

CIS 602 Big Data Analytics
Fall'23
Project Summary

Olympic Data Analytics

Group-18
Sai Alekhya Ravi (39)
Preetham Sai Gattamaneni (14)

Dr. Ashok Patel

Department of Computer Science
University of Massachusetts Dartmouth,
MA

1. Problem statement

1.1 Background

The Olympic Games, a four-year worldwide sporting event, create massive amounts of data about participants, sports, performances, and historical trends. The International Olympic Committee (IOC) wants to use data analytics to gather important insights, improve decision-making processes, better the overall experience of athletes and spectators, and maximize future Olympic Games planning.

1.2 Primary Objective

Performance Analysis

Analyze historical performance data of athletes from various Olympic Games to find patterns, outliers, and important performance markers. This entails comprehending the impact of numerous aspects such as training regimens, technological improvements in sports, and regulation changes.

Athlete Profiling:

Create athlete profiles based on performance measures, demographics, and participation history. Investigate the relationship between an athlete's upbringing and their achievement in various sports or events.

Viewership Trends:

During the Olympic Games, analyze viewership data such as television ratings, online streaming, and social media participation. To improve the spectator experience, identify patterns, preferences, and factors impacting audience involvement.

Resource Optimization

Using previous data, optimize resource allocation for future Olympic Games. This involves venue selection, event scheduling, and resource allocation for things like security, transportation, and lodging.

Predictive Modelling

Develop predictive models that estimate medal positions, individual performances, and potential areas for improvement for certain countries or athletes. This can help participating nations with strategic planning and resource allocation.

1.3 Expected Outcomes

The Olympic Data Analytics project seeks to provide the International Olympic Committee, participant countries, and stakeholders with actionable insights. Improved decision-making, improved athlete preparation tactics, optimized resource allocation, and an enhanced spectator experience are among the outcomes.

1.4 Methodology

We will extract the data from the API using Azure data factory that is kind of like data pipeline tool available on azure. It will build a flow like this and load our data onto the Azure data Lake storage. First, we will load raw data then using azure data bricks we will write our spark code and transform our data and load our data back to our transform data lake storage. Once that is done, we will use synapse analytics to run the SQL queries on top of the transform data so that we can find the insights and get the visualization on top.

1.5 Software

Azure, sometimes known as Microsoft Azure, is a cloud computing platform and service provided by Microsoft. It offers a variety of cloud services, including as computing power, storage, networking, databases, machine learning, analytics, and more. Azure is intended to assist enterprises in developing, deploying, and managing applications and services via Microsoft's global network of data centers.

We used Azure services like Data Factory, Data Lake Gen 2, DataBricks for this project.

2. Dataset

2.1 Data Collection

We found our Olympic Data Analytics Dataset from Kaggle. This is a dataset containing every modern Olympic Games from Athens in 1896 to Rio in 2016. Until 1992, the Winter and Summer Games were held in the same year. Following that, they staggered them such that the Winter Games occur on a four-year cycle beginning in 1994, followed by Summer in 1996, Winter in 1998, and so on. When evaluating this data, many individuals make the error of assuming that the Summer and Winter Games have always been staggered.

2.2 Data Contents

Athlete_events.csv is a spreadsheet with 271116 rows and 15 columns. Each row represents a single athlete competing in a single Olympic event (athlete-events).

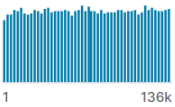
Columns:

1. **ID** - Unique number for each athlete
2. **Name** - Athlete's name
3. **Sex** - M or F
4. **Age** - Integer
5. **Height** - In centimeters
6. **Weight** - In kilograms
7. **Team** - Team name
8. **NOC** - National Olympic Committee 3-letter code
9. **Games** - Year and season
10. **Year** - Integer
11. **Season** - Summer or Winter
12. **City** - Host city
13. **Sport** - Sport
14. **Event** - Event
15. **Medal** - Gold, Silver, Bronze, or NA

2.3 Exploring the data

About this file

Each row is an athlete-event. The ID column can be used to uniquely identify athletes, since some athletes have the same name.

ID	Name	Sex	Age	Height	Weight
 1136k	134732 unique values	M73% F27%	238% 248% Other (227521)84%	NA22% 1805% Other (198453)73%	NA
1	A Dijiang	M	24	180	80
2	A Lamusi	M	23	170	60
3	Gunnar Nielsen Aaby	M	24	NA	NA
4	Edgar Lindenau Aabye	M	34	NA	NA
5	Christine Jacoba Aaftink	F	21	185	82
5	Christine Jacoba Aaftink	F	21	185	82
5	Christine Jacoba Aaftink	F	25	185	82

About this file

- 1. NOC (National Olympic Committee 3 letter code)
- 2. Country name (matches with regions in map_data("world"))
- 3. Notes

A NOC	A region	A notes
230 unique values	Germany 2% Czech Republic 1% Other (223) 97%	[null] 91% Netherlands Antilles 0% Other (20) 9%
AFG	Afghanistan	
AHO	Curacao	Netherlands Antilles
ALB	Albania	
ALG	Algeria	
AND	Andorra	
ANG	Angola	
ANT	Antigua	Antigua and Barbuda
...

3. Azure Services

3.1 Azure

Azure is widely utilized by businesses of all sizes, from startups to major corporations, to capitalize on cloud computing features such as scalability, flexibility, and cost efficiency. Azure services can be accessed via a web-based portal, command-line interface, or RESTful APIs. Azure's worldwide presence, which includes data centers all over the world, ensures low-latency access to services as well as compliance with regional data residency regulations.

3.2 Services

Data factory

Data integration service that enables you to create, schedule, and manage data pipelines for efficient data movement and transformation between various sources and destinations in Azure and beyond. It simplifies ETL (Extract, Transform, Load) and data integration

Data Lake Gen 2

Data lake solution that combines the capabilities of a data lake with the power of Azure Blob Storage, allowing you to store and analyze large volumes of structured and unstructured data with enhanced performance, security, and analytics capabilities.

Azure Databricks

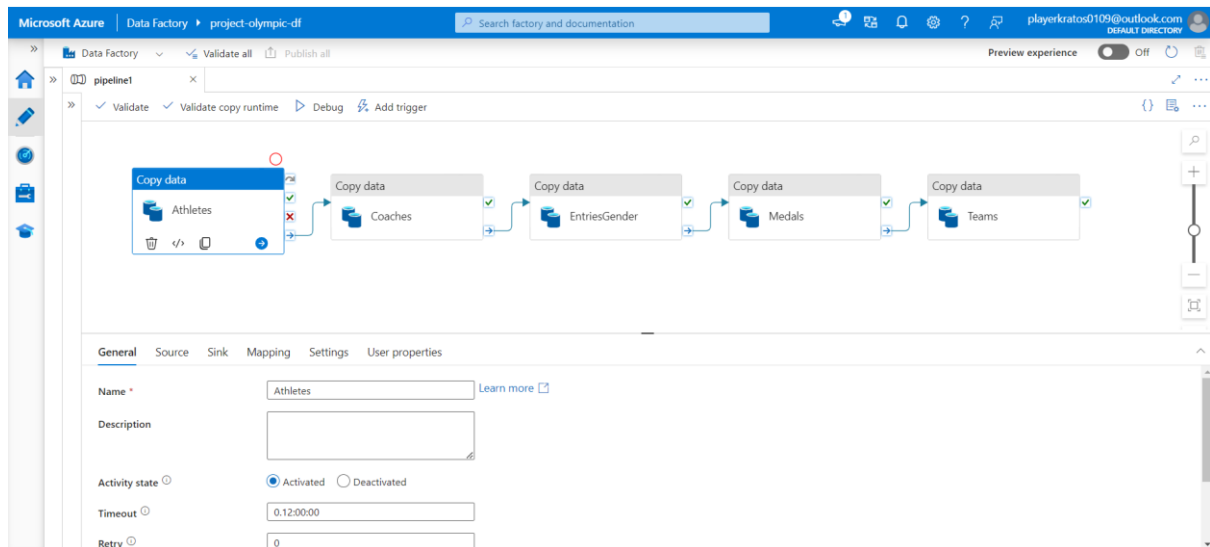
Databricks is a unified analytics platform built on top of Apache Spark, designed to help data engineers and data scientists collaborate on big data processing and machine learning tasks. It provides a secure data catalog, data processing, and building machine learning

Synapse Analytics

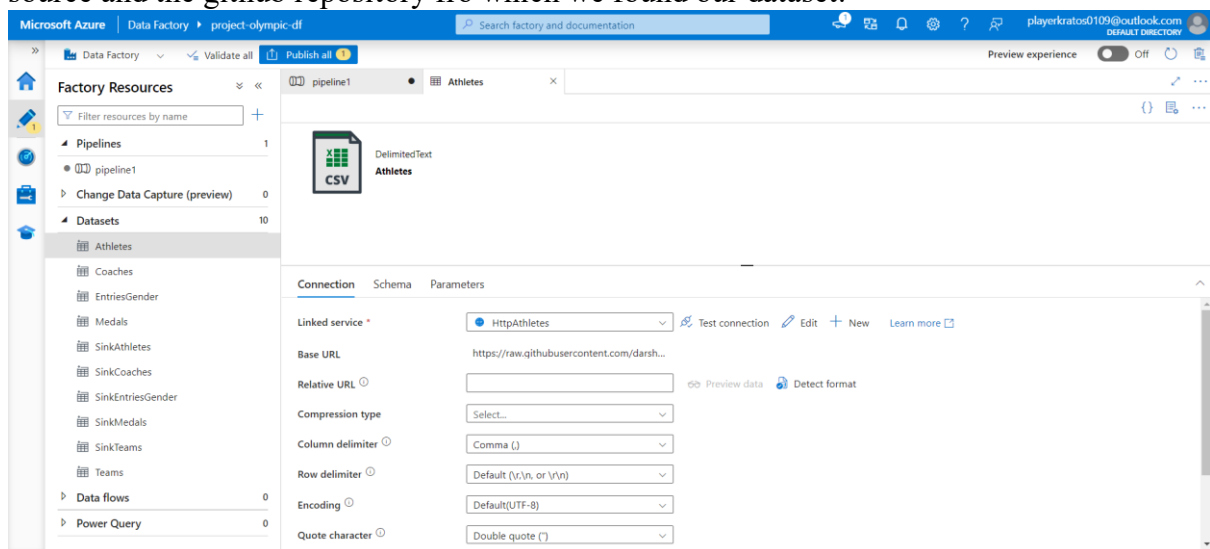
SQL Data Warehouse, is a cloud-based analytics service provided by Microsoft Azure. It combines big data and data warehousing into a single integrated platform, allowing organizations to analyze and process large volumes of data for business intelligence and data analytics purposes.

3.3 Implemented Services

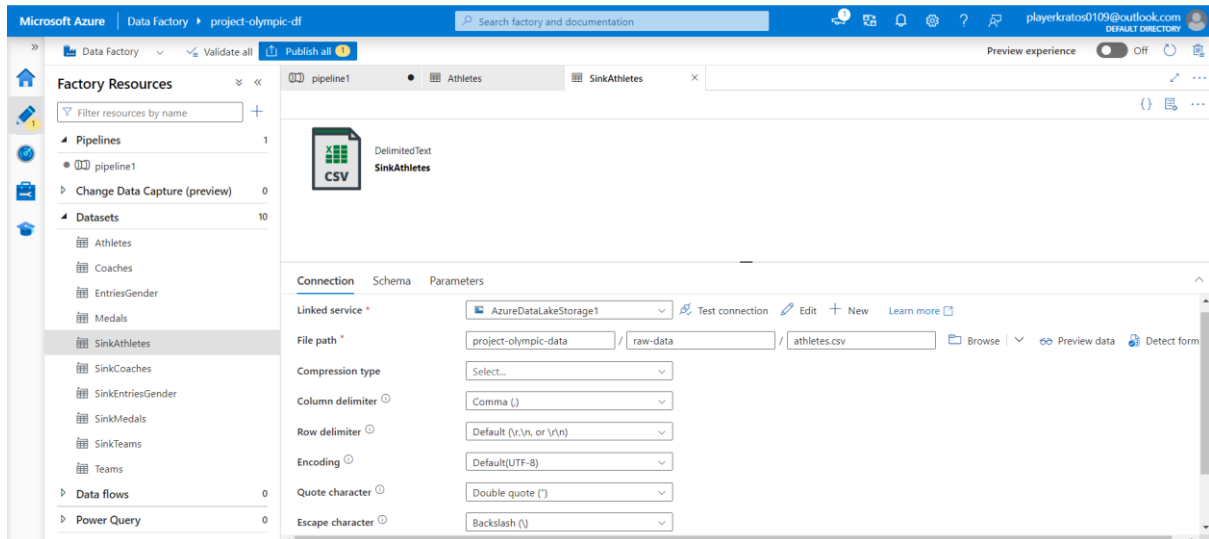
Data Factory:



For creating the above data pipeline we have ingested the raw data from the data storage container for all the above intents like athletes, coaches , entries gender, models and teams. We are creating a link to the data factory from the github repository that has the raw data for the file athletes and for that we create a linked service to create one between the data source and the github repository from which we found our dataset.



After linking our source to the data factory and reviewing it by clicking on the preview data , we do need to do a sink that means loading our data onto the data lake 2nd gen storage . For that we have a create a new source under the sink tab and while selecting properties we create a new linked service and make sure we select the storage account we created to finish setting up.

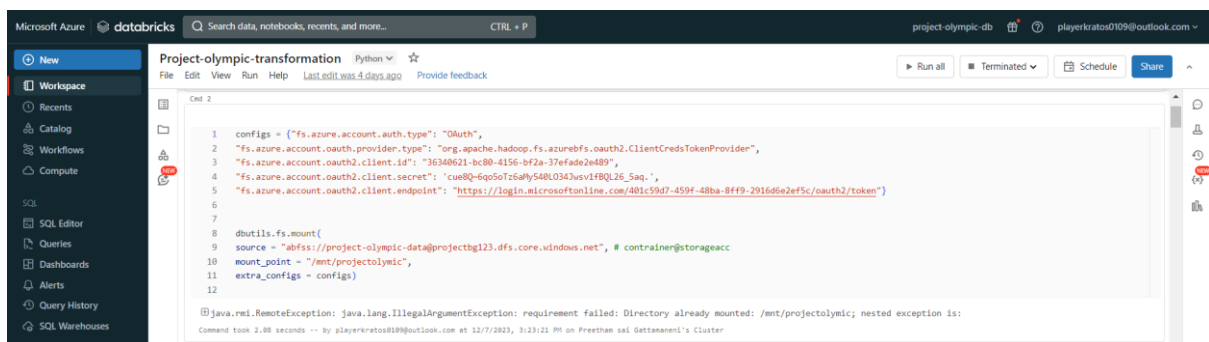


This way we successfully created the pipeline part for the file with the athletes file from the raw data. For the remaining files coaches, teams, Entries gender and medals; the same steps are followed to create the link from the data factory to the git hub repository that contains the required raw .csv file and to load our data onto the Data lake storage but while creating the new Sink connection every time make sure that the same data storage space is selected from the linked services.

Data Bricks and Visualization:

Breaking down the Data Bricks Query

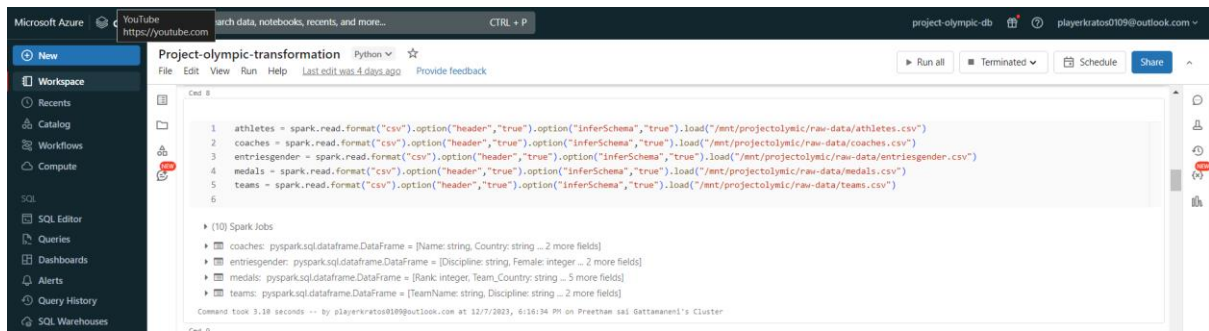
Mounting the Azure Data Lake Storage:



Explanation:

This code installs an Azure Data Lake Storage Gen2 container named project-olympic-data to the Databricks file system at /mnt/projectolympic. The credentials are supplied via the configuration dictionary.

Loading Data into Spark Data Frames:



The screenshot shows a Databricks workspace interface. The top bar indicates the user is logged in as 'playerkratos0109@outlook.com'. The workspace is named 'Project-olympic-transformation'. The main area displays a Python script in a code editor, which is being executed. The script loads five CSV files from the '/mnt/projectolympic/raw-data/' directory into Spark DataFrames: athletes, coaches, entriesgender, medals, and teams. The output of the script is visible below the code, showing the schema and the first few rows of each DataFrame. The command took 3.18 seconds to execute.

```
1 athletes = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/projectolympic/raw-data/athletes.csv")
2 coaches = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/projectolympic/raw-data/coaches.csv")
3 entriesgender = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/projectolympic/raw-data/entriesgender.csv")
4 medals = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/projectolympic/raw-data/medals.csv")
5 teams = spark.read.format("csv").option("header","true").option("inferSchema","true").load("/mnt/projectolympic/raw-data/teams.csv")
6
```

▶ (10) Spark Jobs

- coaches: pyspark.sql.dataframe.DataFrame = [Name: string, Country: string ... 2 more fields]
- entriesgender: pyspark.sql.dataframe.DataFrame = [Discipline: string, Female: integer ... 2 more fields]
- medals: pyspark.sql.dataframe.DataFrame = [Rank: integer, Team_Country: string ... 5 more fields]
- teams: pyspark.sql.dataframe.DataFrame = [TeamName: string, Discipline: string ... 2 more fields]

Command took 3.18 seconds -- by playerkratos0109@outlook.com at 12/7/2023, 6:16:34 PM on Preethan sai Gattamaneni's Cluster

Explanation:

The code extracts the following data into Spark DataFrames from CSV files in the /mnt/projectolympic/raw-data/ directory: athletes, coaches, entries, gender, medals, and teams.

Displaying data and printing schema :

athletes.show()

athletes.printSchema()

coaches.show()

coaches.printSchema()

entriesgender.show()

entriesgender.printSchema()

medals.show()

medals.printSchema()

teams.show()

teams.printSchema()

Explanation:

These commands display the first few rows and the schema of each Spark DataFrame.

Data Analysis:



```
1 top_gold_medal_countries = medals.orderBy("Gold", ascending=False).select("Team_Country","Gold").show()
```

Command skipped

Command took 2.81 seconds -- by playerkratos0109@outlook.com at 12/7/2023, 3:23:21 PM on Preethan sai Gattamaneni's Cluster

```
1 average_entries_by_gender = entriesgender.withColumn(
2   'Avg_Female', entriesgender['Female'] / entriesgender['Total']
3 ).withColumn(
4   'Avg_Male', entriesgender['Male'] / entriesgender['Total']
5 )
6 average_entries_by_gender.show()
```

Command skipped

Command took 2.81 seconds -- by playerkratos0109@outlook.com at 12/7/2023, 3:23:21 PM on Preethan sai Gattamaneni's Cluster

Explanation:

Retrieves and displays the top countries with the highest number of gold medals.

Calculates and displays the average entries by gender for each discipline.

Data Transformation and writing to Azure Data Lake:



```
1 top_gold_medal_countries = medals.orderBy("Gold", ascending=False).select("Team_Country","Gold").show()
```

Command skipped

Command took 2.81 seconds -- by playerkratos0109@outlook.com at 12/7/2023, 3:23:21 PM on Preethan sai Gattamaneni's Cluster

```
1 average_entries_by_gender = entriesgender.withColumn(
2   'Avg_Female', entriesgender['Female'] / entriesgender['Total']
3 ).withColumn(
4   'Avg_Male', entriesgender['Male'] / entriesgender['Total']
5 )
6 average_entries_by_gender.show()
```

Command skipped

Command took 2.81 seconds -- by playerkratos0109@outlook.com at 12/7/2023, 3:23:21 PM on Preethan sai Gattamaneni's Cluster

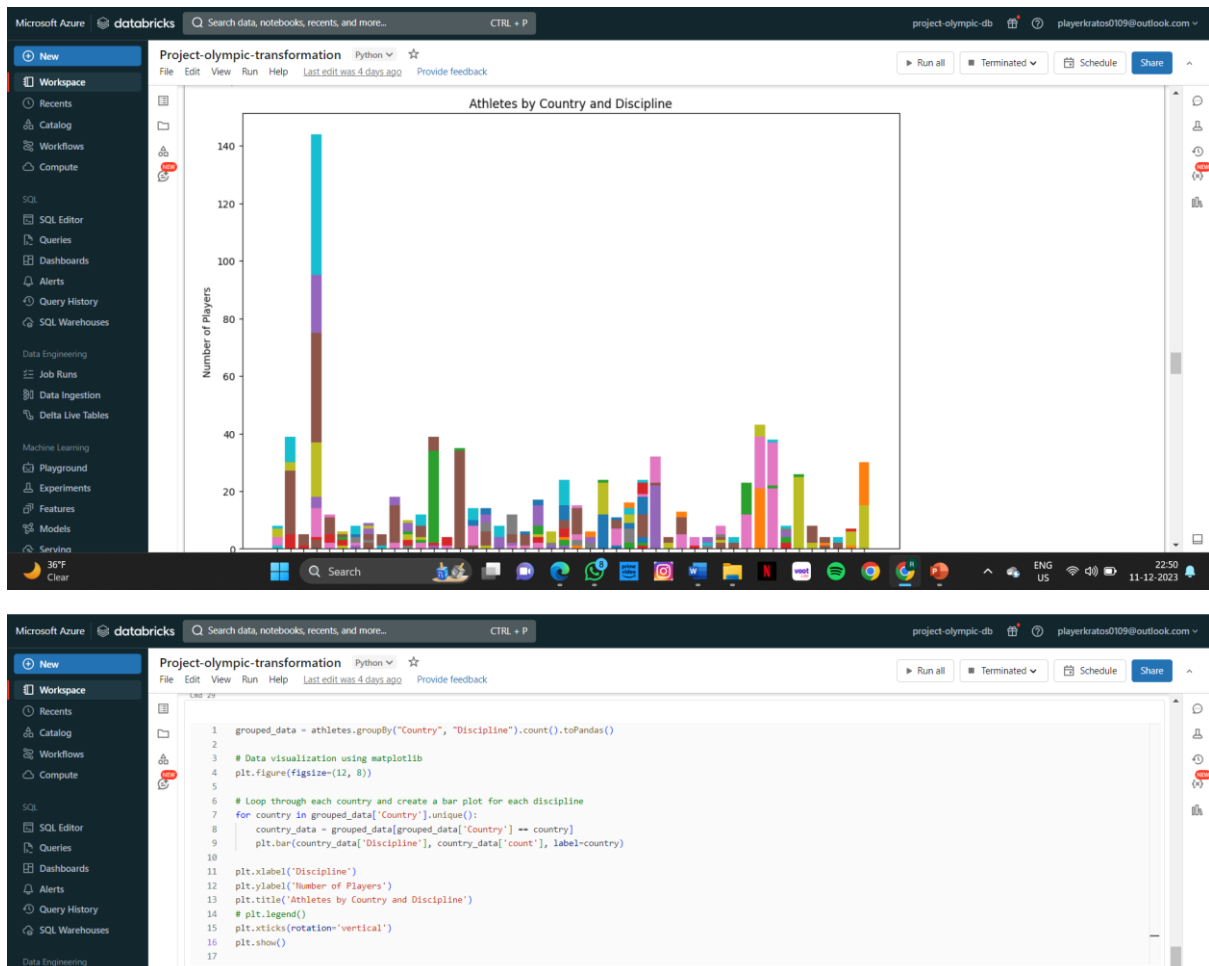
Explanation:

Repartitions the DataFrames into a single partition (helpful for small datasets) and sends them to Azure Data Lake Storage in the location /mnt/projectolympic/transformed-data/.

This code, in short, mounts Azure Data Lake Storage, loads data from CSV files into Spark DataFrames, does data analysis, type casting, and transformations, and then publishes the changed data back to Azure Data Lake Storage. To efficiently handle and process massive amounts of data, PySpark on Databricks is used.

Data Visualisation:

For Data Visualisation we used matplotlib files to depict the transformed data.



Explanation:

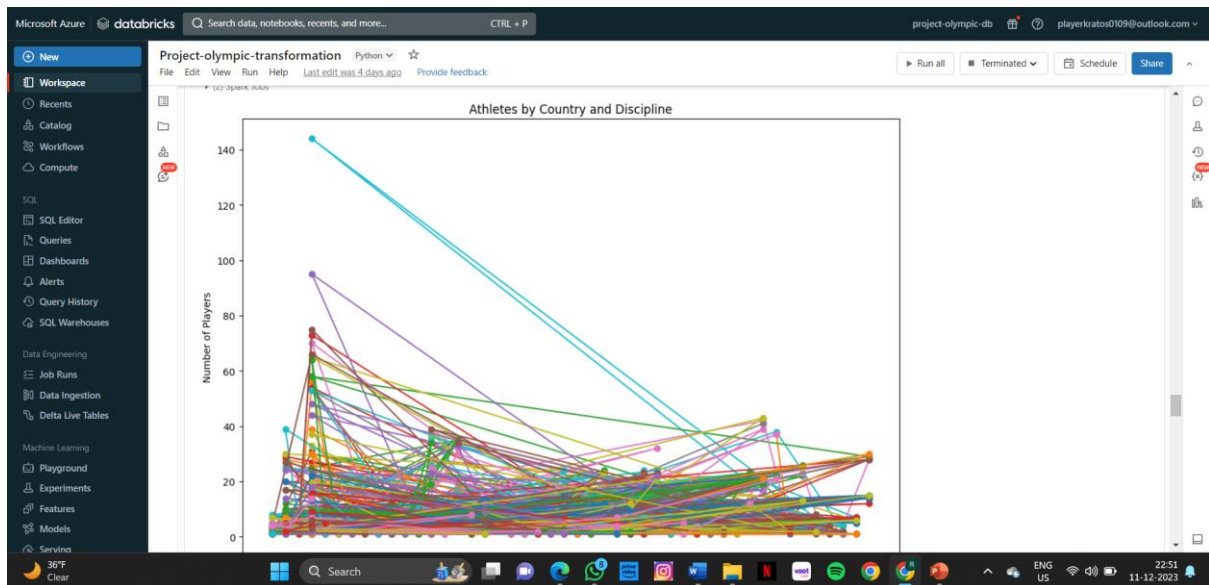
This code divides the athletes DataFrame into groups based on the columns "Country" and "Discipline" and counts the number of occurrences (players) in each group. The outcome is a Pandas DataFrame (grouped_data).

Creates a new figure for the future bar plot with the dimensions 12 inches wide by 8 inches high.

This loop traverses each distinct country in the grouped_data DataFrame.

It selects the related data (country_data) for each country and generates a bar plot with Matplotlib.

The values on the x-axis are disciplines (country_data['Discipline']), and the values on the y-axis are counts (country_data['count']).

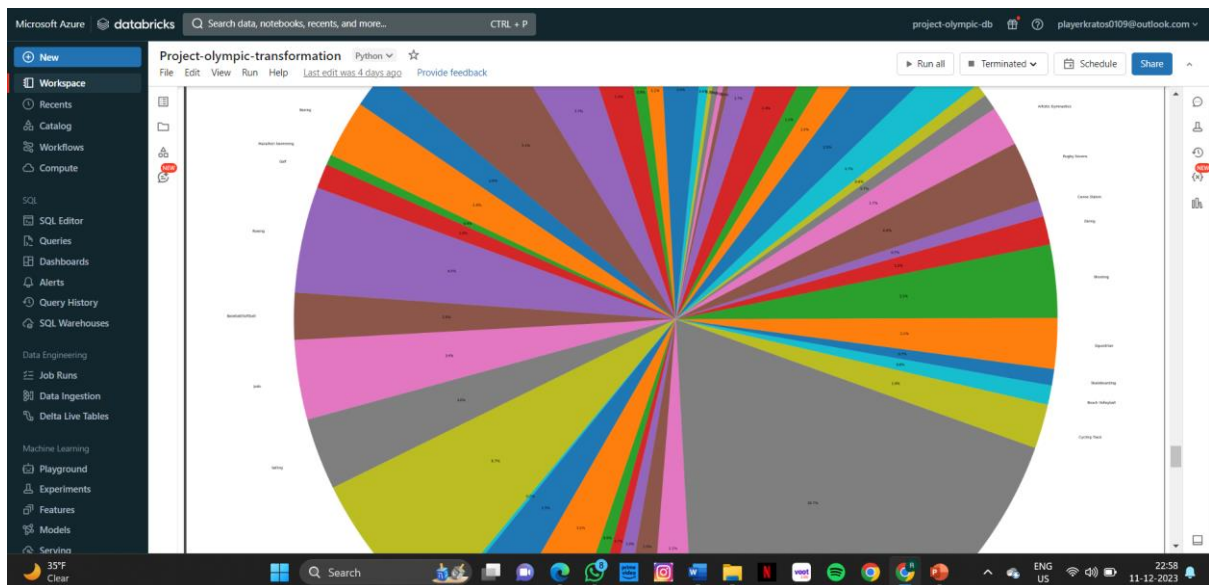


```
1 grouped_data = athletes.groupby("Country", "Discipline").count().toPandas()
2
3 plt.figure(figsize=(12, 8))
4
5 for country in grouped_data['Country'].unique():
6     country_data = grouped_data[grouped_data['Country'] == country]
7     plt.plot(country_data['Discipline'], country_data['count'], label=country, marker='o')
8
9 plt.xlabel('Discipline')
10 plt.ylabel('Number of Players')
11 plt.title('Athletes by Country and Discipline')
12 # plt.legend()
13
14 # Rotate x-axis labels for better readability
15 plt.xticks(rotation='vertical')
16
17 plt.show()
```

Explanation:

This code divides the athletes DataFrame into groups based on the columns "Country" and "Discipline" and counts the number of occurrences (players) in each group. The outcome is a Pandas DataFrame (grouped_data).

Creates a new figure with the provided dimensions of 12 inches in width and 8 inches in height for the future line plot. This loop traverses each distinct country in the grouped_data DataFrame. It selects the related data (country_data) for each country and generates a line plot with Matplotlib. The values on the x-axis are disciplines (country_data['Discipline']), and the values on the y-axis are counts (country_data['count']).



`startangle=140`: Defines the angle at which the first slice will begin. In this case, it begins at a 140-degree angle.

4. Conclusion

The Olympic Data Analytics project resulted in a comprehensive and fascinating analysis of the Tokyo 2020 Olympic Games, shedding light on a wide range of aspects of sports and competition. The comprehensive analysis covered a wide range of factors, from athlete demographics to medal successes, providing a sophisticated understanding of the complexities that define the global athletic event.

The rich tapestry of athlete demographics was one of the analysis's key focal points. The initiative revealed the distribution of athletes across nations, disciplines, and genders through thorough investigation. This examination of participation trends not only highlighted the diverse range of countries that contribute to the Olympic stage, but also dug into the dynamics of gender representation within the competitive scene. The findings highlighted the underlying diversity of the sporting globe, providing insights into the growing fabric of global athletics.

Finally, the Olympic Data Analytics project exemplifies the power of data in uncovering the complexities of one of the world's most recognized athletic events. The findings not only serve as a retrospective evaluation of the Tokyo 2020 Olympic Games, but also as a motivator for further inquiries and ongoing research in the field of sports analytics.

5. References

- Datasets from Kaggle

<https://www.kaggle.com/datasets/arjunprasadsarkhel/2021-olympics-in-tokyo>

- Public Datasets from github

[tokyo-olympic-azure-data-engineering-project/data at main · darshilparmar/Tokyo-olympic-azure-data-engineering-project · GitHub](#)

- Project information

<https://olympics.com/en/>