CPSC 531-02
ADVANCED DATABASE MANAGEMENT

PROJECT
CRICKET MATCH ANALYSIS
DECODING THE GAME THROUGH DATA

TEAM MEMBERS
SUMANTH MITTAPALLY (885157511)
ROHIT CHOWDARY YADLA (885146852)

INTRODUCTION

One of the most well-liked and dynamic sports in the world, cricket has changed a lot in the last few years in terms of gameplay as well as analysis and comprehension. Big data technology has made it possible for sports fans and experts to access never-before-seen volumes of data produced during cricket matches. A wealth of information is available for thorough analysis and insights thanks to this data influx, which includes anything from player statistics and match events to ambient conditions.

The current research explores the field of cricket match analysis, using big data's ability to reveal hidden trends, patterns, and performance indicators. Conventional approaches to analyzing cricket matches frequently fail to capture the subtleties and complexity of the game. This gap is filled by big data analytics, which provides a more detailed and nuanced insight of player strategies, team dynamics, and match results.

PROJECT ABSTRACT

Within the constantly changing field of sports analytics, cricket is a dynamic and complex game that presents a multitude of opportunities and problems for in-depth analysis. The goal of this project, "Leveraging Big Data for Comprehensive Cricket Match Analysis," is to use big data's ability to reveal performance indicators, hidden patterns, and trends in the cricket world.

TECHNOLOGY USED

The project will make use of both the Windows and Mac operating systems. For efficient data processing and analysis, it will be built with Google Cloud Platform (GCP) and Spark. Python will be the major programming language, including important libraries such as PySpark, SparkML, and Pandas. The integrated development environment (IDE) will be Jupyter Notebook, which will provide a user-friendly and collaborative platform for coding and experimenting.

FUNCTIONALITIES

1) Dataset Overview

```
match_info.show(5)
```

```
+-------+-------+----------+----------+-----------+----------+-----------+-------------+------+----------+---------
--+-----------+--------------+---------------+--------------------+--------------------+---------------+-----------
---------+
|     id| season|      city|      date|      team1|     team2|toss_winner|toss_decision|result|dl_applied|     winn
er|win_by_runs|win_by_wickets|player_of_match|               venue|             umpire1|        umpire2|
umpire3|
+-------+-------+----------+----------+-----------+----------+-----------+-------------+------+----------+---------
--+-----------+--------------+---------------+--------------------+--------------------+---------------+-----------
---------+
|1389389|2023/24|    Indore|2023/09/24|      India| Australia|  Australia|        field|   D/L|         1|      Ind
ia|         99|             0|        SS Iyer|Holkar Cricket St...|       J Madanagopal|HDPK Dharmasena|KN Ananthap
admana...|
|1336129|   2023|Nottingham|2023/09/23|    England|   Ireland|    Ireland|        field|normal|         0|    Engla
nd|         48|             0|       WG Jacks|Trent Bridge, Not...|           DJ Millns|      RJ Tucker|          P
R Reiffel|
|1395701|   2023|     Dhaka|2023/09/23|New Zealand|Bangladesh|New Zealand|          bat|normal|         0|New Zeala
nd|         86|             0|       IS Sodhi|Shere Bangla Nati...|          M Erasmus|   Sharfuddoula|         Ni
tin Menon|
|1389388|2023/24|Chandigarh|2023/09/22|  Australia|     India|      India|        field|normal|         0|      Ind
ia|          0|             5| Mohammed Shami|Punjab Cricket As...|KN Ananthapadmana...|HDPK Dharmasena|       J Ma
danagopal|
|1395700|   2023|     Dhaka|2023/09/21|New Zealand|Bangladesh| Bangladesh|        field|normal|         0|       nu
ll|          0|             0|           null|Shere Bangla Nati...|         Nitin Menon|   Sharfuddoula|
M Erasmus|
```

```
match_data.show(5)
```

```
+--------+-------+-------------------+--------------------+-------+----+------------+------------+----------+------
------+----------+------------+------+-----+-------+----+-------+-------+-----------+----------------+-------------
----+----------------------+------------+
|match_id| season|         start_date|               venue|innings|ball|batting_team|bowling_team|   striker| non_s
triker|    bowler|runs_off_bat|extras|wides|noballs|byes|legbyes|penalty|wicket_type|player_dismissed|other_wicket_
type|other_player_dismissed|cricsheet_id|
+--------+-------+-------------------+--------------------+-------+----+------------+------------+----------+------
------+----------+------------+------+-----+-------+----+-------+-------+-----------+----------------+-------------
----+----------------------+------------+
| 1389389|2023/24|2023-09-24 00:00:00|Holkar Cricket St...|      1| 0.1|       India|   Australia|RD Gaikwad|Shubma
n Gill|SH Johnson|           4|     0| null|   null|null|   null|   null|       null|            null|
null|                  null|     1389389|
| 1389389|2023/24|2023-09-24 00:00:00|Holkar Cricket St...|      1| 0.2|       India|   Australia|RD Gaikwad|Shubma
n Gill|SH Johnson|           0|     0| null|   null|null|   null|   null|       null|            null|
null|                  null|     1389389|
| 1389389|2023/24|2023-09-24 00:00:00|Holkar Cricket St...|      1| 0.3|       India|   Australia|RD Gaikwad|Shubma
n Gill|SH Johnson|           4|     0| null|   null|null|   null|   null|       null|            null|
null|                  null|     1389389|
| 1389389|2023/24|2023-09-24 00:00:00|Holkar Cricket St...|      1| 0.4|       India|   Australia|RD Gaikwad|Shubma
n Gill|SH Johnson|           0|     0| null|   null|null|   null|   null|       null|            null|
null|                  null|     1389389|
```

The project involves collecting data from different matches on Kaggle from 2002 until 2023. There are several examples in the dataset. Analyzing the data and giving the team and players useful information is the main objective. By looking through this dataset, the study intends to gain understanding and assist the team in selecting the best player and location.

An overview of the dataset's details, including the number of rows, columns, and data types for each column, is returned by the code dataset.info(). It is employed to obtain a concise synopsis of the dataset's attributes and organization.

dataset.info() yields details like these:

1. The total number of rows (entries) in the dataset.
2. The quantity of columns and each one's name.
3. The quantity of items in each column that are not zero.
4. The data type of each column, such as text, datetime, float, or integer.
5. Extra information about memory usage.

2) Data Cleaning

When referring to a dataset, data cleaning is the act of identifying and fixing errors in the dataset to improve its dependability, consistency, and quality. Data cleaning techniques can be applied to address inconsistent values, handle missing values, ensure consistent formatting, find and address outliers, correct data types, resolve conflicting values, and verify data.
On the project dataset, the following data cleaning procedures might be applied:

a. Data Imputation – Missing values

Finding any missing values in the dataset is the first step in the pre-processing process. Finding columns in the project dataset that lack important data like venue or season is the first step in handling missing values. One approach was to remove the rows that have missing values; this might reduce the overall size of the dataset while guaranteeing that the remaining data is complete. In order to fill in the blanks, missing values are imputed using techniques like mean or median imputation, which use the center or average value of the available data. Imputation ensures a comprehensive dataset for research while preserving important data. The amount and importance of missing values, as well as any potential effects on further research and interpretations, all play a role in the selection between removal and imputation.

b. Removing Duplicates

We looked over the dataset carefully to look for duplicate entries. We specifically looked at season, batting team, and bowling team to ensure data integrity. We looked through the records and found and removed any duplicates. By eliminating these duplicates, we made sure that every possible combination of season, batting team, and bowling team was only included once in the dataset. The removal of duplicate entries was essential to preserving the accuracy and dependability of our data since they could introduce biases and distort the analytical findings. We ensured that the dataset was free of duplicate data.

c. Consistent Formatting

We were aware of the importance of the dataset's consistent presentation. To do this, we focused on standardizing the formatting of columns such venue, player of the match, and seasons. We meticulously reviewed each input, ensuring that any disparities in abbreviations or capitalization were fixed. By standardizing the formatting, we eliminated inconsistencies and created consistency throughout the dataset. This action was essential for enhancing data clarity and enabling accurate analysis. Working together, we painstakingly adjusted the formatting as needed, producing a dataset with consistently formatted items. As a result, it was easier to evaluate the data and extract insightful information from it.

d. Outliner Detection and Handling

We understood that identifying and handling the dataset's outliers was necessary. Using thorough research, we were able to identify whether these extraordinary figures were genuine data points or incorrect entries. We reduced the impact of outliers on analytical findings, when appropriate, by eliminating them or making the necessary changes. Because of our joint efforts, the dataset was assured to be free of outlier distortions, allowing for the generation of more precise and reliable insights from the data.

e. Data Validation and Type Correction

Data validation and ensuring the integrity of the dataset were our top priorities. We worked together to conduct data validation tests to guarantee accuracy and consistency. For instance, in order to identify any abnormalities, we cross-validated data across pertinent columns and made sure that records were within reasonable bounds. In order to make sure that numerical values were stored as numeric data types rather than text or category variables, we also carefully examined the data types of the columns. The reliability of the dataset was increased by our joint effort to carry out these tests and change the data types as necessary, which raised the caliber and precision of upcoming analyses and interpretations.

The dataset will be more dependable, consistent, and of higher quality if various data cleaning techniques are used on it. This

makes it possible for research and insights on cricket to be more accurate.

3) Data Analysis

In our study, we created trustworthy visualization models using PySpark, a current technology. We successfully processed and analyzed enormous volumes of data from the dataset by utilizing Spark's distributed computing capabilities. We will use PySpark machine learning components to build numerous regression approaches. These models provide the teams with the best insights for improved team growth by precisely analyzing.

4) Google Cloud Platform

In order to enhance our data analysis on the dataset for our project, we made use of Google Cloud Platform (GCP) and specifically Cloud Dataproc. Thanks to GCP, we had access to a robust and expandable infrastructure that made it possible for us to process and analyze cricket match data effectively. With Cloud Dataproc, we constructed single-node and three-node clusters to handle enormous volumes of data. Apache Spark is installed on these clusters, which enables us to perform distributed data processing and analysis. By spreading the burden across multiple nodes, we were able to get faster execution and improved performance.

5) DATA VISUALIZATION AND EXPLORATION

In our project, the Google Cloud Platform (GCP) assisted with data visualization for the analysis of cricket matches.

Here's how we carried it out:

1. Our main method of data storage was uploading the match info and match data datasets to Google Cloud Storage. The dataset became easier to access and handle as a result.

2. To build our Spark project, we used GCP's Dataproc service. We configured a cluster in Dataproc with a single master node. The processing of large amounts of data was made efficient by this networked computer system.

3. To create a collaborative and interactive setting for data processing and visualization, we built a PySpark Jupyter notebook inside the cluster. The dataset was read from cloud storage using Dataproc, which made it simple to link cloud storage to the Hadoop Distributed File System (HDFS).

4. To extract crucial data on cricket analysis, we used PySpark SQL to run queries on the dataset. The query results were then saved to the cloud for easy access and further examination.

By applying these techniques, we were able to analyze the data and gain important insights. We were able to display and discuss trends, patterns, and statistics in an effective and scalable manner

thanks to the utilization of GCP, Dataproc, and PySpark SQL to establish a platform for data visualization. This visual assistance enhanced our understanding of the dataset and helped players and teams make wise decisions.

ARCHITECTURE AND DESIGN

To ensure consistent and dependable data for further study, we used PySpark and Spark to clean and analyze the dataset for our project.

We started by installing Spark and setting up a Spark Session, which functions as a gateway for using the Dataset and Data Frame API for Spark programming. We learned how to create Data Frames, register them as tables, and query the data using SQL during this session. Using the builder pattern method builder() and the getOrCreate() function, we were able to successfully establish a Spark session. Our CSV-formatted dataset was then put into a Data Frame for further processing.

The data was then cleaned using PySpark, which was a crucial step in guaranteeing the accuracy and consistency of the dataset. Using the PIP Install PySpark command, we installed PySpark and made use of its features to remove errors, duplicates, and superfluous data from our dataset. By generating Data Frames, registering them as tables, and running SQL queries, we were able to successfully clean the data. Using a clean dataset, we performed several analyses on the data, such as creating data frames and evaluating the data with SQL queries.

For analysis, we put the cleaned dataset into a Google Cloud Storage bucket. This made it possible for us to safely store and retrieve the dataset, enabling effective cloud-based data analysis. We used PySpark's and Spark's capabilities to clean and analyze the dataset. With the use of these technologies, we were able to decrease discrepancies, check the quality of the data, and gain important insights into cricket patterns. Using Spark, PySpark, and cloud storage together, we can quickly and scalable analyze large amounts of data, which improves decision-making.

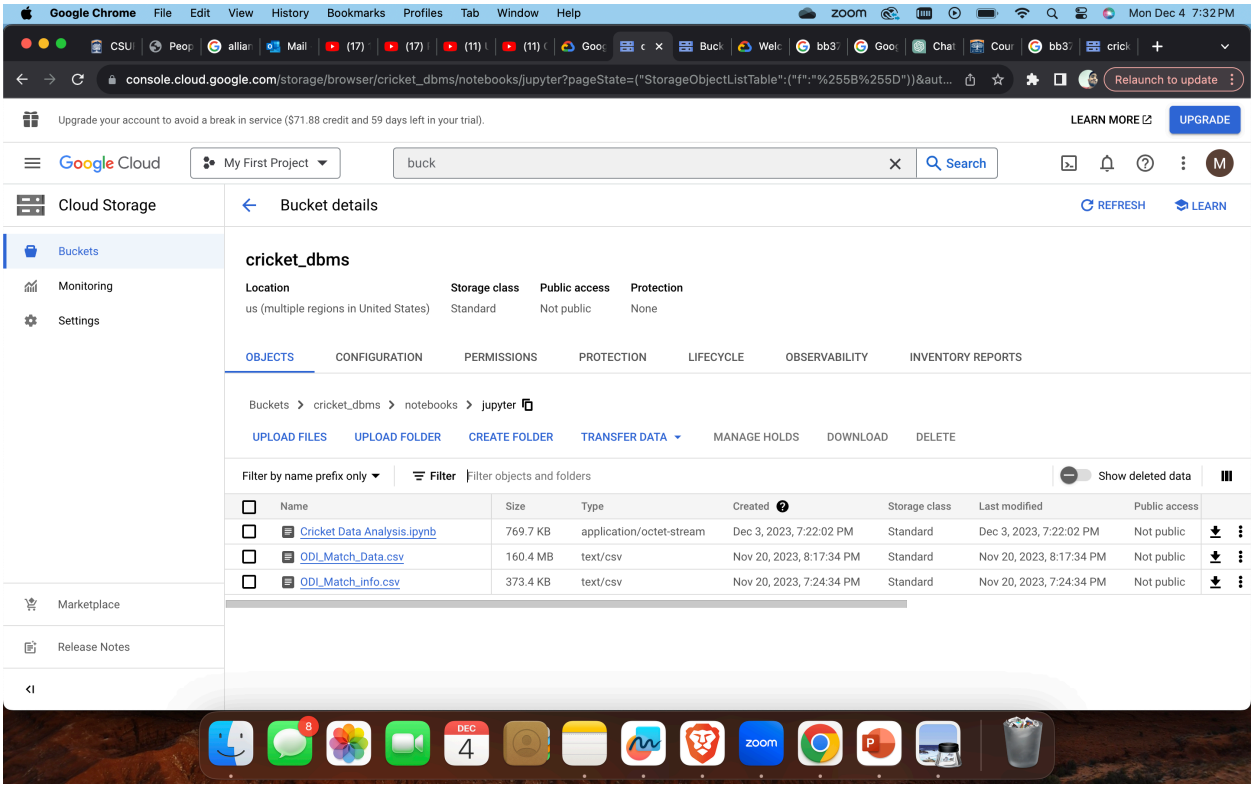


Fig. Bucket Details

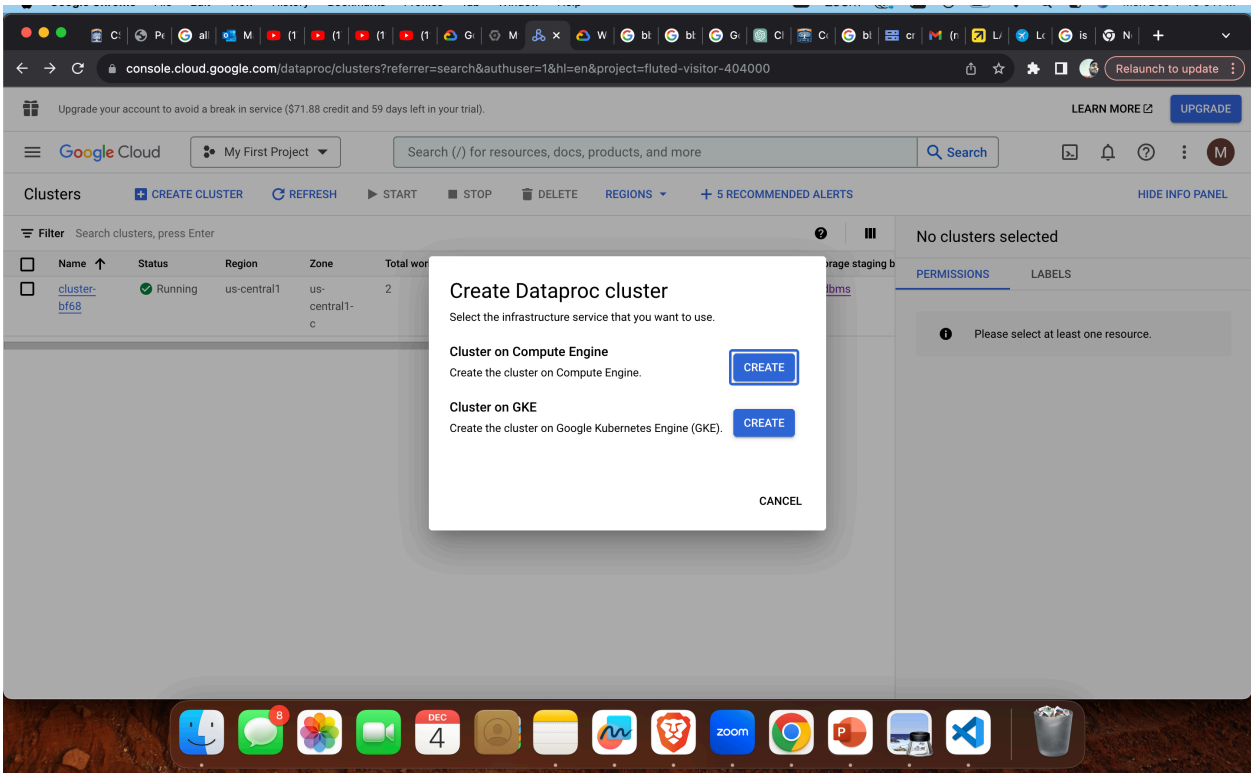
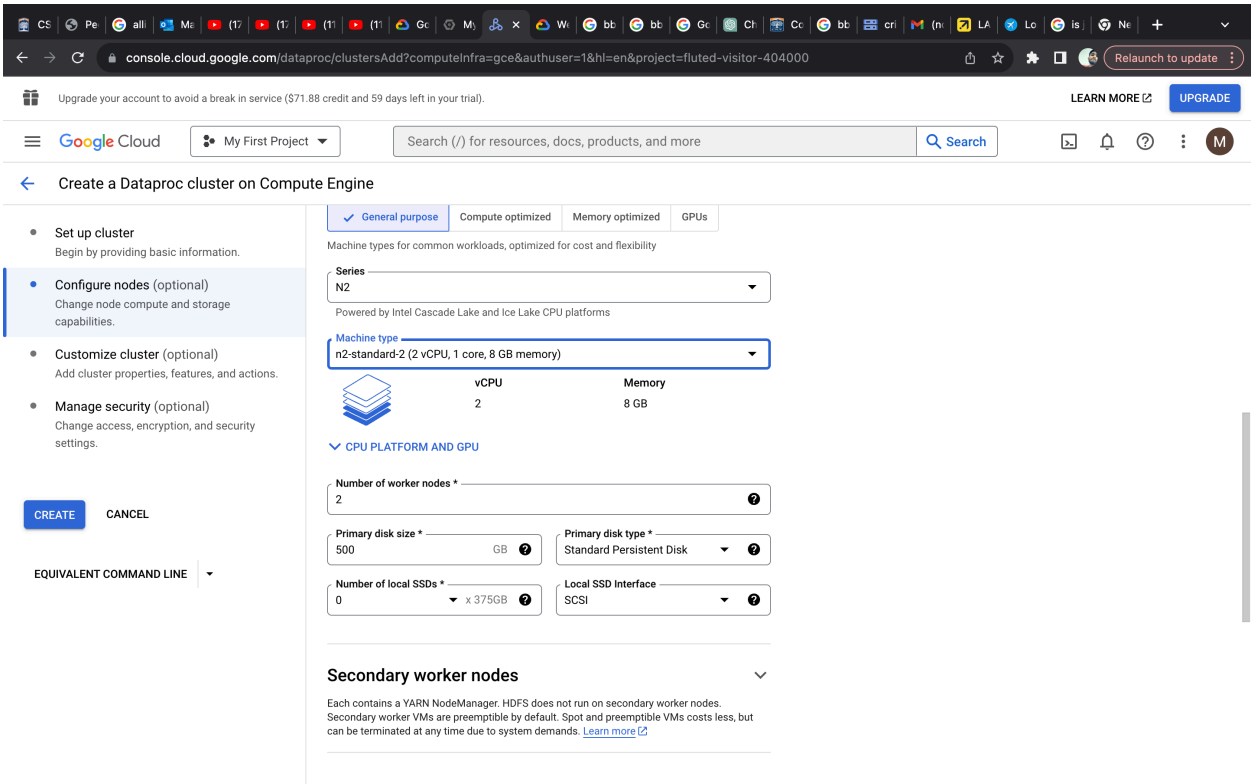


Fig. Cluster creation



DEPLOYMENT INSTRUCTIONS

We went through a lot of procedures in our project to set up the infrastructure and instruments required for efficient data visualization and analysis.

Our data processing and analysis processes were based on Apache Spark, which we first downloaded and installed from its official website. The framework for distributed computing allowed us to effectively handle large volumes of data.

After that, we installed the Python components and configured Jupyter Notebook to create an interactive and user-friendly writing environment. We were able to work with Spark and Python more readily because to this interface, which made data handling and analysis easier.

To take use of cloud computing, we set up a cluster in the Google Cloud Platform (GCP) using Dataproc. This provided us with an effective and scalable computing system to handle our dataset.

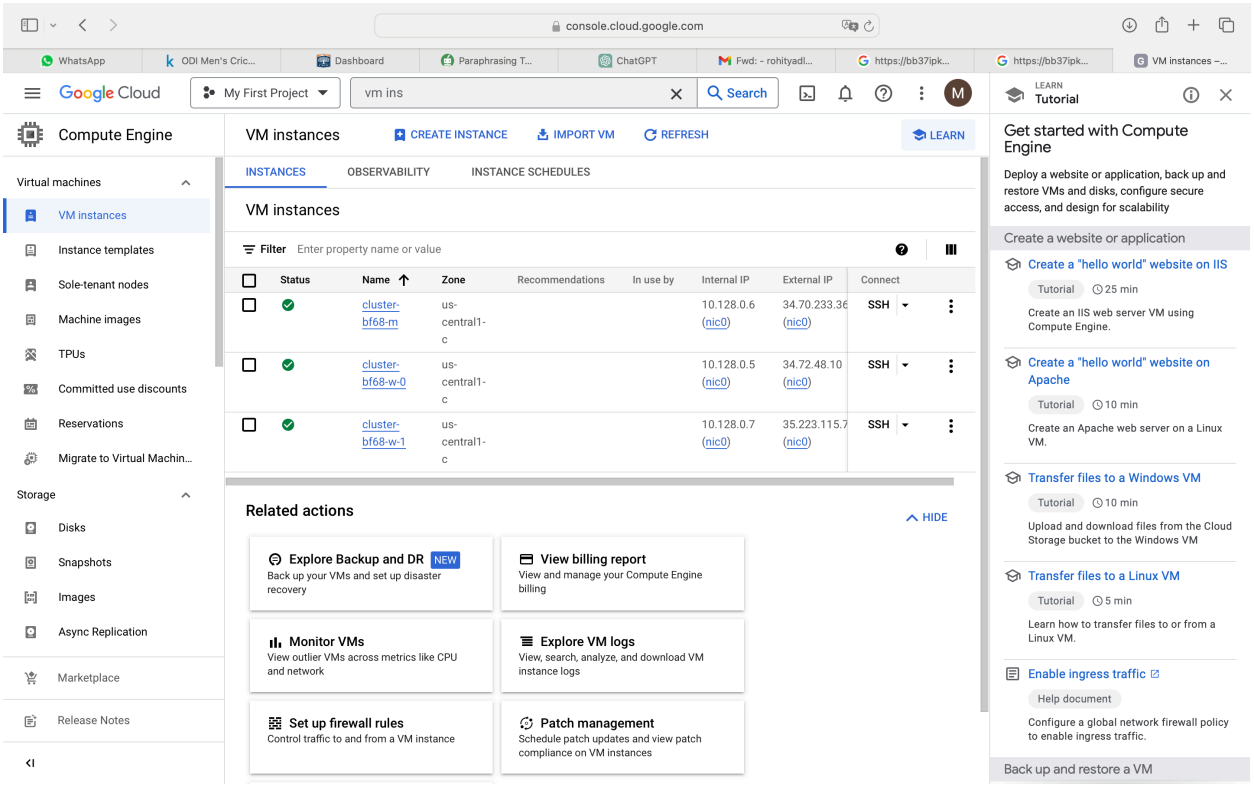
In advance of our data analysis, we installed the necessary libraries and dependencies to support our code. To securely store our dataset, we also built a Google Cloud Storage bucket.

Lastly, we investigated our data and learned new things from it. In our Jupyter Notebook environment, we were able to do SQL queries, evaluate data, and create pertinent visualizations.

During these stages, we created a strong framework for data analysis and visualization that helped us make informed decisions and obtain insightful knowledge.

STEPS TO RUN

Step 1: We opened the Google Cloud Platform (GCP) application to start our project. After we were allowed access to our GCP account, we created a new project specifically tailored to our study.



Following the project's successful creation, we set up a virtual machine (VM) instance inside the GCP architecture. Our computer environment was the virtual machine instance, which provided us with the resources and abilities required to do our

data processing tasks. During the VM instance generation process, several parameters were defined, including operating system, disk size, and machine type. These criteria were selected taking into account the demands and anticipated workload of our project.

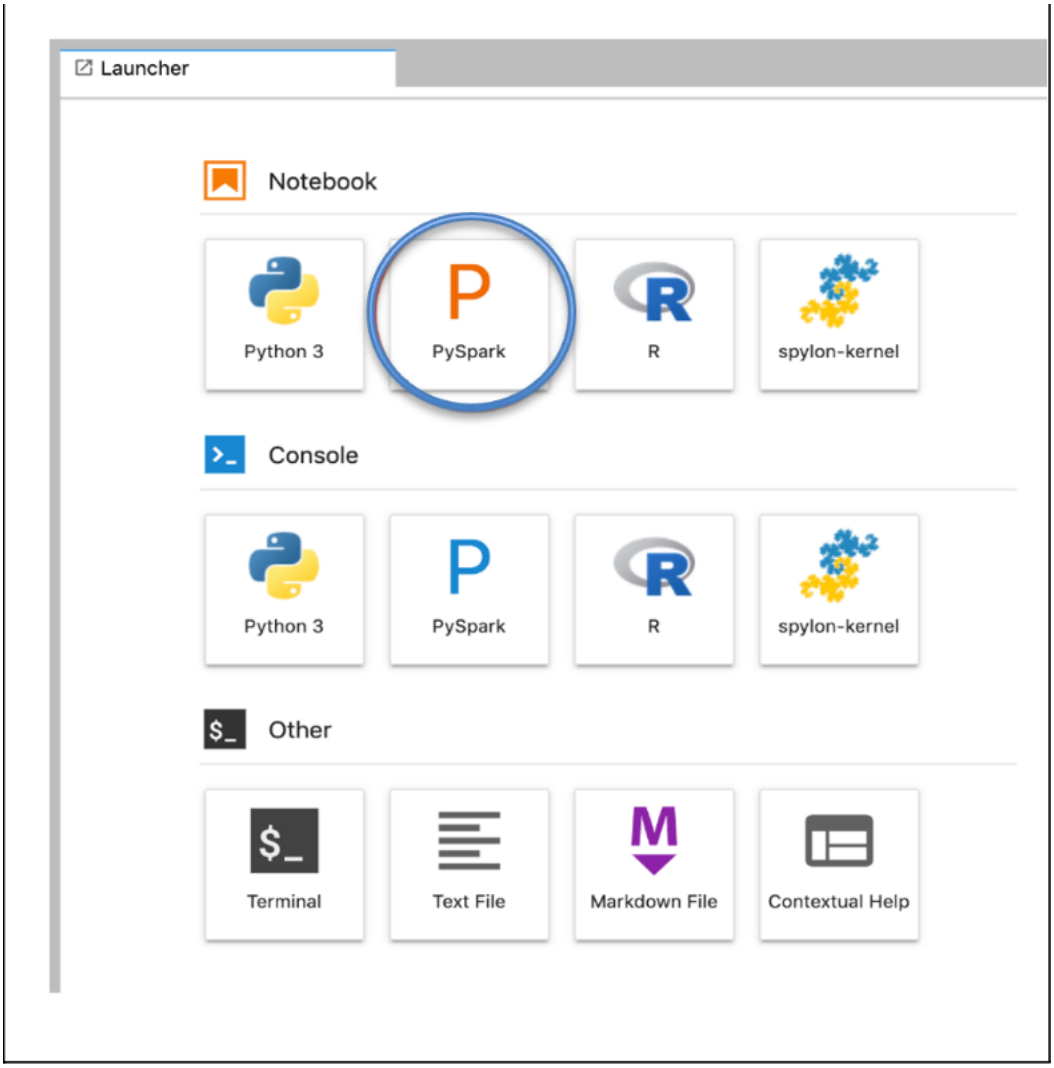
Once the virtual machine (VM) instance was provisioned correctly, we were able to log in and use it remotely. We were able to install software, run instructions for data processing and analysis, and remotely manage the virtual machine.

Step 2: We changed the firewall settings and started the virtual machine instance from the Google Cloud Platform (GCP) to ensure sufficient network connectivity and security.

We first proceeded to the networking section and the GCP interface. We located the firewall rules within the networking configuration and made the necessary adjustments. After updating the firewall rules, we started the virtual machine instance. With the boot-up procedure initiated, the virtual machine became operational and usable. We monitored the instance's status and verified that it had started successfully.

We launched the virtual machine instance and changed the firewall settings to ensure that necessary network access was allowed while maintaining appropriate security precautions. This step gave us access to a functional computing environment so we could perform our analytical tasks.

Step 3: To set up the development environment for our project, we installed Jupyter Notebook and other necessary apps via an SSH terminal.



We executed the appropriate steps to install Jupyter Notebook after successfully establishing an SSH connection to the virtual machine instance. This involved installing the most latest version by using package managers such as pip or conda. We also installed extra packages and libraries that were required for our data processing and analysis tasks. These packages may contain

NumPy, Pandas, Matplotlib, Seaborn, and any other dependencies pertinent to our project requirements.

To ensure that the installs were successful, we closely followed the instructions provided by the package managers throughout the installation process. Installing Jupyter Notebook and other necessary apps within the virtual machine (VM) instance allowed us to establish a reliable development environment. With the Jupyter Notebook interface, this environment allowed us to quickly build, run, and collaborate on code, enabling seamless integration.

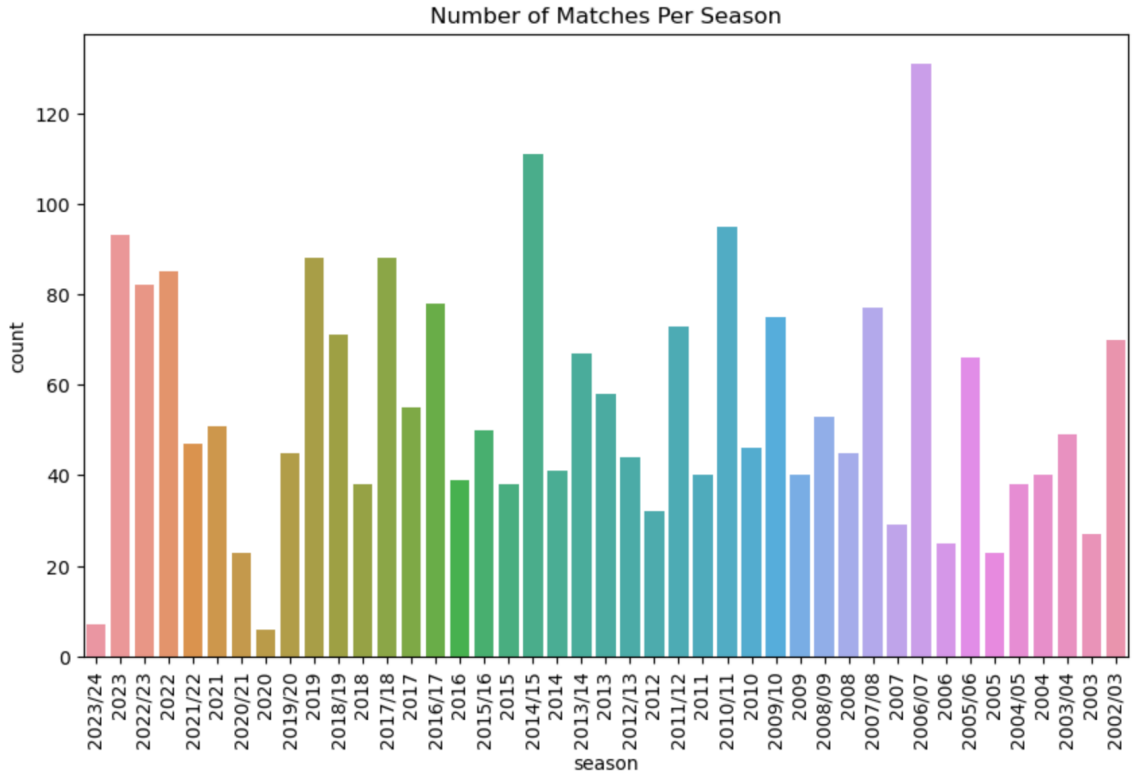
Step 4: We started Jupyter Notebook and set it up to run our data visualization scripts after setting up the virtual machine server.

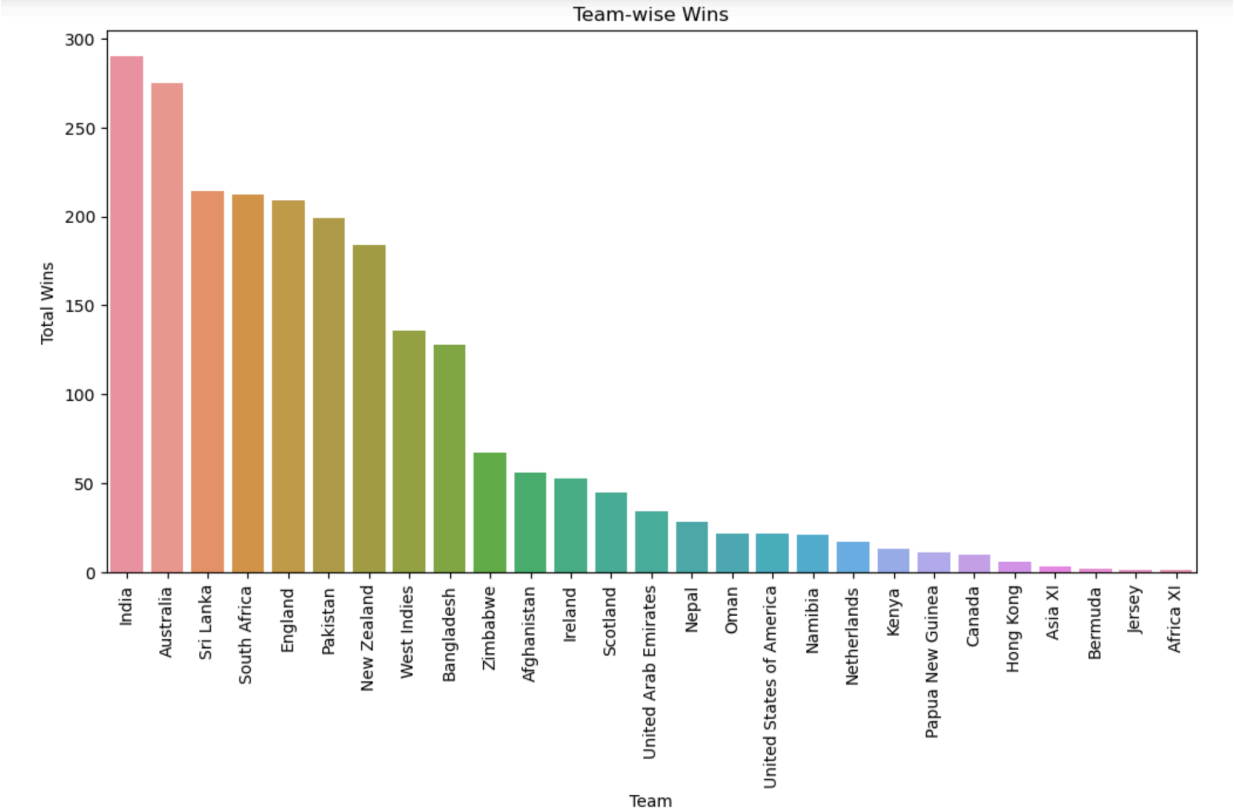
Using the SSH terminal, we typed the appropriate command, specifying the required port and any extra settings, to launch the Jupyter Notebook server. This made it possible for us to view the Jupyter Notebook interface in a web browser. We opened the Jupyter Notebook interface, 22 and then navigated to the directory containing our data visualization code in our.py file. Jupyter Notebook provided an interactive and intuitive interface for editing and executing the code when we opened the file in it. Using the open.py file, we can create meaningful plots, charts, and graphs based on our statistics using visualization programs like Matplotlib, Seaborn, or any other. By executing our data visualization scripts in Jupyter Notebook, we were able to create visual representations of the data, which helped us to gain insightful knowledge and clearly communicate our findings.

GIT LINK:

https://github.com/Sumanth2399/Advance_Database_Management_final_project

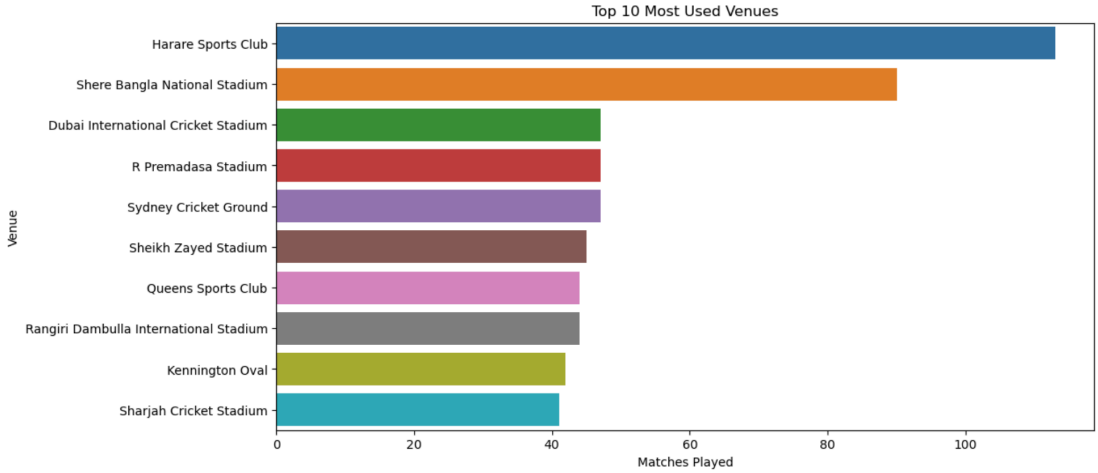
OUTPUT


Number of Matches Per Season
count
season
0
20
40
60
80
100
120
2023/24
2023
2022/23
2022
2021/22
2021
2020/21
2020
2019/20
2019
2018/19
2018
2017/18
2017
2016/17
2016
2015/16
2015
2014/15
2014
2013/14
2013
2012/13
2012
2011/12
2011
2010/11
2010
2009/10
2009
2008/09
2008
2007/08
2007
2006/07
2006
2005/06
2005
2004/05
2004
2003/04
2003
2002/03


Team-wise Wins
Total Wins
Team
0
50
100
150
200
250
300
India
Australia
Sri Lanka
South Africa
England
Pakistan
New Zealand
West Indies
Bangladesh
Zimbabwe
Afghanistan
Ireland
Scotland
United Arab Emirates
Nepal
Oman
United States of America
Namibia
Netherlands
Kenya
Papua New Guinea
Canada
Hong Kong
Asia XI
Bermuda
Jersey
Africa XI

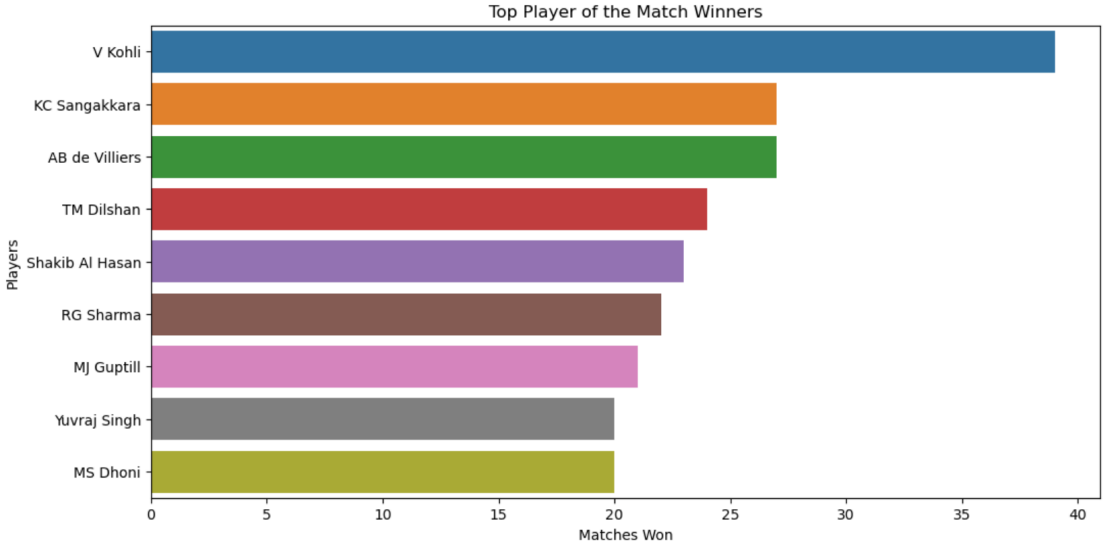
In [24]:
```
#most frequent used venues
top_venues = match_info.groupBy('venue').count().sort('count', ascending=False).limit(10)

#plot the data
plt.figure(figsize=(12,6))
sns.barplot(data=top_venues.toPandas(), x='count', y='venue')
plt.title('Top 10 Most Used Venues')
plt.xlabel('Matches Played')
plt.ylabel('Venue')
```

Out[24]: Text(0, 0.5, 'Venue')



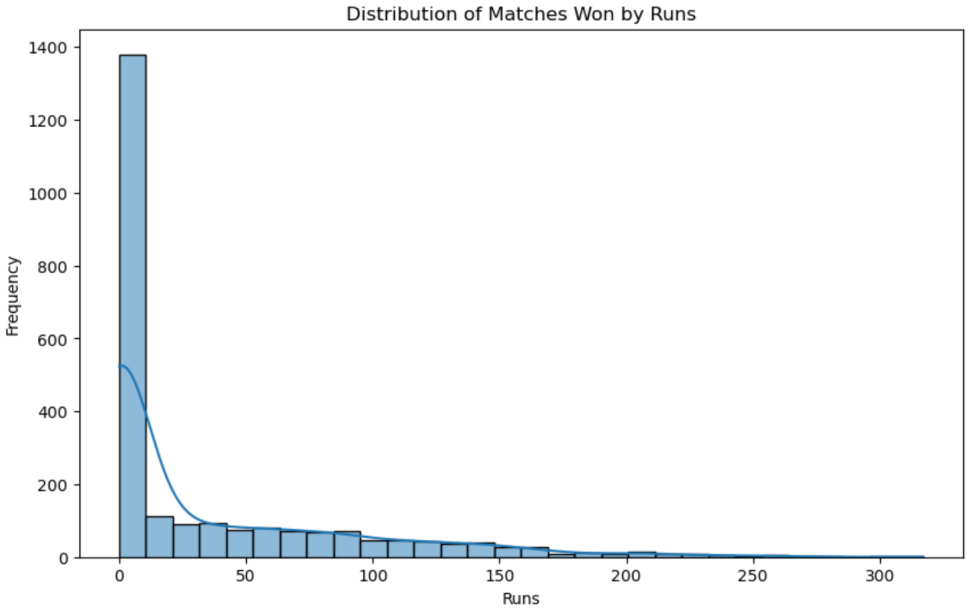
In [25]:
```
#top player of the match winners
top_players = match_info.groupBy('player_of_match').count().sort('count', ascending=False).limit(10)

#plot the data
plt.figure(figsize=(12,6))
sns.barplot(data=top_players.toPandas(), x='count', y='player_of_match')
plt.title('Top Player of the Match Winners')
plt.xlabel('Matches Won')
plt.ylabel('Players')
```

Out[25]: Text(0, 0.5, 'Players')



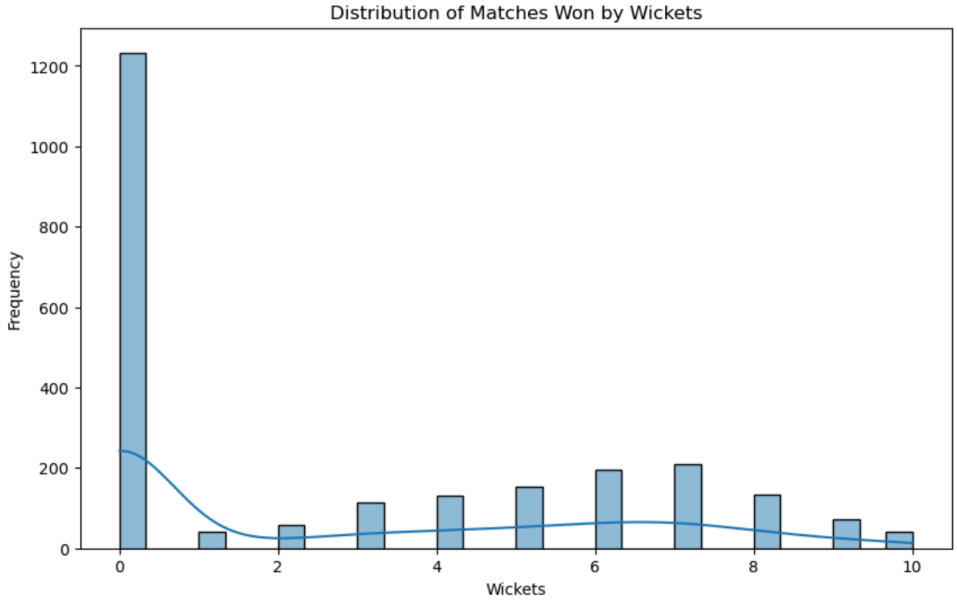
In [26]:
```
#match outcomes
plt.figure(figsize=(10,6))
sns.histplot(data= match_info.toPandas(), x = 'win_by_runs', bins=30, kde=True)
plt.title('Distribution of Matches Won by Runs')
plt.xlabel('Runs')
plt.ylabel('Frequency')
```

Out[26]:
```
Text(0, 0.5, 'Frequency')
```



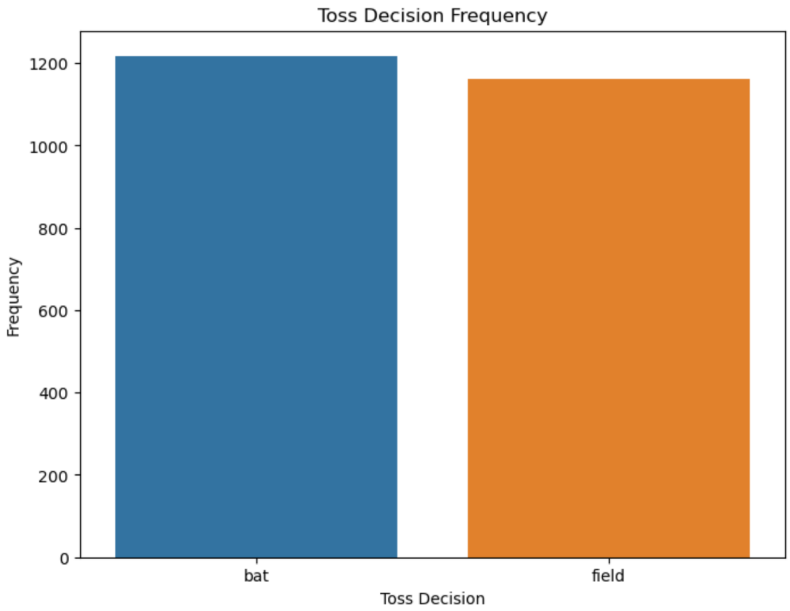
In [27]:
```
plt.figure(figsize=(10,6))
sns.histplot(data= match_info.toPandas(), x = 'win_by_wickets', bins=30, kde=True)
plt.title('Distribution of Matches Won by Wickets')
plt.xlabel('Wickets')
plt.ylabel('Frequency')
```

Out[27]:
```
Text(0, 0.5, 'Frequency')
```



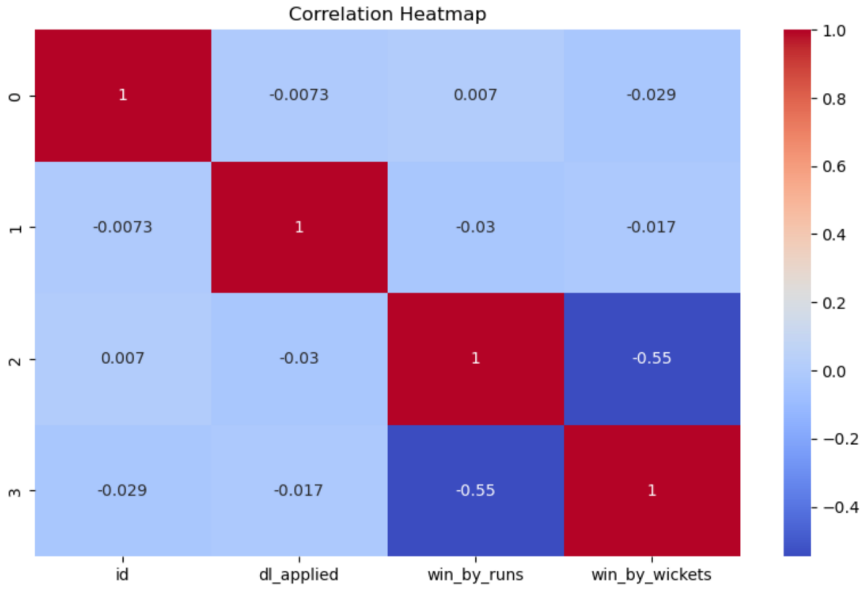
In [28]:
```
#toss decision
toss_decision = match_info.groupBy('toss_decision').count().sort('count', ascending=False)

#plot the data
plt.figure(figsize=(8,6))
sns.barplot(data=toss_decision.toPandas(), x='toss_decision', y='count')
plt.title('Toss Decision Frequency')
plt.xlabel('Toss Decision')
plt.ylabel('Frequency')
```

Out[28]:
```
Text(0, 0.5, 'Frequency')
```



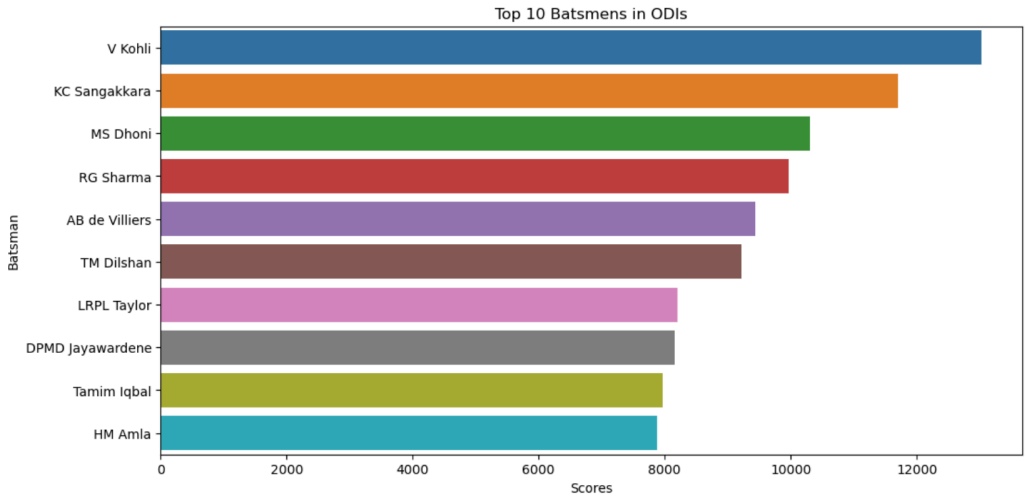
In [31]:
```
plt.figure(figsize=(10,6))
sns.heatmap(pdf, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
```

Out[31]:
```
Text(0.5, 1.0, 'Correlation Heatmap')
```



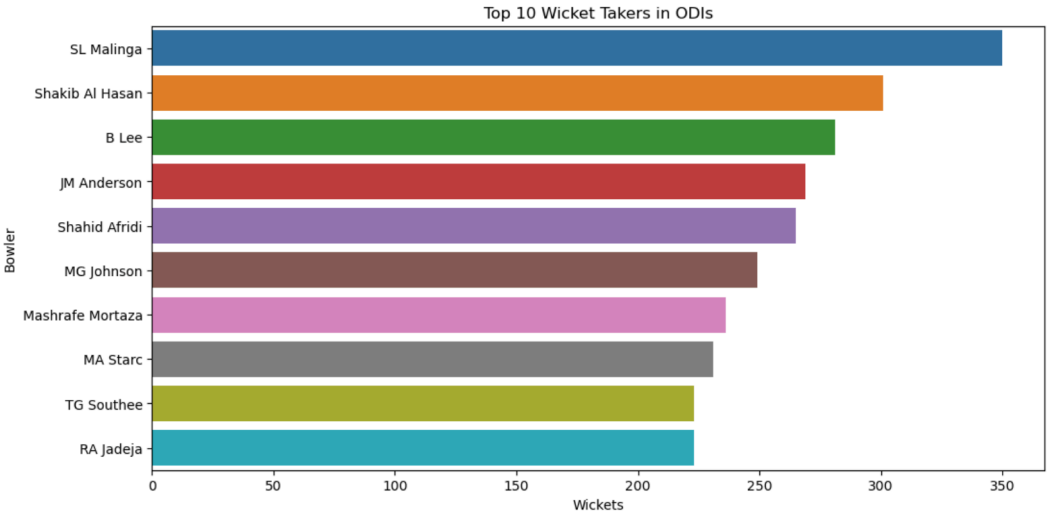
In [33]:
```
from pyspark.sql.functions import sum,avg,max

#identifying top 10 run scorers
batsmen_total_runs = match_data.groupBy("striker").agg(sum("runs_off_bat").alias("sum")).sort('sum', ascending=False

#plot the data
plt.figure(figsize=(12,6))
sns.barplot(data = batsmen_total_runs.toPandas(), x='sum', y='striker')
plt.title('Top 10 Batsmens in ODIs')
plt.xlabel('Scores')
plt.ylabel('Batsman')
```

Out[33]: Text(0, 0.5, 'Batsman')



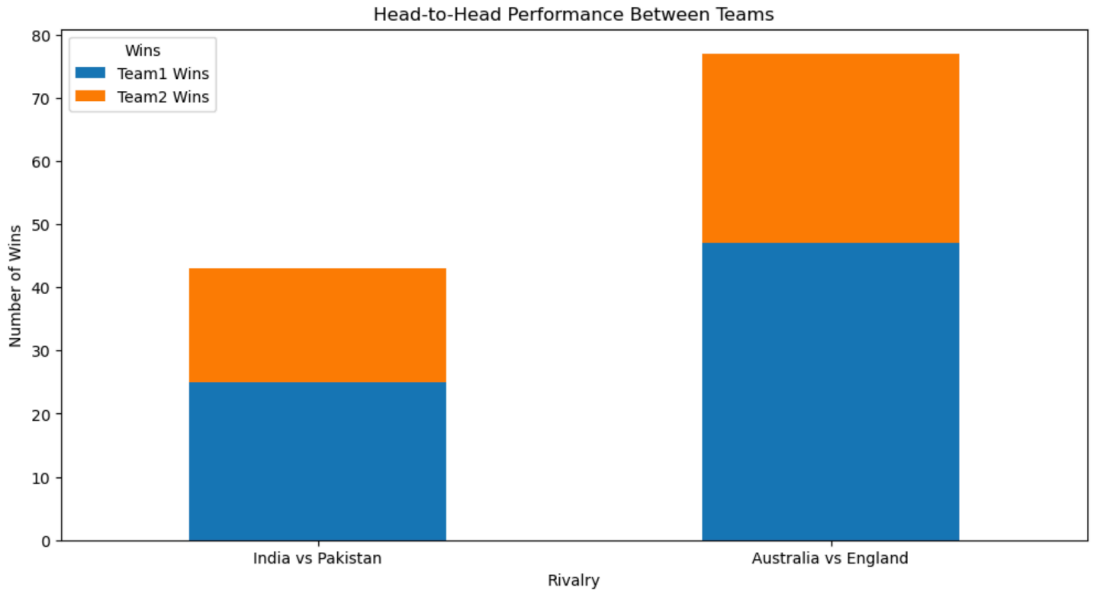
In [34]:
```
from pyspark.sql.functions import sum,avg,max

# filter the not null wicket_type data
not_null_wicket_type = match_data.na.drop(subset=["wicket_type"])

#identifying top 10 wicket takers
top_bowlers_wickets = not_null_wicket_type.groupBy("bowler").agg(count("wicket_type").alias("count")).sort('count',

#plot the data
plt.figure(figsize=(12,6))
sns.barplot(data = top_bowlers_wickets.toPandas(), x='count', y='bowler')
plt.title('Top 10 Wicket Takers in ODIs')
plt.xlabel('Wickets')
plt.ylabel('Bowler')
```

Out[34]: Text(0, 0.5, 'Bowler')



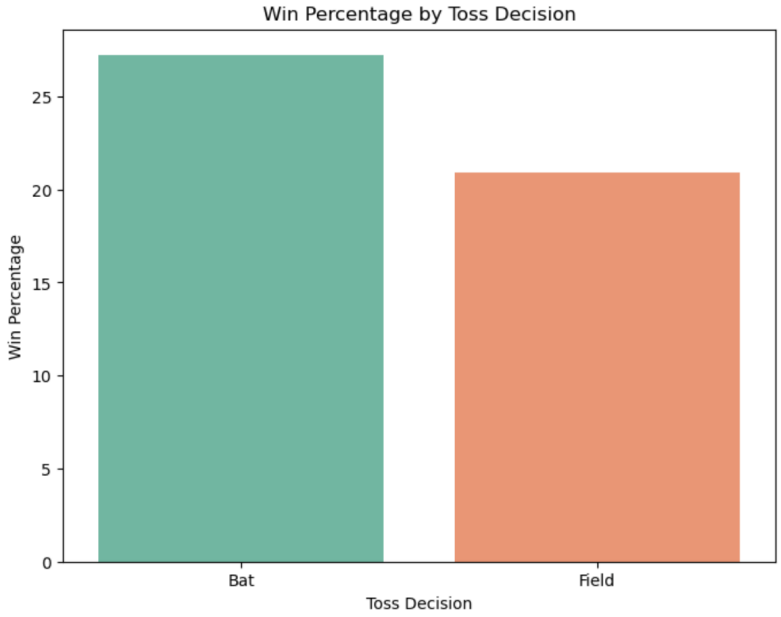
In [47]:
```
rivarly_df[['Team1 Wins', 'Team2 Wins']].plot(kind='bar', stacked=True, figsize=(12,6))
plt.title('Head-to-Head Performance Between Teams')
plt.xlabel('Rivalry')
plt.ylabel('Number of Wins')
plt.xticks(rotation=0)
plt.legend(title='Wins', loc='best')
```

Out[47]:
```
<matplotlib.legend.Legend at 0x7feaffd28340>
```



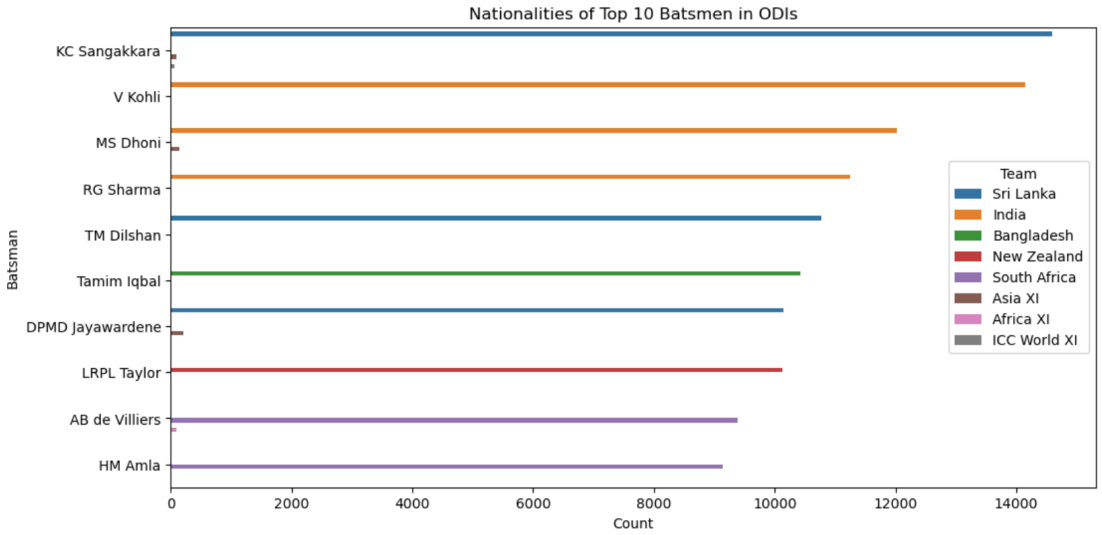
In [51]:
```
plt.figure(figsize=(8,6))
sns.barplot(x=['Bat','Field'], y=[bat_win_percentage, field_win_percentage], palette='Set2')
plt.title('Win Percentage by Toss Decision')
plt.xlabel('Toss Decision')
plt.ylabel('Win Percentage')
```

Out[51]:
```
Text(0, 0.5, 'Win Percentage')
```



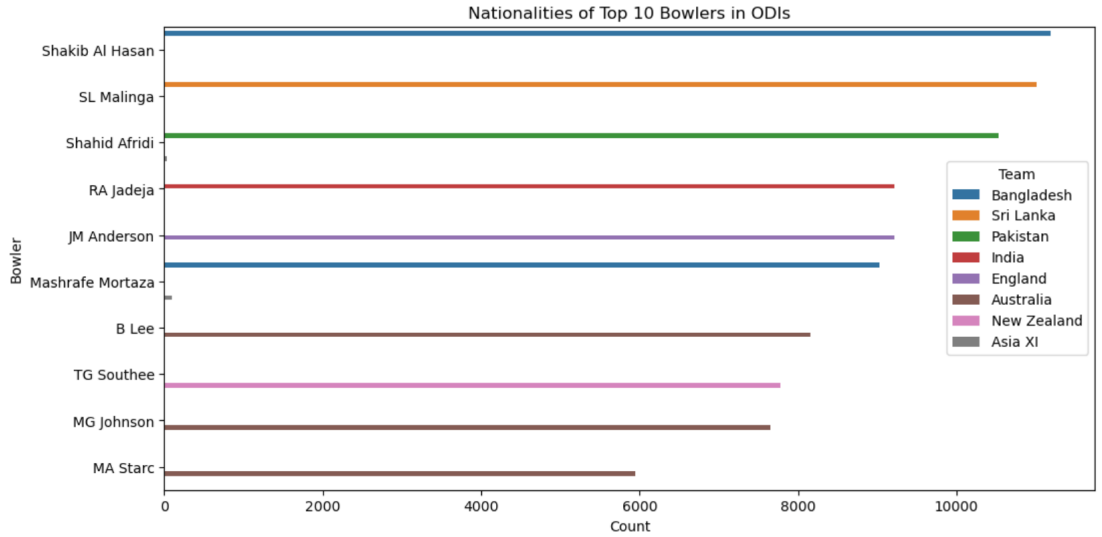
In [40]:
```
plt.figure(figsize=(12,6))
sns.barplot(x='count', y='striker', hue='batting_team', data=nationalities_batsmen.toPandas())
plt.title('Nationalities of Top 10 Batsmen in ODIs')
plt.xlabel('Count')
plt.ylabel('Batsman')
plt.legend(title='Team', loc='best')
```

Out[40]: <matplotlib.legend.Legend at 0x7feb10048040>



In [41]:
```
plt.figure(figsize=(12,6))
sns.barplot(x='count', y='bowler', hue='bowling_team', data=nationalities_bowlers.toPandas())
plt.title('Nationalities of Top 10 Bowlers in ODIs')
plt.xlabel('Count')
plt.ylabel('Bowler')
plt.legend(title='Team', loc='best')
```

Out[41]: <matplotlib.legend.Legend at 0x7feb12e38400>



CHALLENGES

We experienced various challenges during our project:

1. **Data quality and consistency:** By fixing issues including missing data, erroneous entries, and formatting or labeling variations, we were able to ensure the dataset's quality and consistency. Thorough attention to detail was necessary during the data cleaning and verification process to guarantee the study's quality and dependability.

2. **Managing enormous volumes of data:** The cricket dataset we worked with was huge and included information from many teams, venues, and seasons. Due to computational and resource constraints brought about by processing and analyzing such a large amount of data, we had to optimize our procedures and make use of technologies like Spark for efficient data processing.

3. **Data integration and compatibility:** Due to the diverse sources and formats of the data in the dataset, data harmonization and integration were required. We had to convert, clean, and standardize the data, which required cautious handling and close attention to detail, to ensure compatibility for review.

4. **Interpretation and communication:** To analyze cricket data, insightful conclusions must be drawn and effectively communicated to teams and players. We had to use data visualization tools and effective communication strategies to convey the complex analytical results in a clear and

understandable way to ensure that our findings were applicable and significant.

CONCLUSION

Finally, our project's main goal is to analyze cricket data and identify trends in the sport by utilizing advanced technologies like Spark, Google Cloud Platform, and machine learning algorithms. Through data analysis and visualization, we were able to take valuable information out of the dataset.

The results of our initiative will have significant implications for improved player scouting, planning, and decision-making within the cricket industry. Through careful strategy planning, teams may increase productivity, make better decisions, and make fewer mistakes. The study demonstrated how combining data analysis, machine learning, and data visualization may produce insightful findings and reliable conclusions.

Subsequent research endeavors may focus on expanding the scope of the study, integrating more data sources, and exploring sophisticated techniques to enhance planning and foster stronger team dynamics. All things considered, our experience proved the usefulness of data-driven methods in the cricket industry. By harnessing the potential of contemporary data analytics, we can create a future that is more efficient, profitable, and sustainable.

FUTURE WORK

The main goal of our study was to use advanced analytics approaches to forecast trends in cricket. Going forward, a number of other study topics might be looked at in order to enhance our projections and add to the results. In order to improve the precision of our forecasts, we might first explore machine learning models that are more potent than regression. Complicated patterns and nonlinear linkages in a dataset can be captured with the help of deep learning and ensemble techniques. This may include building real-time analytics pipelines and integrating streaming data to provide current information for decision-making.

It may be necessary to broaden the research in the future to incorporate sustainability criteria. Future work on our project will primarily focus on improving our models, including real-time data, looking into sustainability issues, and fostering greater collaboration within the sports industry. We may improve our predictions, support the cricket industry, and improve cricket as a sport if we keep looking at these areas.