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

Experiment No:10

AIM: To perform Batch and Streamed Data Analysis using Apache Spark.

THEORY:

1. What is Streaming? Explain Batch and Stream Data.

Soln:

Streaming in Data Science:

- Streaming refers to processing and analyzing data continuously as it is being generated.
- In data science, streaming is used when real-time insights or actions are needed.
- Instead of waiting for all the data to be collected (like in batch processing), data is processed immediately when it arrives.
- Example:
    - A fraud detection system in a bank that checks every transaction as it happens.
    - A YouTube recommendation system that updates based on your recent activity, not just your old data.
- Tools used:
    - Apache Spark Streaming
    - Apache Kafka
    - Flink
    - Python with streaming APIs (like PySpark Streaming)

Batch Data in Data Science:

- Batch data means that data is collected over a period of time and then processed all at once.
- It is the most common method used in data science for training machine learning models, data cleaning, or creating visualizations.
- In batch processing, real-time results are not important — accuracy and depth are prioritized.
- Example:
    - A telecom company collects all call records from the entire day, and processes them at night to detect call patterns.
    - A retailer analyzes the past 3 months of sales data to forecast future demand.
- Tools used:
    - Pandas, NumPy (Python)
    - Apache Spark (Batch mode)
    - SQL databases
    - Hadoop

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

Stream Data in Data Science:

- Stream data is a type of data that keeps coming in continuously, like a live feed.
- It is unbounded — meaning it doesn't stop — and needs to be processed in small portions, not all at once.
- In data science, it is mostly used for:
    - Real-time predictions
    - Monitoring systems
    - Alerting on anomalies (e.g. sudden spike in temperature from sensors)
- Examples:
    - Live GPS data from delivery trucks
    - Tweets posted every second
    - Stock market price updates

2. How Does Data Streaming Take Place Using Apache Spark?

Soln:

- Apache Spark offers a powerful feature called **Spark Streaming**, and its improved version **Structured Streaming**, to handle **real-time data** as it arrives.
- **Working:**
    - **Live Data Source**: Spark connects to continuously updating sources like Apache Kafka, sockets, or folders that get new files in real-time.
    - **Streaming Engine**: Even though it's called streaming, Spark breaks the incoming data into small time-based chunks, known as micro-batches. Each small batch is processed every few seconds.
    - **Processing Logic**: Just like with normal data, you can filter, group, or aggregate the data as it comes in.
    - **Output Destination**: Once processed, the results can be saved to a database, file system, or streamed to dashboards or alert systems for quick insights.
    - **Unified Programming Model**: What makes Spark special is that you can use the same code and logic whether you're dealing with batch data or streaming data — making it super convenient for data scientists and engineers.
- **Steps for Batch Data Analysis using Apache Spark**

1. **Start the Spark Environment:**
    a. Set up Spark either locally, on the cloud, or using notebooks like **Jupyter** or **Databricks** to begin your analysis.
2. **Load a Batch Dataset:**
    a. Import a complete dataset (e.g., CSV, JSON) that contains historical data with a defined beginning and end.
3. **Understand the Data:**
    a. Inspect column names, types, and sample values to get a sense of the dataset and identify what needs fixing.
4. **Clean the Data**

   a. Fix issues like missing entries, rename confusing column names, correct data types, and remove repeated rows.
5. **Perform Data Transformations**
   a. Use filters, groupings, and aggregations (like total sales or average rating) to get meaningful insights from the data.
6. **Save or Visualize the Results**
   a. Output can be written to files, shown in the console, or used to create visualizations using dashboards.

- **Steps for Streamed Data Analysis using Apache Spark**
  1. **Set Up Structured Streaming:** Start a SparkSession configured for Structured Streaming, Spark's modern method for processing live data.
  2. **Connect to a Real-Time Data Source:** Hook Spark up to sources like Kafka, sockets, or directories where fresh data files are added regularly.
  3. **Define a Data Schema:** Since streaming data often lacks headers, you need to manually define what each field represents (e.g., time, value, ID).
  4. **Apply Logic While Data Flows:** As new data comes in, apply transformations like filtering by condition (e.g., values above a certain threshold), time-based grouping (e.g., every 10 seconds), or calculations (e.g., max, average).
  5. **Send Results to a Destination:** Output the continuously updated results to a console, file, database, or even live dashboards.
  6. **Keep an Eye on the Stream:** Monitor the pipeline for speed, data flow, and memory usage. Spark keeps the job running until you stop it manually.

CONCLUSION:

This experiment helps us understand two powerful ways of working with data using Apache Spark: batch and streamed processing. Batch analysis is well-suited for historical or static data, allowing deep dives and summary reports. On the other hand, streamed analysis enables real-time decision-making by processing live data as it arrives. Apache Spark handles both seamlessly using its unified framework and scalable infrastructure. By learning both approaches, we equip ourselves with the flexibility to tackle a wide variety of real-world data problems whether they require immediate insight or deep historical trends.