AC – 28/03/2025
Item No. – 6.2 (N) (3a) Sem. IV

As Per NEP 2020

University of Mumbai

Syllabus for Basket of OE Vertical 3	
Faculty of Science	
Board of Studies in Data Science	
Second Year Programme	
Semester	**IV**
Title of Paper	**Credits**
Apache Spark	**2**
From the Academic Year	**2025-26**

Title of Paper: Apache Spark

Sr.No.	Heading	Particulars
1	**Description of the course: Including but not limited to:**	The Apache Spark Practical Course equips learners with the fundamentals and advanced features of Apache Spark for big data processing. It covers Spark architecture, Data Frames, SQL optimization, file formats (CSV, Parquet), data processing techniques, streaming analytics, and machine learning with MLlib. Through real-world projects, learners gain hands-on experience in filtering, aggregating, and analyzing data, preparing them to tackle real-world big data challenges effectively.
2	**Vertical :**	Open Elective
3	**Type :**	Practical
4	**Credit**:	2 credits (1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted**:	50 Marks
7	**Course Objectives (CO) :** CO1: **Understand the Fundamentals**: Gain a comprehensive understanding of Apache Spark's core concepts, architecture, and the Big Data problem it addresses. CO2: **Perform Data Manipulation**: Learn how to load, transform, and manipulate data using Apache Spark DataFrames and RDDs. CO3: **Master Aggregations and Grouping**: Develop the ability to perform complex aggregations and grouping operations to derive meaningful insights from data. CO4: **Execute Advanced Operations**: Acquire skills to perform advanced data operations such as joins, unions, and Spark SQL queries to manipulate and analyze large datasets.	
8	**Course Outcomes (OC) :** OC1: **Create and Manipulate DataFrames**: Load various data formats into Spark DataFrames, perform basic and advanced transformations, and manipulate data effectively. OC2: **Implement Aggregations and Grouping**: Demonstrate proficiency in using Spark's aggregation and grouping functions to analyze and summarize data, including handling null values and performing window functions. OC3: **Perform Joins and Unions**: Execute different types of joins and unions on DataFrames to combine and analyze data from multiple sources. OC4: **Leverage Spark SQL**: Will have the capability to create temporary views, write and execute SQL queries, and use Spark SQL for data manipulation and analysis.	

9	**Modules:**	
	Module 1:	**30 Hrs**
	1. Introduction to Apache Spark, Apache Spark Philosophy, The Bigdata Problem, History of Spark, Basic Architecture, Spark's Language APIs, Spark's APIs, Spark Session, Data Frames, Transformations, Actions, Spark UI. 2. Data Loading and Basic Transformations a. Load a CSV file into a Spark DataFrame and display the schema. b. Load a JSON file into a Spark DataFrame and show the first 10 rows. c. Load a Parquet file and count the number of rows. d. Load a text file and count the number of lines. e. Filter a DataFrame based on a specific column value. f. Select specific columns from a DataFrame and create a new DataFrame. g. Add a new column to a DataFrame using withColumn. h. Rename a column in a DataFrame using withColumnRenamed. i. Drop a column from a DataFrame using drop. j. Sort a DataFrame by a specific column in ascending and descending order. k. Perform a distinct count of a column. l. Convert a column's data type (e.g., string to integer). m. Calculate the length of a string column. n. Concatenate two string columns into a new column. o. Convert all column names to lowercase. p. Replace null values in a column with a specific value using fillna. q. Remove rows with null values using dropna. r. Create a new DataFrame with only the first 100 rows. s. Create a new DataFrame by sampling 10% of the original data. t. Use explode to flatten an array column into rows. 3. Aggregations and Grouping a. Group data by a column and calculate the count of rows in each group. b. Group data and calculate the sum of a numeric column. c. Group data and calculate the average of a numeric column. d. Group data and find the minimum and maximum values of a numeric column. e. Use agg to perform multiple aggregations at once. f. Use groupBy with multiple columns. g. Calculate the standard deviation of a numeric column. h. Calculate the variance of a numeric column. i. Use pivot to transform rows into columns.	

	j. Use rollup to generate subtotals. k. Use cube to generate all possible combinations of subtotals. l. Calculate the median of a numeric column (using approximate percentile). m. Calculate quartiles of a numeric column. n. Find the most frequent value in a column. o. Calculate the cumulative sum of a column within each group. p. Calculate the running average of a column within each group. q. Use window functions for time-based aggregations (if applicable with a date/time column). r. Calculate the rank of rows within each group. s. Calculate the dense rank of rows within each group. t. Calculate the percentage of total for each group. 4. Joins and Unions a. Perform an inner join between two DataFrames. b. Perform a left outer join between two DataFrames. c. Perform a right outer join between two DataFrames. d. Perform a full outer join between two DataFrames. e. Perform a self-join on a DataFrame. f. Perform a cross join (cartesian product) between two DataFrames. g. Perform a left semi join to filter the left DataFrame. h. Perform a left anti join to find rows only in the left DataFrame. i. Union two DataFrames with the same schema. j. Union two DataFrames with different schemas (using unionByName). k. Join DataFrames based on multiple columns. l. Join a DataFrame with a broadcast variable. m. Use broadcast join to improve performance of a large-small join. n. Perform a natural join. o. Perform a join using complex join conditions.	
	Module 2:	**30 Hrs**
	1. **Spark SQL** a. Create a temporary view from a DataFrame. b. Execute a simple SQL query on the temporary view. c. Use spark.sql to filter data using SQL. d. Use spark.sql to perform aggregations using SQL. e. Use spark.sql to perform joins using SQL. f. Create a managed table in the default database. g. Create an unmanaged table pointing to a specific file location. h. Insert data into a managed table from a DataFrame. i. Query metadata about a table using DESCRIBE TABLE. j. Create a database and switch to it.	

	k. Use CASE WHEN statements in SQL queries. l. Use window functions in SQL queries. m. Create and use a user defined function (UDF) in spark SQL. n. Use subqueries inside of SQL queries. o. Use LIMIT and OFFSET to paginate results in SQL. 2. **Data Manipulation and Transformations** a. Apply a user-defined function (UDF) to a column. b. Use map to transform each row of a DataFrame. c. Use flatMap to transform each row into multiple rows. d. Use filter with a lambda function. e. Use reduce to aggregate data. f. Use foreach to iterate over rows (for debugging purposes). g. Use repartition to change the number of partitions. h. Use coalesce to decrease the number of partitions. i. Use partitionBy when writing data to disk. j. Use sortWithinPartitions to sort data within partitions. k. Use collect to retrieve all rows as a list (use cautiously). l. Use take to retrieve the first N rows as a list. m. Use foreachPartition to process data within each partition. n. Use mapPartitions to transform data within each partition. o. Use struct to create a complex column. 3. **Advanced Spark Concepts** a. Use broadcast variables to share data across executors. b. Use accumulators to track metrics. c. Use persist or cache to store DataFrames in memory. d. Experiment with different storage levels for caching. e. Optimize Spark jobs by understanding the execution plan. f. Analyze the Spark UI to identify performance bottlenecks. g. Use Spark MLlib for basic machine learning tasks (e.g., linear regression). h. Use Spark GraphFrames for basic graph analysis. i. Use foreachBatch with structured streaming (if supported in community edition). j. Use checkpointLocation to make structured streaming fault-tolerant. k. Create a custom partitioner. l. Handle skewed data using salting or other techniques. m. Use Spark's built-in functions for date and time manipulations. n. Create and use a custom schema. o. Use explain to understand the query execution plan. p. Write data to different file formats (CSV, JSON, Parquet). q. Read and write data to/from cloud storage (if supported in community edition).	

	r. Use when and otherwise to create conditional columns. s. Implement a custom aggregator. t. Write data to multiple output locations based on a column value. u. Use sequence function to generate an array of numbers. v. Use array_contains to check if an array column contains a value. w. Use aggregate function for complex aggregations on array columns. x. Use posexplode to get the position of elements in an exploded array. y. Use zip_with to combine two arrays. z. Implement a sliding window aggregation. aa. Perform approximate distinct count using approx_count_distinct. bb. Use monotonically_increasing_id to generate unique IDs. cc. Use least and greatest to find the minimum and maximum of multiple columns. dd. Use regexp_extract to extract parts of a string using regular expressions. ee. Use regexp_replace to replace parts of a string using regular expressions. ff. Use date_add and date_sub to manipulate date columns. gg. Use datediff to find the difference between two dates. hh. Use to_date and to_timestamp to convert strings to date/timestamp types. ii. Debug spark applications using spark logs and spark UI.	
10	**Textbooks:** 1. Spark-the Definitive Guide, Bill Chambers, Matei Zaharia, 1st Edition, O'Reilly 2. Learning Spark, Jules S. Damji, Brooke Wenig, Tathagata Das & Denny Lee, 2nd Edition, O'Reilly	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.	**Format of Question Paper:** 30 marks practical exam of 2 hours duration

14	**Format of Question Paper: Duration 2 hours. A certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Sd/-	**Sd/-**	**Sd/-**
Sign of the BOS Chairman	**Sign of the Offg. Associate Dean**	**Sign of the Offg. Dean**
Dr. Srivaramangai R	**Dr. Madhav R. Rajwade**	**Prof. Shivram S. Garje**
BOS in Data Science	**Faculty of Science & Technology**	**Faculty of Science & Technology**