

## Assignment 7

### Big Data Analytics and Visualization

#### Lab 7 : Apache Spark Assignment

**Aim:** To perform data processing tasks in Apache Spark using Databricks, including reading files into RDDs, applying transformations like map and filter, and using actions like reduce for data manipulation and aggregation.

#### Theory:

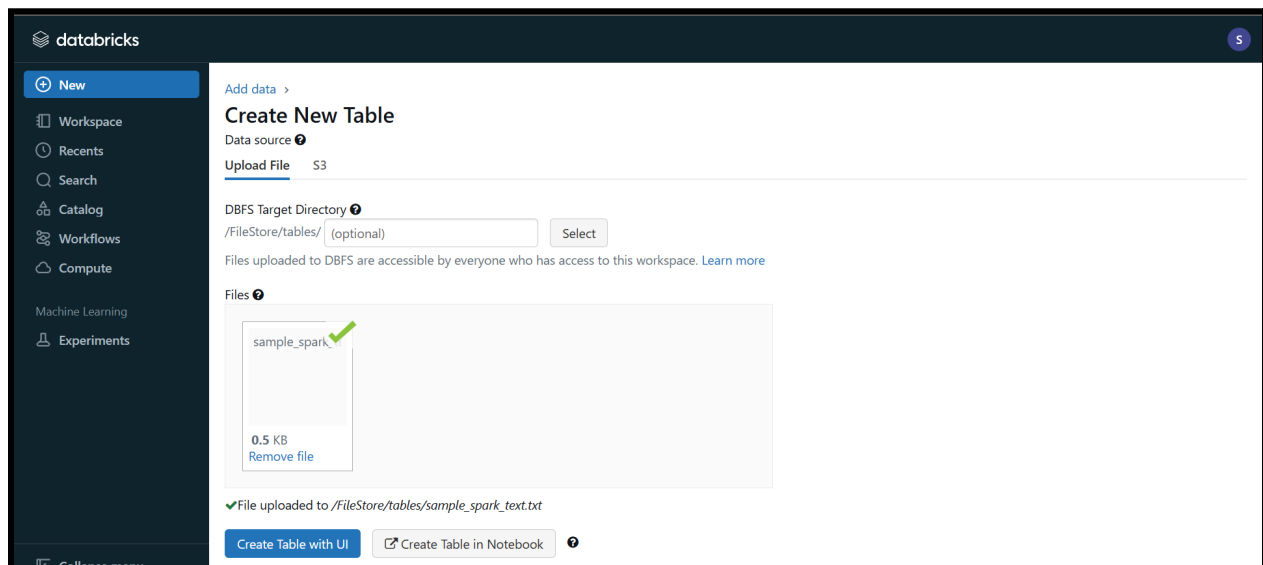
Apache Spark is a powerful distributed computing system that allows for processing large datasets across clusters. It supports a range of transformations and actions on Resilient Distributed Datasets (RDDs), enabling efficient data processing. Databricks provides a collaborative environment with cloud infrastructure, making it an ideal platform for performing big data operations with Spark.

Resilient Distributed Datasets (RDDs) are Spark's core abstraction, providing fault-tolerant, parallel data structures that allow operations on large data sets. RDDs support two main types of operations:

1. Transformations (e.g., map, filter): These create new RDDs from existing ones, enabling data transformation without modifying the original dataset.
2. Actions (e.g., reduce, collect): These trigger computation and return results.

#### Steps:

**Add the .txt and .csv files to FileStore in DataBricks.**

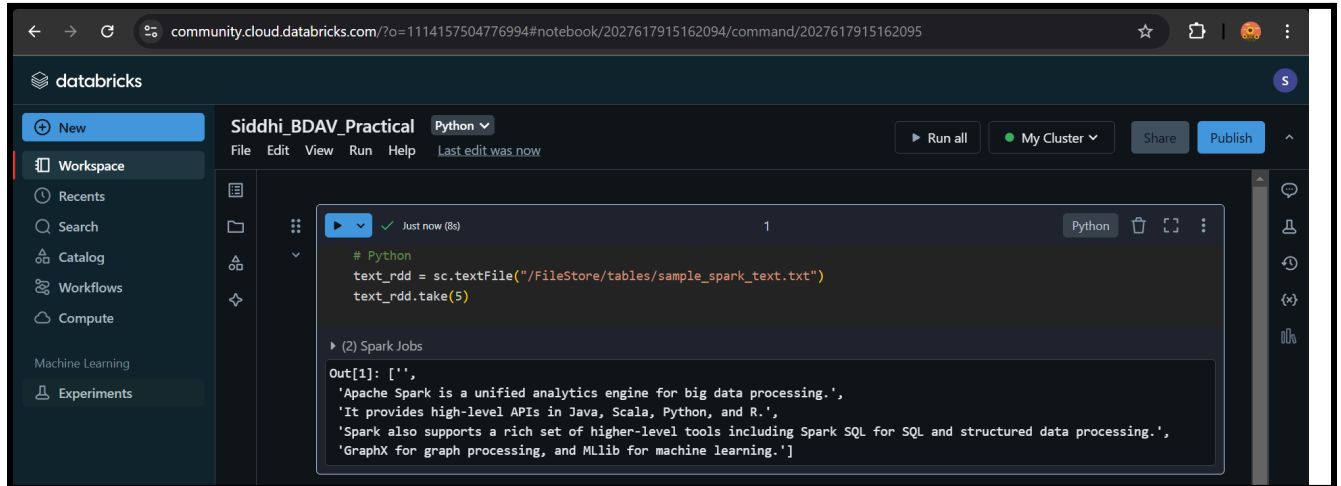


1. Read the .txt file into RDD

**Code:**

```
text_rdd = sc.textFile("/FileStore/tables/sample_spark_text.txt")  
text_rdd.take(5)
```

**Output:**



The screenshot shows a Databricks notebook interface. The notebook is titled "Siddhi\_BDAV\_Practical" and is in "Python" mode. The code cell contains the following Python code:

```
# Python  
text_rdd = sc.textFile("/FileStore/tables/sample_spark_text.txt")  
text_rdd.take(5)
```

The output of the code is displayed below the code cell, showing the first five lines of the text file:

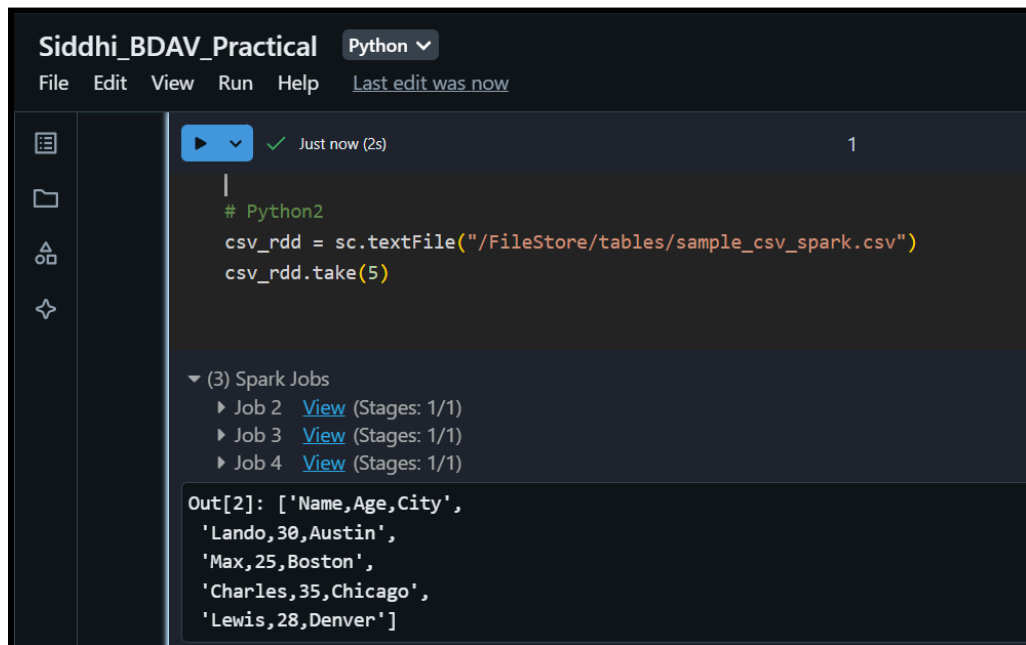
```
Out[1]: ['',  
'Apache Spark is a unified analytics engine for big data processing.',  
'It provides high-level APIs in Java, Scala, Python, and R.',  
'Spark also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing.',  
'GraphX for graph processing, and MLlib for machine learning.']
```

2. Read CSV file into RDD

**Code:**

```
csv_rdd = sc.textFile("/FileStore/tables/sample_csv_spark.csv")  
csv_rdd.take(5)
```

**Output:**



The screenshot shows a Databricks notebook interface. The notebook is titled "Siddhi\_BDAV\_Practical" and is in "Python" mode. The code cell contains the following Python code:

```
# Python2  
csv_rdd = sc.textFile("/FileStore/tables/sample_csv_spark.csv")  
csv_rdd.take(5)
```

The output of the code is displayed below the code cell, showing the first five lines of the CSV file:

```
Out[2]: ['Name, Age, City',  
'Lando, 30, Austin',  
'Max, 25, Boston',  
'Charles, 35, Chicago',  
'Lewis, 28, Denver']
```

3. Display a limited number of record

**Code:**

```
csv_rdd.take(10) # Display first 10 records
```

**Output:**



Siddhi\_BDAV\_Practical Python

File Edit View Run Help Last edit was now

Run all My Cluster Share

Just now (2s) 1 Python

```
csv_rdd.take(10) # Display first 10 records
```

▶ (5) Spark Jobs

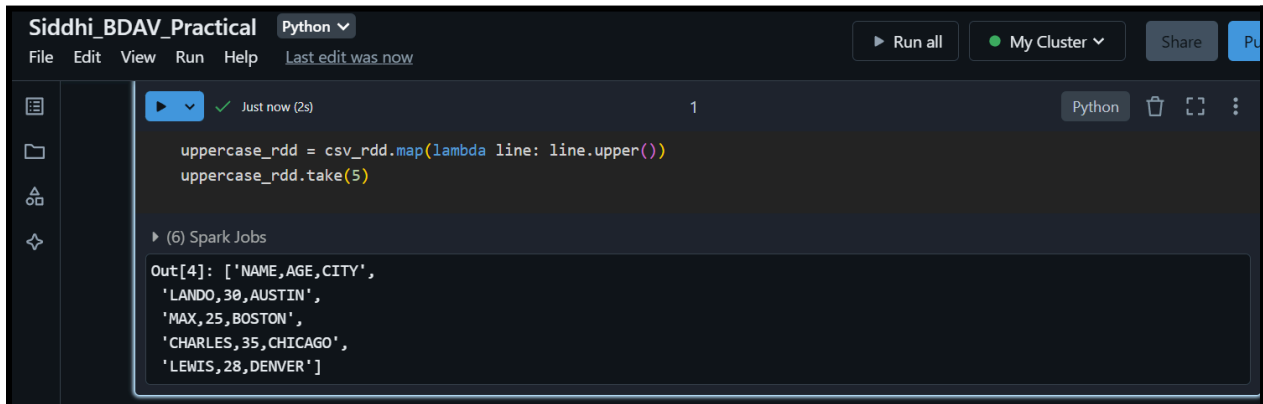
```
Out[3]: ['Name, Age, City',  
'Lando, 30, Austin',  
'Max, 25, Boston',  
'Charles, 35, Chicago',  
'Lewis, 28, Denver',  
'Oscar, 22, El Paso',  
'Frank, 32, Fresno',  
'Grace, 27, Galveston']
```

4. Convert data into uppercase using the map function

**Code:**

```
uppercase_rdd = csv_rdd.map(lambda line: line.upper())  
uppercase_rdd.take(5)
```

**Output:**



Siddhi\_BDAV\_Practical Python

File Edit View Run Help Last edit was now

Run all My Cluster Share

Just now (2s) 1 Python

```
uppercase_rdd = csv_rdd.map(lambda line: line.upper())  
uppercase_rdd.take(5)
```

▶ (6) Spark Jobs

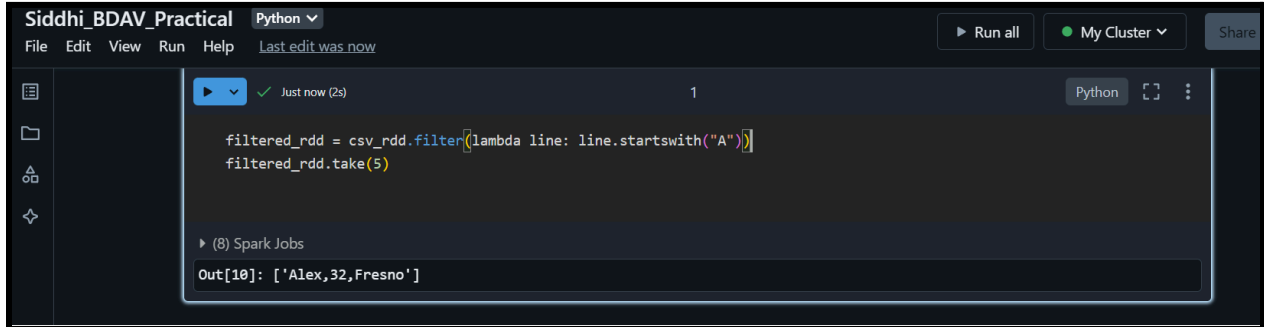
```
Out[4]: ['NAME, AGE, CITY',  
'LANDO, 30, AUSTIN',  
'MAX, 25, BOSTON',  
'CHARLES, 35, CHICAGO',  
'LEWIS, 28, DENVER']
```

5. Display data Start with A using the filter function

**Code:**

```
filtered_rdd = csv_rdd.filter(lambda line: line.startswith("A"))  
filtered_rdd.take(5)
```

**Output:**



The screenshot shows a Jupyter Notebook titled 'Siddhi\_BDAV\_Practical' with a Python kernel. The code cell contains the following code:

```
filtered_rdd = csv_rdd.filter(lambda line: line.startswith("A"))  
filtered_rdd.take(5)
```

The output of the code is displayed below the code cell, showing the result of the filter function:

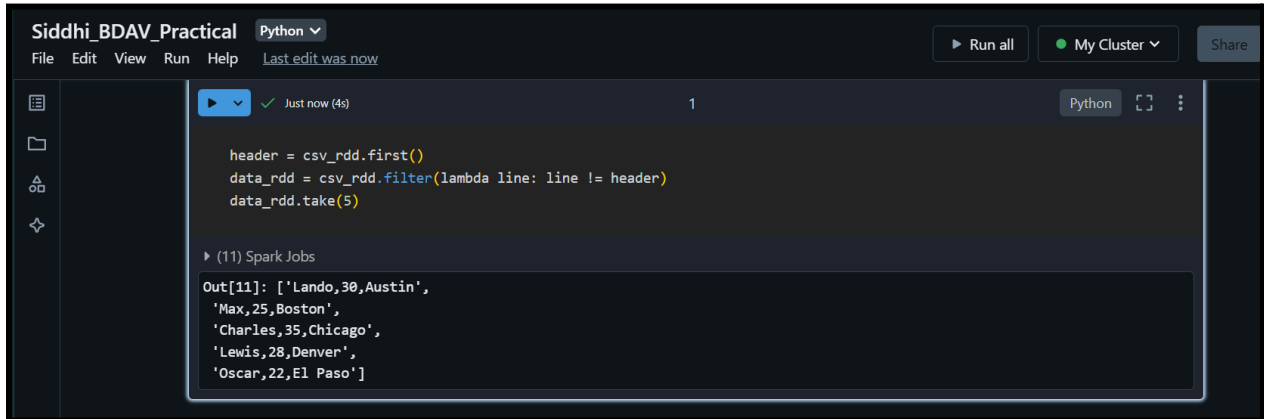
```
Out[10]: ['Alex,32,Fresno']
```

6. Skip header from CSV file

**Code:**

```
header = csv_rdd.first()  
data_rdd = csv_rdd.filter(lambda line: line != header)  
data_rdd.take(5)
```

**Output:**



The screenshot shows a Jupyter Notebook titled 'Siddhi\_BDAV\_Practical' with a Python kernel. The code cell contains the following code:

```
header = csv_rdd.first()  
data_rdd = csv_rdd.filter(lambda line: line != header)  
data_rdd.take(5)
```

The output of the code is displayed below the code cell, showing the result of the filter function:

```
Out[11]: ['Lando,30,Austin',  
'Max,25,Boston',  
'Charles,35,Chicago',  
'Lewis,28,Denver',  
'Oscar,22,El Paso']
```

## 7. Read multiple CSV files into RDD

### Code:

```
multi_csv_rdd = sc.textFile("/FileStore/tables/*.csv")  
multi_csv_rdd.take(5)
```

### Output:



```
Siddhi_BDAV_Practical Python  
File Edit View Run Help Last edit was now  
Run all My Cluster Share  
Just now (3s) 1 Python  
multi_csv_rdd = sc.textFile("/FileStore/tables/*.csv")  
multi_csv_rdd.take(5)  
(12) Spark Jobs  
Out[12]: ['Name, Age, City',  
'Lando, 30, Austin',  
'Max, 25, Boston',  
'Charles, 35, Chicago',  
'Lewis, 28, Denver']
```

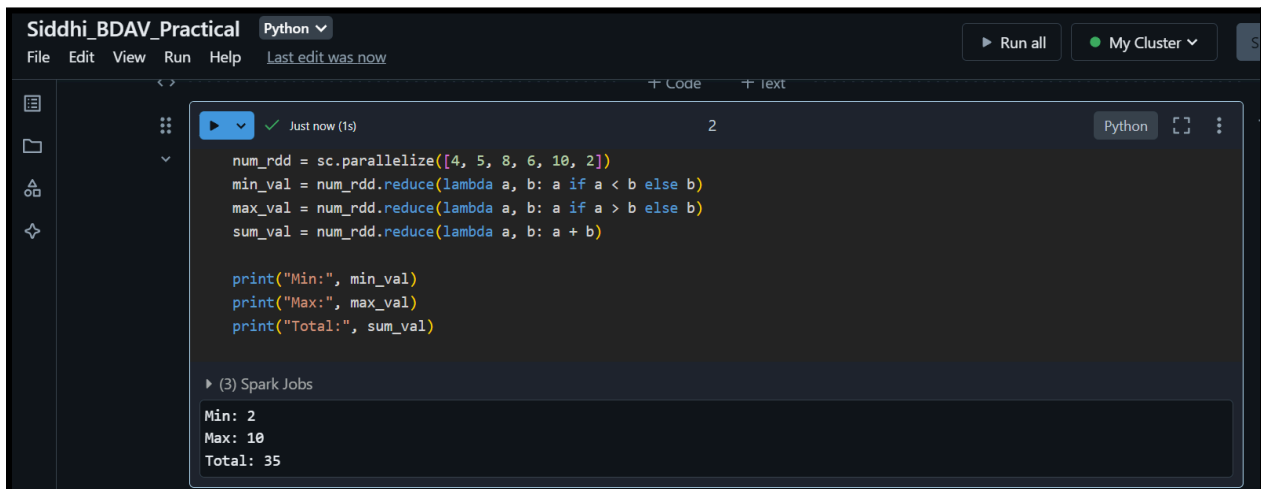
## 8. Reduce a list – Calculate min, max, and total of elements

### Code:

```
num_rdd = sc.parallelize([4, 5, 8, 6, 10, 2])  
min_val = num_rdd.reduce(lambda a, b: a if a < b else b)  
max_val = num_rdd.reduce(lambda a, b: a if a > b else b)  
sum_val = num_rdd.reduce(lambda a, b: a + b)
```

```
print("Min:", min_val)  
print("Max:", max_val)  
print("Total:", sum_val)
```

### Output:



```
Siddhi_BDAV_Practical Python  
File Edit View Run Help Last edit was now  
Run all My Cluster Share  
Just now (1s) 2 Python  
num_rdd = sc.parallelize([4, 5, 8, 6, 10, 2])  
min_val = num_rdd.reduce(lambda a, b: a if a < b else b)  
max_val = num_rdd.reduce(lambda a, b: a if a > b else b)  
sum_val = num_rdd.reduce(lambda a, b: a + b)  
  
print("Min:", min_val)  
print("Max:", max_val)  
print("Total:", sum_val)  
(3) Spark Jobs  
Min: 2  
Max: 10  
Total: 35
```

### 9. Reduce function on Tuple RDD(String,Int)

#### Code:

```
tuple_rdd = sc.parallelize([("apple", 10), ("banana", 20), ("apple", 30)])  
reduced_rdd = tuple_rdd.reduceByKey(lambda a, b: a + b)  
reduced_rdd.collect()
```

#### Output:



Siddhi\_BDAV\_Practical Python

File Edit View Run Help Last edit was now

Total: 35

Run all My Cluster Share

Just now (2s) 3 Python

```
tuple_rdd = sc.parallelize([("apple", 10), ("banana", 20), ("apple", 30)])  
reduced_rdd = tuple_rdd.reduceByKey(lambda a, b: a + b)  
reduced_rdd.collect()
```

(1) Spark Jobs

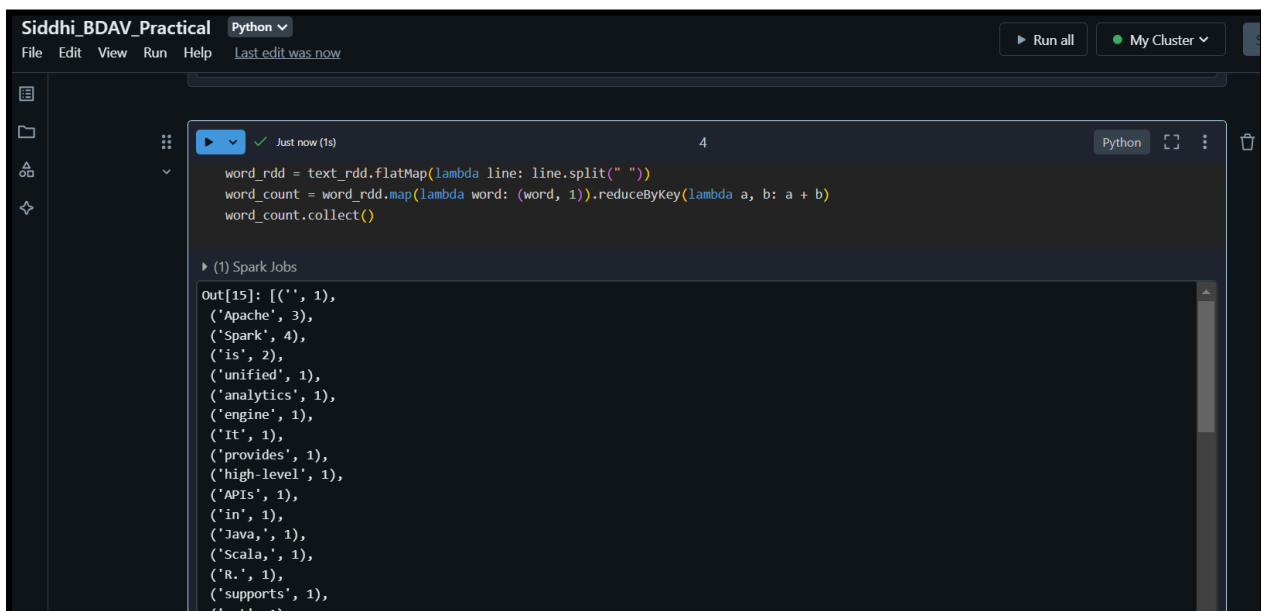
Out[14]: [('apple', 40), ('banana', 20)]

### 10. implement word count in Apache spark using a map and reduce

#### Code:

```
word_rdd = text_rdd.flatMap(lambda line: line.split(" "))  
word_count = word_rdd.map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)  
word_count.collect()
```

#### Output:



Siddhi\_BDAV\_Practical Python

File Edit View Run Help Last edit was now

Run all My Cluster Share

Just now (1s) 4 Python

```
word_rdd = text_rdd.flatMap(lambda line: line.split(" "))  
word_count = word_rdd.map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)  
word_count.collect()
```

(1) Spark Jobs

Out[15]: [(' ', 1), ('Apache', 3), ('Spark', 4), ('is', 2), ('unified', 1), ('analytics', 1), ('engine', 1), ('It', 1), ('provides', 1), ('high-level', 1), ('APIs', 1), ('in', 1), ('Java', 1), ('Scala', 1), ('R.', 1), ('supports', 1), ('...')]