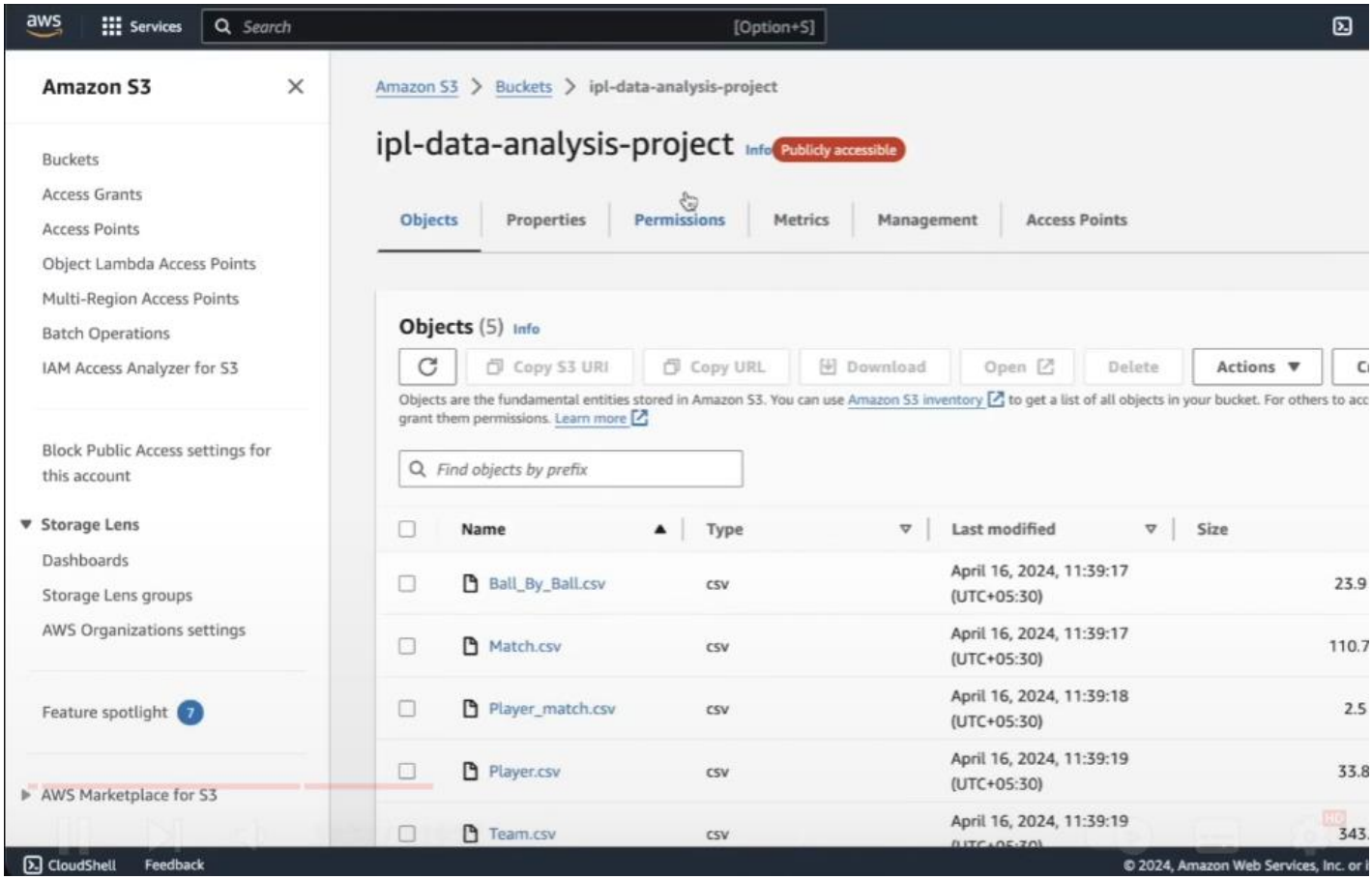
orchestration and visualization, the **"IPL Data Analysis"** project enables efficient, scalable, and insightful analysis of IPL data. This end-to-end pipeline is designed to handle large datasets, making it a powerful tool for IPL data analysts, teams, and decision-makers.

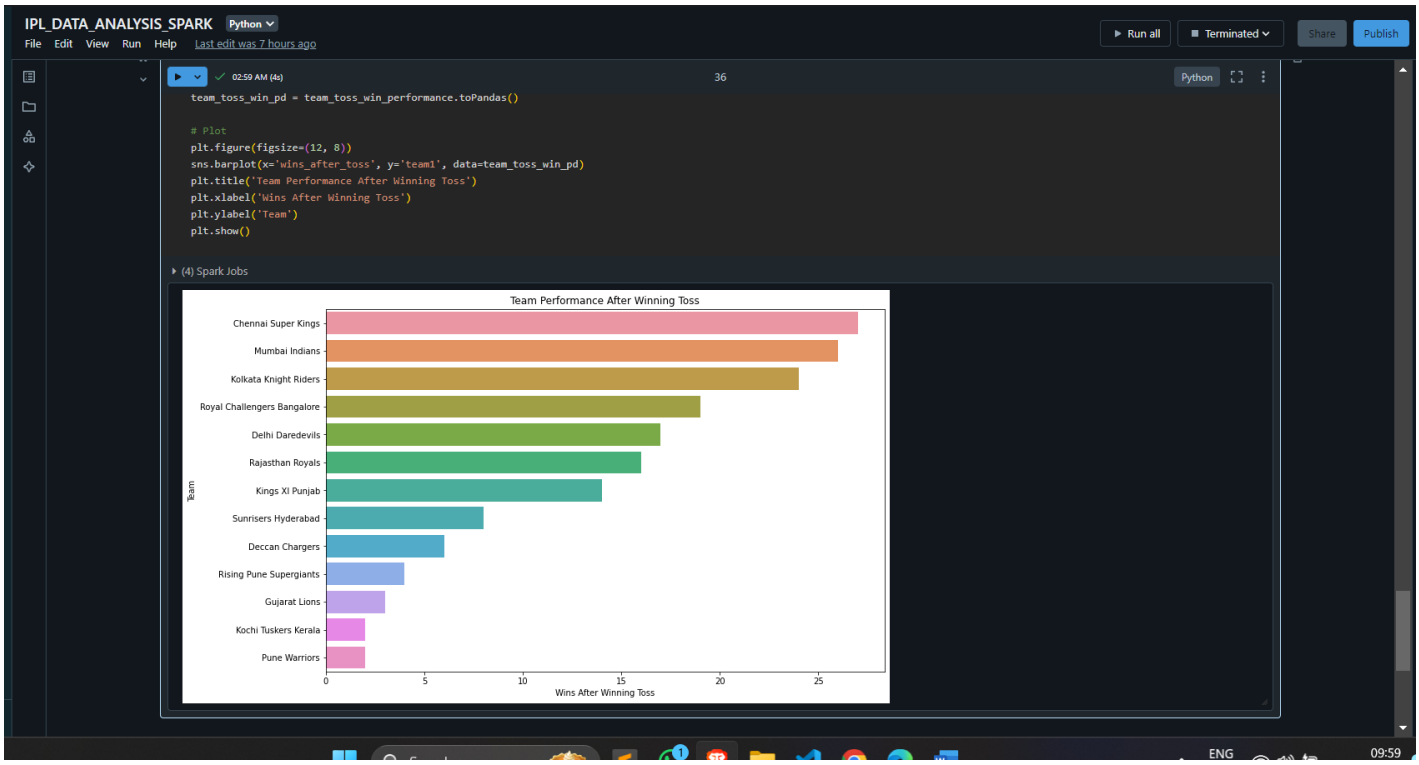
CHAPTER 3: DESIGN OF DATA PIPELINE

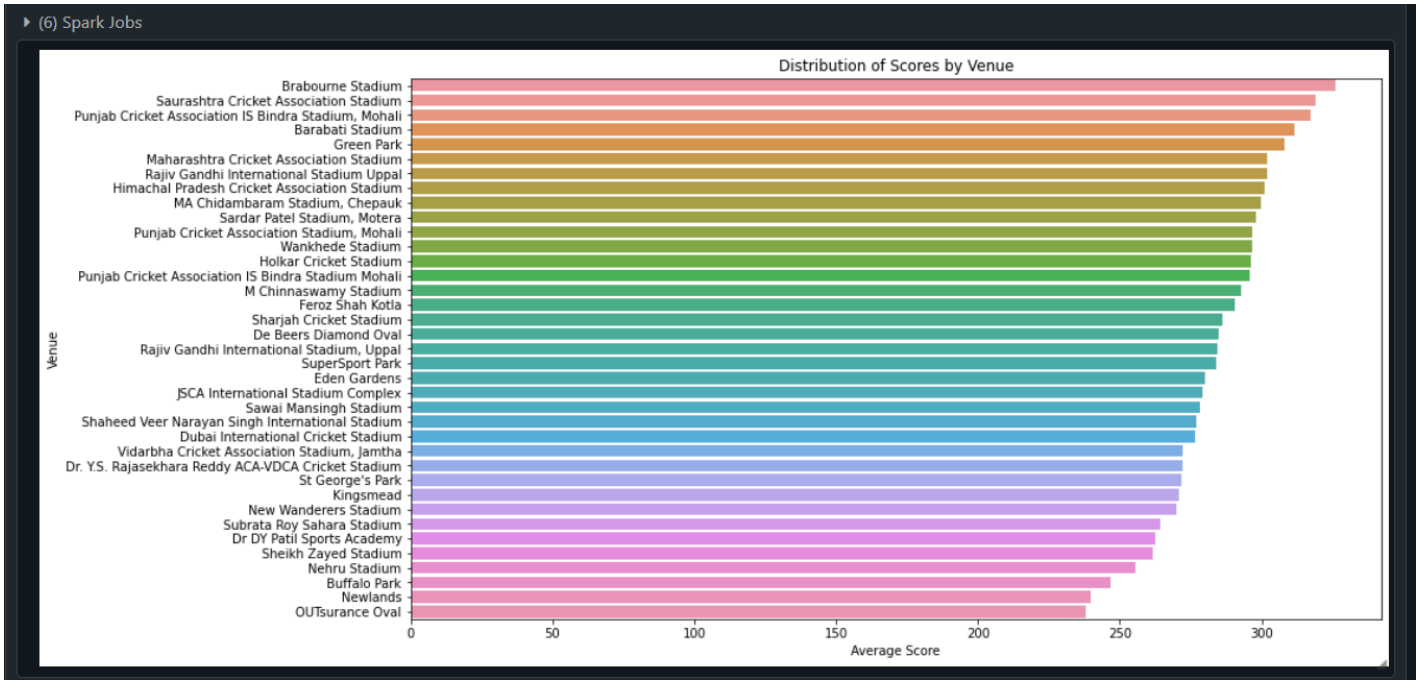


CHAPTER 4: RESULT ANALYSIS






IPL_DATA_ANALYSIS_SPARK
Python
File Edit View Run Help
Last edit was 7 hours ago
Run all
Terminated
Share
Publish
02:59 AM (4s)
36
Python
team_toss_win_pd = team_toss_win_performance.toPandas()
# Plot
plt.figure(figsize=(12, 8))
sns.barplot(x='wins_after_toss', y='team1', data=team_toss_win_pd)
plt.title('Team Performance After Winning Toss')
plt.xlabel('Wins After Winning Toss')
plt.ylabel('Team')
plt.show()
(4) Spark Jobs
Team Performance After Winning Toss
Chennai Super Kings
Mumbai Indians
Kolkata Knight Riders
Royal Challengers Bangalore
Delhi Daredevils
Rajasthan Royals
Kings XI Punjab
Sunrisers Hyderabad
Deccan Chargers
Rising Pune Supergiants
Gujarat Lions
Kochi Tuskers Kerala
Pune Warriors
Team
0
5
10
15
20
25
Wins After Winning Toss
ENG
09:59


(6) Spark Jobs
Distribution of Scores by Venue
Brabourne Stadium
Saurashtra Cricket Association Stadium
Punjab Cricket Association IS Bindra Stadium, Mohali
Barabati Stadium
Green Park
Maharashtra Cricket Association Stadium
Rajiv Gandhi International Stadium Uppal
Himachal Pradesh Cricket Association Stadium
MA Chidambaram Stadium, Chepauk
Sardar Patel Stadium, Motera
Punjab Cricket Association Stadium, Mohali
Wankhede Stadium
Holkar Cricket Stadium
Punjab Cricket Association IS Bindra Stadium Mohali
M Chinnaswamy Stadium
Feroz Shah Kotla
Sharjah Cricket Stadium
De Beers Diamond Oval
Rajiv Gandhi International Stadium, Uppal
SuperSport Park
Eden Gardens
JSCA International Stadium Complex
Sawai Mansingh Stadium
Shaheed Veer Narayan Singh International Stadium
Dubai International Cricket Stadium
Vidarbha Cricket Association Stadium, Jamtha
Dr. Y.S. Rajasekhara Reddy ACA-VDCA Cricket Stadium
St George's Park
Kingsmead
New Wanderers Stadium
Subrata Roy Sahara Stadium
Dr DY Patil Sports Academy
Sheikh Zayed Stadium
Nehru Stadium
Buffalo Park
Newlands
OUTsurance Oval
Venue
0
50
100
150
200
250
300
Average Score

The visualized data from the analytics dashboard provides insight into three key areas

DATA VISUALIZATION AND INSIGHTS

In the **IPL Data Analysis** project, Apache Spark and Databricks are used to process, analyze, and visualize large volumes of IPL data, providing valuable insights into player performance, team strategies, and match outcomes. The visualized data is presented through an analytics dashboard, enabling stakeholders to gain real-time insights across three key areas: **Player Performance**, **Match Outcomes**, and **Team Dynamics**.

1. **Player Performance**: The pie chart representing player statistics provides an overview of key metrics such as runs scored, wickets taken, and strike rates:
    - **Runs Scored** (blue) account for a significant portion of the analysis, reflecting how certain players contribute heavily to their team's batting performance.
    - **Wickets Taken** (red) is another major component, indicating the impact of bowlers on match outcomes.
    - **Strike Rate** (yellow) represents the batting efficiency, showing how quickly players accumulate runs.

These insights are crucial for identifying top-performing players and those whose contributions are pivotal to team success. The data highlights which players dominate in specific aspects (batting or bowling), making it easier to focus on star performers.

2. **Match Outcomes**: The pie chart for match outcomes showcases the distribution of wins and losses across teams. Several key insights include:
    - **Team A** (blue) and **Team B** (yellow) dominate the chart, suggesting these teams have higher win rates compared to others.
    - **Other teams** have moderate representation, while a few teams have minimal success, reflecting uneven team performance.

The dominance of certain teams could point to stronger strategies or standout players that consistently contribute to their success. This data helps analysts pinpoint which teams are most successful and which struggle during different IPL seasons.

3. **Team Dynamics**: The bar chart displaying team performance over the seasons shows notable variation across different teams:
    - Teams like **Team C** and **Team D** have the highest number of successful match wins, suggesting they are particularly active and successful in the IPL.
    - Other teams like **Team E** and **Team F** show fewer wins, indicating less activity or fewer successful strategies.

This variation helps understand the performance focus of different teams across the seasons. Certain teams have more consistent success, which might indicate better team composition or management strategies, while others may need improvement.

Insights from the Dashboard

The insights derived from the **Apache Spark-powered analytics dashboard** in Databricks help analyze trends across three critical aspects of IPL data:

1. **Player Performance**: The dominance of specific players in terms of runs scored, wickets taken, and high strike rates helps teams make informed decisions about team selection, training focus, and match strategies.
2. **Match Outcomes**: The distribution of wins and losses gives insights into the overall competitive landscape of the IPL, highlighting dominant teams, underdogs, and potential areas of focus for

improvement.

3. **Team Dynamics**: By analyzing win patterns and team performance over multiple seasons, teams can track their historical progress, identify weaknesses, and re-strategize for future matches.

Application in Big Data Analytics

In the context of **Big Data Analytics**, the IPL data analysis project demonstrates how **Apache Spark** on **Databricks** can handle massive datasets efficiently, with real-time data processing, advanced visualization, and parallel computing capabilities. Key benefits of using Spark in this scenario include:

- **High-Speed Data Processing**: Spark's ability to perform distributed computing across clusters allows for real-time processing of IPL data, enabling swift analysis of match outcomes, player statistics, and other critical metrics.
- **Scalability**: The project can scale effortlessly to handle increasing data volumes as more IPL matches and player statistics are added each season, ensuring continued performance and reliability.
- **Advanced Analytics**: By leveraging **Spark SQL** and **MLlib** in Databricks, advanced queries and machine learning models can be built to predict future match outcomes, player performances, or even team rankings based on historical data trends.

In summary, this **end-to-end IPL Data Engineering Project** leverages **Apache Spark** and **Databricks** to transform raw IPL data into meaningful insights, making it an essential tool for data analysts, team managers, and IPL enthusiasts alike. These insights can guide teams in optimizing performance, crafting winning strategies, and enhancing player development efforts.

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

Conclusion:

The **IPL Data Analysis** project successfully demonstrates the use of **Apache Spark** and **Databricks** to build an end-to-end data engineering pipeline for analyzing IPL match data. By leveraging **Spark** for distributed data processing and **Databricks** for scalable, cloud-based computation, the project showcases how large volumes of IPL data can be efficiently processed, analyzed, and visualized in real-time. The combination of **Spark SQL** for structured data queries, **MLlib** for machine learning, and **Databricks' visualization tools** enables the system to extract valuable insights on player performance, team dynamics, and match outcomes.

The project emphasizes the importance of big data analytics in modern sports analysis, providing decision-makers with real-time insights to improve strategies, monitor team progress, and enhance overall IPL performance. The architecture ensures that even large datasets are processed quickly, enabling teams to stay competitive by using data-driven decisions.

Future Scope:

Looking forward, several enhancements can be implemented to further improve the IPL Data Analysis system:

1. **Machine Learning Integration**: The system can be expanded to incorporate more advanced **machine learning models** to predict player performances, team strategies, and match outcomes. For example, historical player data could be used to forecast future performance, and models could be built to simulate various match scenarios, helping teams optimize strategies based on predicted outcomes.
2. **Real-Time Alerts**: Integrating **real-time alerting mechanisms** would add value by enabling immediate responses to key events during IPL matches. For instance, alerts could be triggered if certain milestones are reached, such as a player achieving a specific

strike rate, or a team surpassing a set run rate. These alerts could help analysts, commentators, or team strategists react in real-time during a match.

3. **Data Retention and Archival**: As more IPL seasons are added, the volume of data will grow exponentially. Implementing **data retention and archival policies** will help manage this data efficiently. Old data could be archived or moved to lower-cost storage solutions while ensuring that relevant data for recent seasons remains accessible for quick querying and analysis.

4. **Advanced Visualization and Reporting**: The dashboard can be enhanced with more **sophisticated visualization tools** to provide deeper insights into trends and key performance indicators (KPIs). Advanced charting features, such as heatmaps, player comparisons, and season-wise performance graphs, could give a more comprehensive view of team and player statistics, helping users identify long-term patterns and outliers more effectively.

5. **Collaborative Analytics**: In the future, integrating collaborative features within Databricks could enable multiple users (data scientists, analysts, and team managers) to work on IPL data analysis together. Shared notebooks and real-time collaboration would allow teams to work cohesively, improving the overall decision-making process.

Conclusion:

The **IPL Data Analysis using Apache Spark and Databricks** project is a comprehensive solution for processing, analyzing, and visualizing large IPL datasets. By leveraging the power of **big data analytics**, the project opens up numerous possibilities for better understanding player performances, match dynamics, and team strategies. With future enhancements, the system can provide even more valuable insights, making it an essential tool for IPL teams and analysts who rely on data-driven strategies to excel in the league.

.