Experiment 9

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 a fast, open-source engine for big data processing. It's designed to handle everything from batch jobs to real-time streams, machine learning, and even graph-based data all in one place. Unlike the older Hadoop MapReduce, Spark processes data in-memory, which makes it way quicker for things like iterative algorithms or complex transformations.

It plays well with languages like Python, Scala, Java, and SQL, and can be run on cluster managers like YARN, Kubernetes, or just a standalone setup.

Main components of Spark:

- **Spark Core** – Handles the basic functions like task scheduling and memory management.
- **Spark SQL** – Lets you use SQL queries and DataFrames.
- **Structured Streaming** – For live data streams.
- **MLlib** – Machine learning tools.
- **GraphX** – For graph-based computations.

How it works

- A **driver program** definies your app's logic and managing execution.
- A **cluster manager** divides computing resources across machines.
- **Executors** run the actual tasks.
- It keeps data in memory when possible for speed.
- APIs like RDDs and DataFrames give you different levels of control and optimization.

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

- **Start a Spark session** – This initializes the Spark environment.
- **Load your data** – Usually from CSV, JSON, databases, or Parquet files.
- **Peek at the data** – Use .show() to check out the first few rows.
- **Inspect the schema** – Understand what columns exist and what types they are.
- **Get summary stats** – Use .describe() to view metrics like count, mean, min, max.
- **Check for missing data** – Null values can mess up your model later, so identify them early.
- **Group or filter** – Group by categories, count frequencies, or filter rows to dig deeper.
- **Drill down further** – Use queries or filters to explore patterns or issues.

Conclusion: This experiment showed how Apache Spark and Pandas can be used to explore data efficiently. We looked at structure, missing values, summary stats, and groupings. Spark's computing and scalability make it ideal for handling large datasets during the initial analysis phase.