**Madiha Aimon Tappal**

[madihaaimon@gmail.com](mailto:madihaaimon@gmail.com)

**Data Engineering Batch – 1**

**Day – 17 Assignment**

**Azure Databricks**

**Visualizations in Azure Databricks: -**

Azure Databricks is a cloud-based big data and analytics platform built on Apache Spark. It provides an integrated environment for data engineers, data scientists, and business analysts to collaborate and work with big data. Visualizations in Azure Databricks are essential for exploring and understanding data, as well as communicating insights. Here's a theoretical overview of visualizations in Azure Databricks:

1. **Notebooks and Cells:**
   * In Azure Databricks, you typically work within notebooks, which contain cells that can hold code, text, or visualizations.
   * Cells can include Spark code (Scala, Python, SQL, or R), Markdown for documentation, and visualizations.
2. **Displaying Visualizations:**
   * You can create visualizations using various libraries such as Matplotlib, Seaborn, Plotly, and others in Python; ggplot2 in R; and other plotting libraries in different languages supported by Databricks.
   * Visualizations can be displayed directly in the notebook cells, making it easy to explore and interpret the results.
3. **Interactive Charts:**
   * Databricks supports interactive charts, allowing users to zoom in, pan, and interact with the visualizations for a more detailed analysis.
   * This is particularly useful for exploring large datasets and identifying patterns or anomalies.
4. **Widgets and Parameters:**
   * Widgets are interactive elements that allow users to manipulate parameters and dynamically update visualizations.
   * By using widgets, you can create dashboards or reports where users can customize their views based on specific criteria.
5. **Built-in Visualizations:**
   * Databricks provides built-in visualizations for common operations like histograms, line charts, bar charts, scatter plots, and more.
   * These visualizations can be generated with minimal code, making it easy for users to quickly analyze and understand their data.
6. **Integration with Azure Services:**
   * Azure Databricks integrates with Azure services like Azure Synapse Analytics, Azure Data Lake Storage, and Azure SQL Data Warehouse, allowing seamless data integration and visualization.
7. **Job and Cluster Monitoring:**
   * Visualizations are also used for monitoring the performance of jobs and clusters in Azure Databricks.
   * Users can track the resource utilization, job status, and other performance metrics through visual representations.
8. **MLflow Tracking:**
   * MLflow, an open-source platform for managing the end-to-end machine learning lifecycle, is integrated into Azure Databricks. It includes visualizations for tracking and comparing machine learning experiments.

In summary, visualizations in Azure Databricks play a crucial role in data exploration, analysis, and communication of insights. They enable users to create dynamic, interactive reports and dashboards, making it easier to derive meaningful insights from large and complex datasets.

**Creating visualizations in Azure Databricks** involves using a combination of code, data manipulation, and visualization libraries. Here are the general steps to create visualizations in Azure Databricks:

**1. Set Up Azure Databricks:**

* Ensure you have an active Azure Databricks workspace.
* Create a cluster to run your code.

**2. Create or Import a Notebook:**

* Open the Azure Databricks workspace.
* Create a new notebook or import an existing one.

**3. Choose a Language:**

* Decide on the programming language for your notebook (e.g., Scala, Python, SQL, or R).

**4. Load Data:**

* Load your data into a DataFrame or another suitable data structure.
* You can read data from various sources, such as Azure Data Lake Storage, Azure Blob Storage, or external databases.

**5. Data Exploration:**

* Explore your data using Spark DataFrame operations or SQL queries in the notebook cells.
* Ensure you understand the structure and characteristics of your data.

**6. Data Cleaning and Transformation:**

* Preprocess your data as needed, handling missing values, transforming columns, or aggregating information.

**7. Choose a Visualization Library:**

* Select a visualization library based on your chosen programming language.
* Common libraries for Python include Matplotlib, Seaborn, and Plotly.
* For R, you might use ggplot2 or other relevant packages.

**8. Create Visualizations:**

* Write code to generate visualizations based on your data.
* Use functions provided by the chosen library to create charts, graphs, or other visual representations.
* Include the visualization code in separate notebook cells.

**9. Display Visualizations:**

* Execute the cells containing visualization code to generate the charts.
* Visualizations are typically displayed directly below the code cells in the notebook.

**Creating new visualizations:**

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

Here I provided a data and used display() method.

* After creating a table
* Click on + symbol
* Click on visualization.
* Select the type of visualization, here I selected “Scatter”















