Name : Ganesh Gupta Roll NO : 13 Div : D15C

## Experiment 9

**Aim:** To perform exploratory data analysis using Apache Spark and Pandas.

**Theory:**

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

Ans) Apache Spark is an open-source data processing engine designed for speed and scalability. It supports batch processing, real-time streaming, machine learning, and graph analytics in one unified platform.

Unlike Hadoop MapReduce, Spark processes data in memory, making it much faster for iterative and complex tasks. It supports languages like Python, Scala, Java, and SQL, and can run on various cluster managers like YARN, Kubernetes, or standalone. Spark is widely used in big data environments for its performance, ease of use, and flexibility.

Core Components of Apache Spark are:

- **Spark Core**: Core engine (scheduling, memory, etc.)
- **Spark SQL**: For SQL queries and DataFrames
- **Structured Streaming**: Real-time processing
- **MLlib**: Machine learning
- **GraphX**: Graph processing

It works in the following ways:

1. **Driver Program**: Starts the app, defines the workflow (DAG), and manages coordination.
2. **Cluster Manager**: Allocates resources (Standalone, YARN, Mesos, Kubernetes).
3. **Executors**: Run tasks on worker nodes and store data.
4. **In-Memory Computing**: Keeps data in memory for fast processing.
5. **APIs**: Uses RDDs for low-level tasks, DataFrames/Datasets for optimized, SQL-like operations.

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

Ans) Data exploration in Apache Spark is done as follows:

1. **Initialize Spark Session**: Start a Spark session to enable interaction with Spark's features and APIs.
2. **Load Data**: Import your dataset into Spark from sources like CSV, JSON, databases, or Parquet files.
3. **View Data Sample**: Look at the first few rows to get a quick sense of the data content and format.
4. **Check Schema**: Examine the structure of the dataset, including column names and data types.
5. **Summary Statistics**: Generate basic statistics like mean, count, min, and max for numerical columns to understand distributions.
6. **Check for Missing Values**: Identify columns that have null or missing values to assess data quality.
7. **Value Counts / Grouping**: Group data by specific columns to analyze category frequencies or distributions.
8. **Filter and Query Data**: Apply filters or queries to focus on specific subsets of the data for deeper analysis.

**Conclusion:** In this experiment, we learned how to perform exploratory data analysis using Apache Spark and Pandas. By examining the data's structure, statistics, missing values, and groupings, we gain key insights to guide cleaning, feature selection, and modeling. Spark's speed and scalability make it ideal for exploring large datasets efficiently.