Name-Ishan Kiran Joshi Div-D15C Roll No-21 A.Y.-2024-25

Aim:-To perform Exploratory Data Analysis using Apache Spark and Pandas.

Theory:-

1. What is Apache Spark and how does it work?

Apache Spark is an open-source distributed computing system used for big data processing and analytics. It is designed for speed and ease of use, providing APIs in Python, Java, Scala, and R. Spark operates in-memory, meaning it processes data much faster than traditional disk-based engines like Hadoop MapReduce.

How Spark Works:

- Spark uses a cluster of machines to process data in parallel.
- It performs computations in memory using Resilient Distributed Datasets (RDDs) or DataFrames.
- Spark jobs are divided into stages and tasks, which are executed by the Spark engine across worker nodes.
- It supports multiple components like Spark SQL, MLlib (for machine learning), GraphX, and Spark Streaming.

2. How is data exploration done in Apache Spark? Explain the steps.

Data exploration in Apache Spark involves examining and summarizing the main characteristics of data, often using visual methods. Below are the general steps:

Step 1: Loading the Data

- Use SparkSession.read to load datasets in formats like CSV, JSON, or Parquet into a DataFrame.
  Example:

df = spark.read.csv("data.csv", header=True, inferSchema=True)

Step 2: Viewing Basic Information

- Use df.show() to preview rows.
- Use df.printSchema() to check the data types of each column.

Step 3: Summary Statistics

- Use df.describe().show() to get count, mean, stddev, min, and max of numerical columns.

Step 4: Handling Missing or Duplicate Values

- Use functions like dropna(), fillna() for missing values and dropDuplicates() for removing duplicates.

Step 5: Filtering and Grouping

- Use filter() or where() to filter rows based on conditions.
- Use groupBy().agg() to perform group-wise aggregation.

Step 6: Visualizing Insights (via Pandas)

- Convert Spark DataFrame to Pandas using df.toPandas() for visualization with matplotlib or seaborn.
  Example:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
pandas_df = df.toPandas()

sns.histplot(pandas_df['column_name'])

plt.show()
```

Conclusion:

Exploratory Data Analysis (EDA) using Apache Spark and Pandas allows for scalable and efficient handling of large datasets. Apache Spark provides a robust backend for data processing, while Pandas and visualization libraries offer detailed insights into data patterns. Together, they form a powerful toolkit for data scientists to clean, summarize, and understand data effectively.