Experiment No: 9

Aim: To perform Exploratory data analysis using Apache Spark and Pandas

Theory:

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

    1. **Apache Spark** is a powerful, open-source distributed computing framework specifically built for processing large volumes of data quickly and efficiently.
    2. It enables parallel data processing across a cluster of computers, allowing it to handle massive datasets that would be too large for a single machine.
    3. Unlike traditional MapReduce in Hadoop, Spark leverages **in-memory computation**, which drastically improves execution speed for iterative algorithms and complex workflows.
    4. Spark offers high-level APIs in multiple languages such as **Python, Java, Scala, and R**, making it accessible to a wide range of developers.
    5. It consists of several integrated components, including **Spark SQL** for structured data processing, **MLlib** for machine learning, **GraphX** for graph computation, and **Spark Streaming** for real-time data streams.
    6. Its resilient distributed dataset (RDD) and DataFrame abstractions allow for both fault tolerance and flexibility in managing unstructured or structured data.
    7. Spark's execution engine dynamically optimizes queries and manages tasks across distributed systems, making it ideal for ETL jobs, interactive queries, machine learning, and more.

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

    1. Step 1: Importing Data – Spark reads data from multiple sources like CSV, JSON, Parquet, or databases and loads it into DataFrames for processing.
    2. Step 2: Data Preprocessing – Clean the dataset by managing null values, correcting column types, and resolving duplicates or errors.
    3. Step 3: Data Manipulation – Perform operations such as filtering, joining, grouping, and aggregation to derive insights.
    4. Step 4: Integration for Visualization – While Spark lacks built-in plotting tools, it can be integrated with libraries like Matplotlib, Seaborn, or Power BI for visualization.
    5. Step 5: Statistical Summary – Use Spark functions to calculate descriptive statistics (mean, std. dev, median, etc.) for a quick overview of distributions.
    6. Step 6: Sampling & Data Inspection – Use .show() or .sample() to inspect data subsets and validate assumptions before deeper analysis.

Conclusion:

Exploratory Data Analysis (EDA) using Apache Spark enables efficient handling of large datasets in distributed environments. Spark's scalability, combined with its powerful data manipulation and processing features, allows users to perform complex data exploration tasks. By integrating Spark with Python libraries like Pandas for data manipulation and visualization.