POPU

Topic .....

Date 25/01/2021

Page No. 8

Experiment-3

Q. RDD's Basics and Lambda Expressions
Experiment-3(a)

1. import spark Context

→
```
1 from pyspark.sql import SparkSession
```

→
```
1 spark = SparkSession.builder\
        .master("local")\
        .appName("Experiment3")\
        .config('spark.ui.port','4050')\
        .getOrCreate()
```

2. Load the movies.txt

→ load in Filestore in shared uploads.

3. Create a RDD from movies.txt file using textFile()

→
```
1. movies = sc.textFile("dbfs:/FileStore/shared_uploads/5000687@stu.upes.ac.in/movies.txt")
```

4. Display records.

→
```
1 movies.take(3)
```

5. Split the RDD based on '::' as delimeter

→
```
1 movies_split = movies.map(lambda line: line.split("::"))
```

Teacher's Signature .....

6 Select only movie fields containing movie names - (second index)

→ movies_names = movies_split.map(lambda name: name[1])

→ movies_names.take(2)

7 Find the movies released in 1993 using filter()

→ 1 movies_release = movienames.filter
(lambda name: "1993" in name)

2. movies_release.take(2)

8 Count the number of movies released in 1993.

→ 1 movies_release.count()

9 Test sparksession variable

→ spark.

10 Demonstrate lambda function - length

→ movies_names_length = movies_names.map(lambda x: len(x))

movies_names_length.take(5)

Experiment - 3(b)

1 ~~Loaded~~ Without use of lambda expressions use of functions are as follows:

→ def square(num):
    result = num ** 2
    return result

→ def square(num):
    return num ** 2

→ def square(num): return num ** 2
(Quad style)

2 Introducing lambda expressions
→ lambda num : num ** 2

→ square = lambda num : num ** 2
square(2)

3 Example 1:
→ even = lambda n : n % 2 == 0
even(10)
even(11)

4 Example 2:
→ first_char = lambda s : s[0]
first_char("Lakshay Sharma")

5 Example 3

→ reverse_str = lambda s : s[::-1]

reverse_str ("Laksnay Sharma")

6 Example 4

→ mult = lambda a, b : a * b

mult (2, 3)

Output:

Date - 25/01/2021

```
> movies.take(2)
Out[1] : ["1::Toy story (1995)::Animation|children|comedy",
          "2::Jumanji (1995)::Adventure|children"]


> movies_split = movies.map(lambda line: line.split("::"))
  movies_split.take(2)
Out[2]: [['1','Toy story (1995)',"Animation|children|comedy"],
         ['2','Jumanji (1995)',"Adventure|children"]]

>movies_names = movies_split.map(lambda name:name[1])
  movies_names.take(2)
Out[3]: ['Toy story (1995)','Jumanji (1995)']

> movie_release = moviesnames.filter(lambda
                        name: "1993" in name)
  movie.release.take(5)
Out[4]: ['Boys of St. Vincent, The (1993)',
         'Love & Human Remains (1993)',
         'My Crazy life (Mi Vida Loca) (1993),
         'Beyond Bedlam (1993)',
         'Three Colors: Blue (1993)']

> movie_release.count()
Out[5]: 165
```

Teacher's Signature .....

```
>spark.
Out[6]:
  SparkSession - hive
  SparkContext
  SparkUI
  Version
        v3.0.1
  Master
        local[8]
  AppName
        Databricks Shell

>movies_names_length = movies_names.map(lambda
                          x: len(x))

  movies_names_length.take(10)
Out[7]: [16, 14, 23, 24, 34, 11, 14, 19, 19, 16]
```

Date - 25/01/2021

Output

```
> square(2)
Out[1] 4

> even = lambda n: n%.2 ==0
  even(10)
Out[2] : True
  even(11)
Out[3]: false.

> first_char = lambda s: s[0]
  first_char("Lakshay Sharma")
Out[4]: 'L'

> reverse_str = lambda s: s[::-1]
  reverse_str("Lakshay Sharma")
Out[5]: 'amrahS yahskaL'

> mult = lambda a,b : a*b
  mult(2,3)
Out[6]: 6
```

Teacher's Signature .....