Experiment-6

Q Delving deeper into spark API.

1 Load the rating.txt file into Databricks cloud.

2 Load the movies.txt file into Databricks cloud.

3 create an RDD from rating.txt file using textFile().
→ rating = sc.textFile("/FileStore/tables/rating.txt")

4 Split the RDD based on '::' as a delimeter.
→ rating_split = rating.map(lambda line: line.split("::"))

rating_split.take([illegible]).

5 create a pair RDD with key being movieID and value being its rating.
→ pair_rating = rating_split.map(lambda x: (int(x[0]), int(x[2])))

pair_rating.collect().

6 Group the rating by key and display the output using take().
→ rating_group = pair_rating.groupByKey()
rating_group.take(5).

7 calculate the average rating for all movies using mapValues().

→ rating_average = rating_group. mapValues (lambda x: sum(x)/len(x))

8 Display 10 elements using take()

→ rating_average. take(10).

9 create a Pair RDD with rating being the key and movie ID being its value and count the number of movies for a certain rating.

→ rating_id = rating_split. map(lambda x: (int(x[2]), int(x[0]))). groupByKey()

rating_rdd = rating_id. mapValues (lamuda v: len(v))

10 Review the output.

→ rating_rdd. collect()

11 create a RDD from movies. txt file using textFile().

→ movies = sc. textFile ("Filestore/tables/movie. txt")

movies. take (5).

12 split the RDD based upon "::" as delimeter.

→ movie~~s~~split = movies. map (lambda line: line.split("::"))

movies. split. take (5).

POPU

Date ....

Topic .... Page No. 21

13. create a Pair RDD with movie ID being the key and movie name being its value and display the output using take()

→ movie_pair = movie_split.map(lambda x: (str(x[0]), str(x[1])).

movie_pair. take(5)

14 Now join the movies.txt in step13 and ratings-rdd in steps to find the average rating of movies along with their names.

→ rdd_join = rating_average. join(movie_pair)

15 Display 10 elements and using take()

→ rdd_join. take(5).

16 save the RDD as textFile using save as TextFile()

→ rdd_join. saveAsTextFile ("result. txt")

rdd = sc. textFile ("result. txt")

rdd. take(5).

Teacher's Signature ....

Outputs: Date- 15/07/2021

Out[1]

4
```
[['1', '1193', '5', '978300760'],
 ['1', '661', '3', '978302109']]
```

Out[2]

5
```
[(1, 5),
    (1, 8),
    (1, 3),
    (2, 4),
    - - -
    - - -
    - - -
    (10, 5),
    - - - ]
```

6 Out[3]:
```
[('1', <pyspark.resultiterable.ResultIterable at 0x7267(6203)>
('4', <pyspark.resultiterable.ResultIterable at 0x7f2c765204>),
('8', <pyspark.resultiterable.ResultIterable at 0x7f2c76205>),
('9', <pyspark.resultiterable.ResultIterable at 0x7f2c7206>),
('10', <pyspark.resultiterable.ResultIterable at 0x7f2c7207>)]
```

Teacher's Signature ....

```
9 Out[7]: [('1', 4.18),
           ('4', 4.19),
           ('8', 3.88),
           ('9', 3.71)
           ('10', 4.11),
           ('11', 3.92),
           ('14', 3.13),
           ('16', 3.02),
           ('17', 4.02),
           ('19', 3.52)]
```

```
10 Out[8]: [(4, 227626), (2,72014), (5, 147187), (3,17512),
                                                  (1,137573)]
```

```
11 Out[9]: [['1', 'Toy Story (1995)', 'Adventure|children|comedy'],
    ['2', 'Jumanji (1995)', 'Adventure|children|comedy'],
    ['3', 'Sweetenpain (1995)', 'Comedy|Romance'],
    ['4', 'Waiting to Exhale (1995)', 'comedy|Drama'],
    ['5', 'Father of the Bride (1995)', 'Comedy'].]
```

Teacher's Signature ....

```
13 out[13]: [('1', 'Toy Story (1995)'),
            ('2', 'Jumanji (1995)',
            '('3', 'Greenpton (1995)'
            ('4', 'waiting to Exhale (1995)'
            ('5', 'Father of Bride (1995)')]

15 out[14]: [('4', (4.19, 'waiting to exhale (1995)')),
     ('10', (4.11, 'GoldenEye (1995)')),
     ('12', (3.82, 'Dracula: Dead and loving it (1995)'))
   ('16', (3.02, 'Casino (1995)')),
     ('20', (4.08, 'Money Train (1995)),
     ('24', (3.94, 'Powder (1995)')),
     ('26', (2.93, 'Othello (1995)')),
     ('40', (3.44, 'Cry, the Beyond country (1995)')),
     ('44', (3.63, 'Mortal Kombat (1995)'))
     ('50', (3.06, 'usual suspects, the (1995)'))],
```