Asmita Porwal
Data Engineering
Batch-1
21/02/2024

Coding Challenge -4 Question-1

Exploratory data analysis (EDA) in Databricks & Visualizing data in Databricks

Exploratory Data Analysis (EDA) is a crucial step in the data analysis process, and Databricks provides a powerful platform for performing EDA and visualizing data. Here's a general guide on how you can perform EDA and visualize data using Databricks:

1. Loading Data:

- Begin by loading your dataset into Databricks. You can do this from various sources such as Azure Data Lake Storage, Azure Blob Storage, Azure SQL Database, etc.
- Databricks supports multiple file formats like CSV, Parquet, JSON, etc. You can use the appropriate reader to load your data into a DataFrame.

2. Understanding Data:

- Once the data is loaded, you can use DataFrame operations to explore its structure and contents. Methods like display, show, describe, schema, etc., can be helpful.
- Check for missing values, data types, summary statistics, unique values, etc., to get a better understanding of your data.

3. Data Visualization:

- Databricks supports various visualization libraries such as Matplotlib, Seaborn, Plotly, etc., which you can use directly in your notebooks.

- You can create different types of plots like histograms, scatter plots, bar plots, line plots, etc., to visualize the distribution, relationships, and patterns in your data.

4. Interactive Visualization:

- Databricks also supports interactive visualization libraries like Bokeh and Plotly, which allow you to create interactive plots for better exploration and analysis.
- Interactive plots enable you to zoom, pan, hover over data points, etc., providing a more dynamic and insightful analysis experience.

5. Dashboarding:

- You can create dashboards in Databricks using the built-in dashboarding functionality. Dashboards allow you to combine multiple visualizations and controls into a single interactive interface.
- You can customize the layout, add filters, and create dynamic interactions between visualizations to build rich and informative dashboards for your data analysis.

6. Sharing Results:

- Once you have performed EDA and created visualizations, you can share your findings with others by exporting notebooks or dashboards, or by granting access to your Databricks workspace.
- Collaborators can view and interact with your analysis, providing feedback and insights to further refine your exploration.

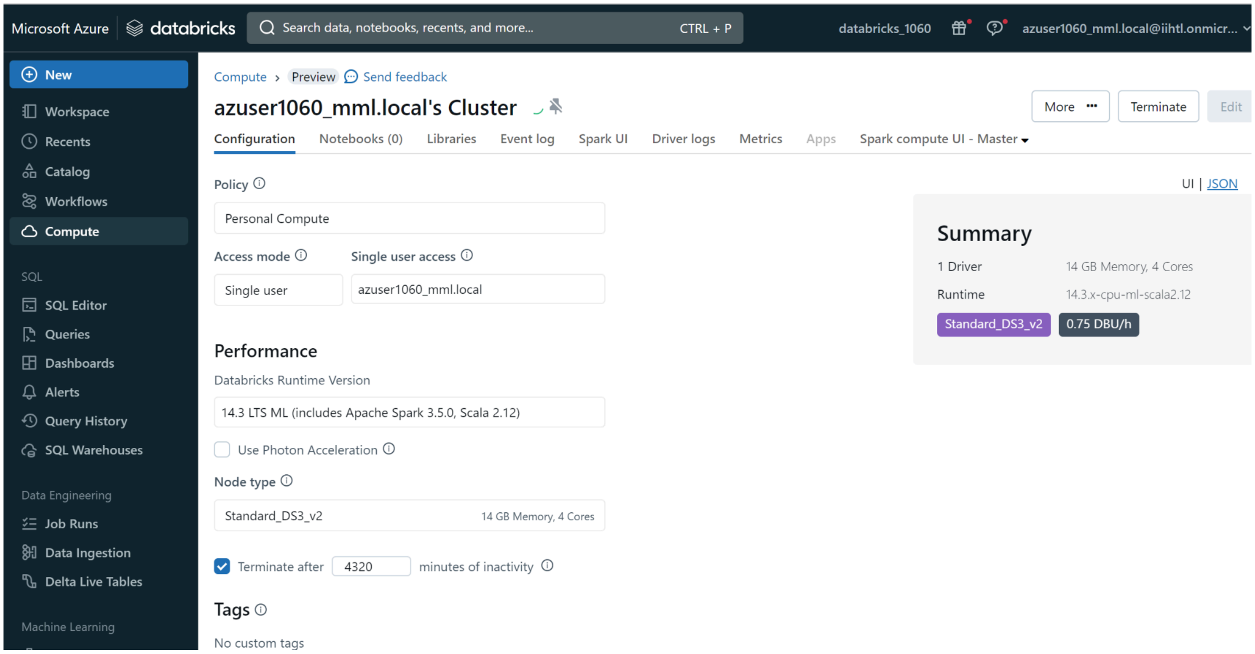
Visualizing data in Databricks

Steps to create visualization

In order to create visualizations, we need to have data.

- After creating a table
- Click on + symbol
- Click on visualization.
- Select the type of visualization, then select Scatter

1.Creating a cluster



azuser1060_mml.local's Cluster

Policy: Personal Compute

Access mode: Single user

Single user access: azuser1060_mml.local

Performance

Databricks Runtime Version: 14.3 LTS ML (includes Apache Spark 3.5.0, Scala 2.12)

☐ Use Photon Acceleration

Node type: Standard_DS3_v2 — 14 GB Memory, 4 Cores

☑ Terminate after 4320 minutes of inactivity

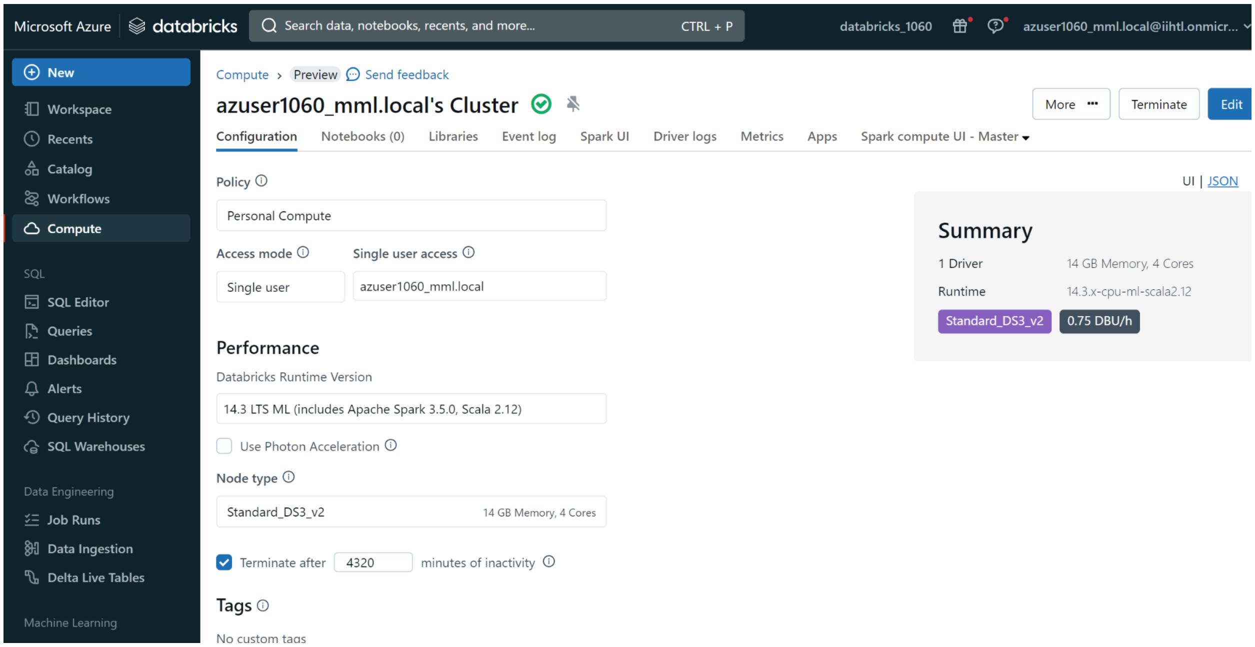
Tags

No custom tags

Summary

1 Driver	14 GB Memory, 4 Cores
Runtime	14.3.x-cpu-ml-scala2.12

Standard_DS3_v2 0.75 DBU/h



azuser1060_mml.local's Cluster ✓

Policy: Personal Compute

Access mode: Single user

Single user access: azuser1060_mml.local

Performance

Databricks Runtime Version: 14.3 LTS ML (includes Apache Spark 3.5.0, Scala 2.12)

☐ Use Photon Acceleration

Node type: Standard_DS3_v2 — 14 GB Memory, 4 Cores

☑ Terminate after 4320 minutes of inactivity

Tags

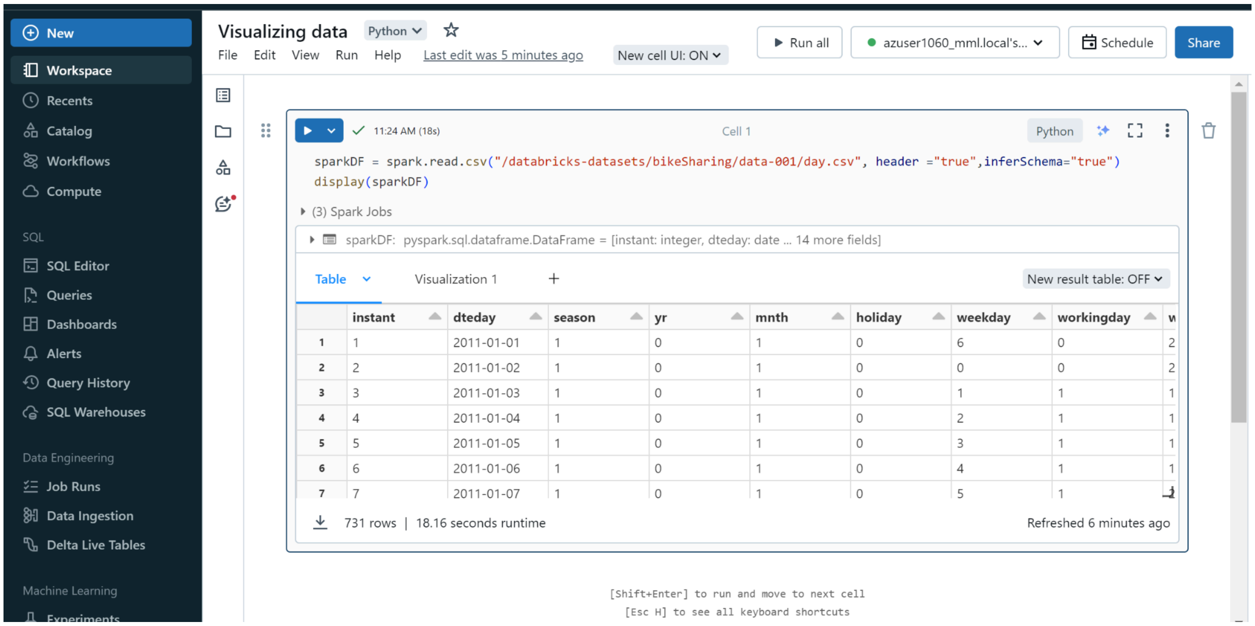
No custom tags

Summary

1 Driver	14 GB Memory, 4 Cores
Runtime	14.3.x-cpu-ml-scala2.12

Standard_DS3_v2 0.75 DBU/h

2.Creating a table



```
sparkDF = spark.read.csv("/databricks-datasets/bikeSharing/data-001/day.csv", header ="true",inferSchema="true")
display(sparkDF)
```

(3) Spark Jobs

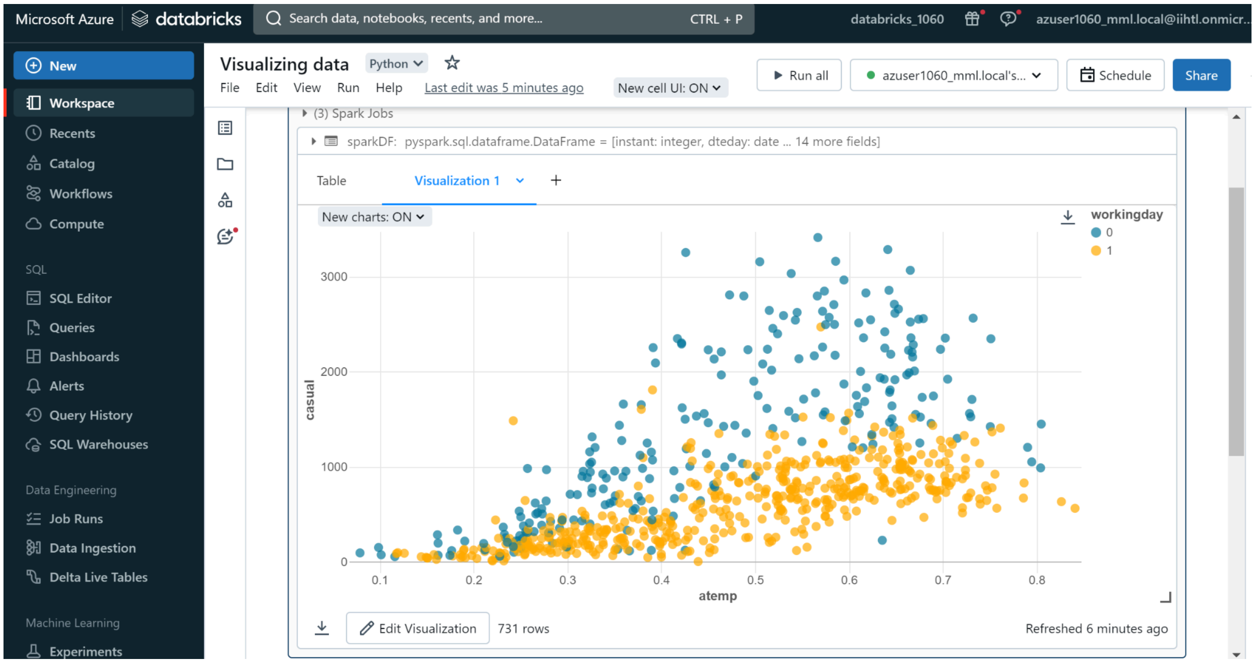
sparkDF: pyspark.sql.dataframe.DataFrame = [instant: integer, dteday: date ... 14 more fields]

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	w
1	1	2011-01-01	1	0	1	0	6	0	2
2	2	2011-01-02	1	0	1	0	0	0	2
3	3	2011-01-03	1	0	1	0	1	1	1
4	4	2011-01-04	1	0	1	0	2	1	1
5	5	2011-01-05	1	0	1	0	3	1	1
6	6	2011-01-06	1	0	1	0	4	1	1
7	7	2011-01-07	1	0	1	0	5	1	1

731 rows | 18.16 seconds runtime

Refreshed 6 minutes ago

```
[Shift+Enter] to run and move to next cell
[Esc H] to see all keyboard shortcuts
```



(3) Spark Jobs

sparkDF: pyspark.sql.dataframe.DataFrame = [instant: integer, dteday: date ... 14 more fields]

731 rows

Refreshed 6 minutes ago