Name: Neeraj Rijhwani Roll No: 64 Class:D15C

Experiment 09

Aim:

To perform Exploratory data analysis using Apache Spark and Pandas

Introduction to Big Data and Spark:

Big Data

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is data with such large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also data but with huge size. The definition of big data is data that contains greater variety, arriving in increasing volumes and with more velocity. This is also known as the three Vs.

The three Vs of big data:

1. Volume
   The amount of data matters. With big data, you'll have to process high volumes of low-density, unstructured data. This can be data of unknown value, such as Twitter data feeds, clickstreams on a web page or a mobile app, or sensor-enabled equipment. For some organizations, this might be tens of terabytes of data. For others, it may be hundreds of petabytes.
2. Velocity
   Velocity is the fast rate at which data is received and (perhaps) acted on. Normally, the highest velocity of data streams directly into memory versus being written to disk. Some internet-enabled smart products operate in real time or near real time and will require real-time evaluation and action.
3. Variety
   Variety refers to the many types of data that are available. Traditional data types were structured and fit neatly in a relational database. With the rise of big data, data comes in new unstructured data types. Unstructured and semistructured data types, such as text, audio, and video, require additional preprocessing to derive meaning and support metadata.

Big Data Mining

Big data mining refers to the collective data mining or extraction techniques that are performed on large sets /volume of data or the big data. Big data mining is primarily done to extract and retrieve desired information or patterns from humongous quantities of data.

This is usually performed on a large quantity of unstructured data that is stored over time by an organization. Typically, big data mining works on data searching, refinement , extraction and comparison algorithms. Big data mining also requires support from underlying computing devices, specifically their processors and memory, for performing operations / queries on large amounts of data.

Big data mining techniques and processes are also used within big data analytics and business intelligence to deliver summarized targeted and relevant information, patterns and/or relationships between data, systems, processes and more.

Big Data Tools

Big Data requires a set of tools and techniques for analysis to gain insights from it. Big Data is an essential part of almost every organization these days and to get significant results through Big Data Analytics a set of tools is needed at each phase of data processing and analysis.

There are a few factors to be considered while opting for the set of tools i.e., the size of the datasets, pricing of the tool, kind of analysis to be done, and many more.

There are a number of big data tools available in the market such as Hadoop which helps in storing and processing large data, Spark helps in-memory calculation, Storm helps in faster processing of unbounded data, Apache Cassandra provides high availability and scalability of a database, MongoDB provides cross-platform capabilities, so there are different functions of every Big Data tool.

Here is the list of top 10 big data tools –

1. Apache Hadoop
2. Apache Spark
3. Flink
4. Apache Storm
5. Apache Cassandra
6. MongoDB
7. Kafka
8. Tableau
9. RapidMiner
10. R Programming

Spark

Apache Spark is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching and optimized query execution for fast queries against data of any size. Simply put, Spark is a fast and general engine for large-scale data processing.

The fast part means that it’s faster than previous approaches to work with Big Data like classical MapReduce. The secret for being faster is that Spark runs on memory (RAM), and that makes the processing much faster than on disk drives.

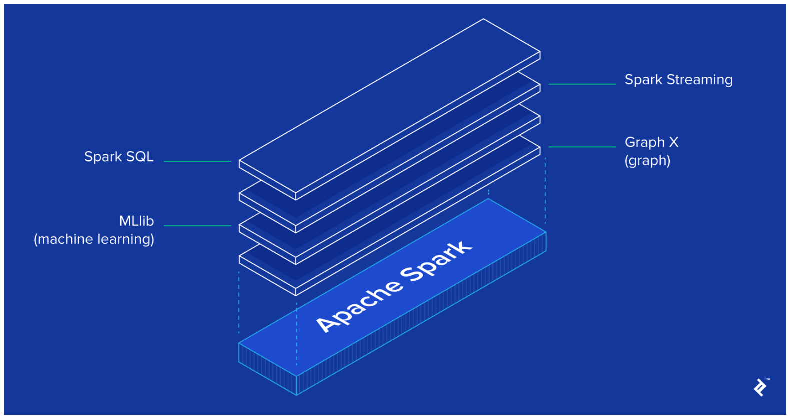
The general part means that it can be used for multiple things like running distributed SQL, creating data pipelines, ingesting data into a database, running Machine Learning algorithms, working with graphs or data streams, and much more.

Features of Spark

1. Fast processing – The most important feature of Apache Spark that has made the big data world choose this technology over others is its speed. Big data is characterized by volume, variety, velocity, and veracity which needs to be processed at a higher speed. Spark contains Resilient Distributed Dataset (RDD) which saves time in reading and writing operations, allowing it to run almost ten to one hundred times faster than Hadoop.
2. Flexibility – Apache Spark supports multiple languages and allows the developers to write applications in Java, Scala, R, or Python.
3. In-memory computing – Spark stores the data in the RAM of servers which allows quick access and in turn accelerates the speed of analytics.
4. Real-time processing – Spark is able to process real-time streaming data. Unlike MapReduce which processes only stored data, Spark is able to process real-time data and is, therefore, able to produce instant outcomes.
5. Better analytics – In contrast to MapReduce that includes Map and Reduce functions, Spark includes much more than that. Apache Spark consists of a rich set of SQL queries, machine learning algorithms, complex analytics, etc. With all these functionalities, analytics can be performed in a better fashion with the help of Spark.

The Spark framework includes:

1. Spark Core as the foundation for the platform
2. Spark SQL for interactive queries
3. Spark Streaming for real-time analytics
4. Spark MLlib for machine learning
5. Spark GraphX for graph processing



Spark Function and Libraries Used:

(Explain the library function used for EDA.)

pyspark.ml module

MLlib is Spark’s machine learning (ML) library. Its goal is to make practical machine learning scalable and easy. At a high level, it provides tools such as:

1. ML Algorithms: common learning algorithms such as classification, regression, clustering, and collaborative filtering
2. Featurization: feature extraction, transformation, dimensionality reduction, and selection
3. Pipelines: tools for constructing, evaluating, and tuning ML Pipelines
4. Persistence: saving and load algorithms, models, and Pipelines
5. Utilities: linear algebra, statistics, data handling, etc.

pyspark.sql module

PySpark SQL is a module in Spark which integrates relational processing with Spark's functional programming API. We can extract the data by using an SQL query language. We can use the queries same as the SQL language.

If you have a basic understanding of RDBMS, PySpark SQL will be easy to use, where you can extend the limitation of traditional relational data processing. Spark also supports the Hive Query Language, but there are limitations of the Hive database.

Important classes of Spark SQL and DataFrames:

- pyspark.sql.SparkSession Main entry point for DataFrame and SQL functionality.
- pyspark.sql.DataFrame A distributed collection of data grouped into named columns.
- pyspark.sql.Column A column expression in a DataFrame.
- pyspark.sql.Row A row of data in a DataFrame.
- pyspark.sql.GroupedData Aggregation methods, returned by DataFrame.groupBy().
- pyspark.sql.DataFrameNaFunctions Methods for handling missing data (null values).
- pyspark.sql.DataFrameStatFunctions Methods for statistics functionality.
- pyspark.sql.functions List of built-in functions available for DataFrame.
- pyspark.sql.types List of data types available.
- pyspark.sql.Window For working with window functions.

class pyspark.sql.DataFrame

It is a distributed collection of data grouped into named columns. A DataFrame is similar to the relational table in Spark SQL, can be created using various functions in SQLContext.

```
sqlContext.read.csv("...")
```

After creation of dataframe, we can manipulate it using the several domain-specific-languages (DSL) which are predefined functions of DataFrame. Let’s get started with the functions:

select(): The select function helps us to display a subset of selected columns from the entire dataframe we just need to pass the desired column names. Let’s print any three columns of the dataframe using select().

withColumn(): The withColumn function is used to manipulate a column or to create a new column with the existing column. It is a transformation function, we can also change the datatype of any existing column.

groupBy(): The groupBy function is used to collect the data into groups on DataFrame and allows us to perform aggregate functions on the grouped data. This is a very common data analysis operation similar to the groupBy clause in SQL.

orderBy(): The orderBy function is used to sort the entire dataframe based on the particular column of the dataframe. It sorts the rows of the dataframe according to column values. By default, it sorts in ascending order.

split(): The split() is used to split a string column of the dataframe into multiple columns. This function is applied to the dataframe with the help of withColumn() and select().

lit(): The lit function is used to add a new column to the dataframe that contains literals or some constant value.

when(): The when the function is used to display the output based on the particular condition. It evaluates the condition provided and then returns the values accordingly. It is a SQL function that supports PySpark to check multiple conditions in a sequence and return the value. This function similarly works as if-then-else and switch statements.

filter(): The filter function is used to filter data in rows based on the particular column values.

isNull()/isNotNull(): These two functions are used to find out if there is any null value present in the DataFrame. It is the most essential function for data processing. It is the major tool used for data cleaning.

Conclusion:

Thus, we have learnt what big data is, how Apache Spark is a great big data tool, and also learnt how to use pyspark libraries to preprocess a dataset, and perform EDA on the same in python.