Name: Shivam.R.Prajapati **Div: D15C** **Roll No:41**

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?

Soln:

- Apache Spark is an advanced, open-source analytics engine designed for large-scale data processing.
- It originated from a research project at UC Berkeley and has since become a widely-used tool for handling complex data tasks across various industries.
- When traditional tools like Excel, SQL, or even Pandas fall short — especially with massive data sizes or distributed environments — Apache Spark steps in to provide speed, scalability, and reliability.
- **Main Features of Apache Spark:**
    1. **Category**: Big Data Framework
    2. **Purpose**: Efficiently processes large datasets beyond the capacity of a single machine
    3. **Execution Model**: Distributes and processes data tasks across multiple machines in a cluster
    4. **Core Advantages**: Speed through in-memory processing, parallel computation, and fault tolerance
- **Working of Apache Spark:**
    - **Cluster-Based Operation:**
        - Spark runs on a cluster setup. One machine plays the role of the driver (the coordinator), and the others are worker nodes (executors). The driver breaks down tasks and distributes them to workers, who process data in parallel.
    - **RDD – The Building Block:**
        - Spark uses Resilient Distributed Datasets (RDDs) to store and operate on data. RDDs divide data into parts and spread them across the cluster. This division allows Spark to process data in parallel and recover easily from failures, making it both efficient and fault-tolerant.
    - **In-Memory Speed:**

    - Unlike older systems (e.g., Hadoop MapReduce) that constantly read/write from disk, Spark keeps data in RAM during processing. This in-memory approach leads to major performance boosts — up to 100 times faster in some scenarios.
  - **Lazy Execution Model:**
    - Spark uses lazy evaluation, meaning it doesn't process data until an action (like show() or collect()) is triggered. This approach helps it optimize execution plans and skip unnecessary steps.
  - **Multi-Language API Support:**
    - Developers can use Spark with popular languages like Python (via PySpark), Scala, Java, and R. This flexibility makes it suitable for different teams — from data scientists to backend engineers.

2. How is Data Exploration Done in Apache Spark?

Soln:

- Exploratory Data Analysis (EDA) is the first step in any data science or machine learning project. It's like getting to know your data — understanding what's inside, spotting errors, identifying trends, and deciding how to clean or transform it.
- With Spark, EDA can be done on huge datasets that don't fit into memory using a distributed approach. Here's a deeper breakdown of how EDA is carried out using Apache Spark:
- **Steps to perform EDA On Apache Spark:**

  Step 1: Initializing Spark Session:
  Initializing Spark Session then A SparkSession is started, which serves as the entry point for working with DataFrames and datasets in Spark.

  Step 2: Reading and Loading Data:
  Data is loaded from various formats (CSV, JSON, Parquet, etc.) into Spark DataFrames. These DataFrames are distributed collections of data, similar to tables.

  Step 3: Data Inspection and Schema Exploration:
  We examine the structure of the data using commands to display column names, data types, row counts, and sample records.

  Step 4: Data Cleaning:

This involves handling missing values, fixing data types, removing duplicates, and filtering out invalid records. This step ensures that the data is accurate and usable.

Step 5: Descriptive Statistics:
We calculate summary statistics like mean, median, standard deviation, min, and max values to understand the distribution of data.

Step 6: Grouping and Aggregation:
Data is grouped by categories and aggregated to analyze trends and patterns across different segments.

Step 7: Filtering and Sorting:
Specific subsets of the data are extracted by applying filter conditions and sorting values for deeper insights.

Step 8: Data Sampling and Conversion:
For visualization or detailed analysis, a small sample of the Spark DataFrame is converted into a Pandas DataFrame.

CONCLUSION:

This experiment aimed to understand how Exploratory Data Analysis (EDA) can be performed using both Apache Spark and Pandas. Spark, with its distributed and in-memory processing, is ideal for analyzing large-scale datasets, while Pandas is better suited for smaller, quick analysis tasks. We explored the step-by-step EDA process—loading, inspecting, cleaning, summarizing, and analyzing data. Each tool serves a specific purpose, and together they offer flexibility and efficiency across data sizes. Understanding how these tools work prepares data professionals to handle diverse analytical challenges with the right approach.