BIG DATA ANALYTICS WITH IBM CLOUD

DEVELOPMNT PART 2:

To develop queries or scripts to explore and analyze a dataset in a database, you'll need to adapt the code to the specific database service you're using (e.g., Db2 or MongoDB) and the dataset you're working with. Below, I'll provide examples for both structured data (Db2) and semi-structured data (MongoDB) and demonstrate basic data cleaning and transformation operations.

- Structured Data (e.g., Db2):

Suppose you have a structured dataset in a Db2 database. You can use SQL queries for exploration and analysis:

- Select Data:

Retrieve a sample of data to get an overview of what's in the dataset.

```
SELECT *
FROM your_table
FETCH FIRST 10 ROWS ONLY;
```

- Data Cleaning: Remove duplicates

```
DELETE FROM your_table
WHERE ROWID NOT IN (
SELECT MAX(ROWID)
 FROM your_table
 GROUP BY column1, column2
);
```

- Data Transformation:

Calculate the sum or average of a numerical column

```
SELECT AVG(numeric_column) AS avg_value
FROM your_table;
```

- Semi-Structured Data (e.g., MongoDB):

If you are working with semi-structured data in a MongoDB database, you can use MongoDB query operations for exploration and analysis:

- Select Data:

Retrieve documents that meet specific criteria.

```
db.collection.find({ field: value }).limit(10);
```

- Data Cleaning:

  Remove documents with missing or null values in a specific field.

  ```
  db.collection.deleteMany({ field: null });
  ```

- Data Transformation:

  Perform an aggregation to calculate the average of a numeric field.

  ```
  db.collection.aggregate([
    {
      $group: {
        _id: null,
        avg_value: { $avg: "$numeric_field" }
      }
    }
  ]);
  ```

METHODS:

1. Data Preprocessing:

Continue cleaning and preparing your data. Ensure it's in the right format for the advanced analysis techniques you plan to apply.

2. Advanced Analysis:

Depending on your dataset and project objectives, apply advanced analysis techniques. These may include:

(i) Machine Learning Algorithms: Implement machine learning models for classification, regression, clustering, or recommendation, depending on the problem you're addressing.

(ii) Time Series Analysis: If your data involves time-based observations, apply timeseries analysis to identify trends, patterns, and seasonality.

(iii) Sentiment Analysis: If your data contains text, perform sentiment analysis togauge public opinion, customer feedback, or social media sentiment.

3. Model Training and Evaluation:

If using machine learning, split your data into training and testing sets. Train your models and evaluate their performance using appropriate metrics.

4. Visualization:

Create visualizations to showcase the results of your advanced analysis. Visualization tools like Matplotlib, Plotly, or IBM Watson Studio can be useful.

Types of visualizations can include:

- Line Charts: For time series analysis or trends over time.
- Bar Charts: To compare different categories or groups.
- Scatter Plots: To visualize relationships between variables.
- Heatmaps: For correlation or sentiment analysis.
- Word Clouds: For visualizing word frequency in text data.
- Interactive Dashboards: Use tools like Plotly Dash or Tableau to create interactive dashboards for in-depth exploration.

5. Insights and Recommendations:

- Interpret the results of your analysis and draw meaningful insights. Considerthe implications for your project's objectives.
- Provide recommendations or actions based on your findings.6.Documentation and Reporting:

Document your analysis process, the techniques used, and the visualizations created.

This documentation will be valuable for presenting your work to stakeholders.7.Testing and Validation:

Test your solution to ensure its accuracy and reliability. Validate the results againstknown benchmarks or ground truth data if available.

8. Optimization:

If needed, optimize your analysis pipeline for performance and efficiency, especiallywhen working with large datasets.

9. Integration:

If this is part of a larger system or application, integrate your big data analysis solutionwith the rest of the infrastructure.

10. Feedback and Iteration:

Collect feedback from relevant stakeholders and iterate on your analysis andvisualizations as necessary.

Code:

```
import matplotlib.pyplot as plt
import plotly.express as px

# Sample data (replace with your own data)
data = [10, 20, 30, 40, 50]
labels = ['A', 'B', 'C', 'D', 'E']

# Matplotlib Bar Chart
plt.figure(figsize=(8, 6))
plt.bar(labels, data)
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Bar Chart')
plt.show()

# Plotly Bar Chart
fig = px.bar(x=labels, y=data, title='Bar Chart')
fig.show()

# Matplotlib Line Chart
plt.figure(figsize=(8, 6))
plt.plot(labels, data, marker='o')
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Line Chart')
plt.show()

# Plotly Line Chart
fig = px.line(x=labels, y=data, markers=True, title='Line Chart')
fig.show()

# Matplotlib Pie Chart
plt.figure(figsize=(8, 8))
```

```
plt.pie(data, labels=labels, autopct='%1.1f%%', startangle=140)
plt.title('Pie Chart')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()

# Plotly Pie Chart
fig = px.pie(values=data, names=labels, title='Pie Chart')
fig.show()

# Matplotlib Scatter Plot
x = [1, 2, 3, 4, 5]
y = [20, 25, 30, 35, 40]
plt.figure(figsize=(8, 6))
plt.scatter(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')
plt.show()

# Plotly Scatter Plot
scatter_fig = px.scatter(x=x, y=y, title='Scatter Plot')
scatter_fig.show()
```