Data Science Mock Assessment:

SPARK

My answers have been published here: https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/4034538166551726/181382269688642/5722510562859543/latest.html

The following Scala code will be used for the Q1 - Q4

trainers = (["John Smith", "Kate York", "Jo Jones", "Frank Lee", "a Smith"])

courses = (["SQL", "Pro Skills", "Data Engineering", "Dot Net"])

trainersRdd = sc.parallelize(trainers)

coursesRdd = sc.parallelize(courses)

1. Given the following Pyspark code what is the output

firstLetter = trainersRdd.filter(lambda x : x[0] >= "K")

print(firstLetter.collect())

['Kate York', 'a Smith']

Note lower case letters in ascii are a higher number for upper case

1. Given the following Pyspark code what is the output

trainersLengthRdd = trainersRdd.filter(lambda x: len(x) > 8)

print(trainersLengthRdd.collect())

['John Smith', 'Kate York', 'Frank Lee']

1. Given the following Scala code what is the output

both = trainersRdd.union(coursesRdd)

print(both.collect())

method of the DataFrame is used to combine two DataFrame’s of the same structure/schema

["John Smith", "Kate York", "Jo Jones", "Frank Lee", "a Smith", "SQL", "Pro Skills", "Data Engineering", "Dot Net"]

1. Given the following Scala code what is the output

both = trainersRdd.zip(coursesRdd)

print(both.collect())

[("John Smith", "SQL"), ("Kate York", "Pro Skills"), ("Jo Jones", "Data Engineering"), ("Frank Lee", "Dot Net"), ("a Smith", )])

Use this list of data for Q5 - Q8

([89, 85, 72, 80, 98, 97, 87, 82, 71, 96])

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Write the Pyspark code to turn this list into RDD values?

rdd = sc.parallelize([89, 85, 72, 80, 98, 97, 87, 82, 71, 96])

1. Write Pyspark code to sort these in ascending order (lowest to highest).

rdd.sortBy(lambda x: x)

1. Write Pyspark code to sort in descending order (highest to lowest).

rdd.sortBy(lambda x: -x)

1. Explain what the following code does

numbersRdd. map(lambda x: (x, 1)).reduceByKey(add).sortBy(lambda x: -x[0]) and how can it be changed to show output in descending or ascending order?

Takes the numbersRdd and creates a key (x,1) as a map, then it reduces it by reduceByKey adding all the repeats of x together, then it sorts them in descending order

1. Write Pyspark code to return the 2 highest numbers from the list.

#sort in descending order

rdd.sortBy(lambda x: -x)

# pick the 1st number

nth=1

rddIs = rdd.take(rdd.count())

print(rddIs[nth-1])

1. Write Pyspark code to return the 2 lowest numbers from the list.

#sort in descending order

rdd.sortBy(lambda x: -x)

# pick the 1st number

nth=1

rddIs = rdd.take(rdd.count())

print(rddIs[- nth-1])

Use the following data for Q11 - Q15

distanceToLondonRDD= sc.parallelize([("Leeds", 190),("Leicester", 100),("Glasgow",320)])

distanceToManchesterRDD = sc.parallelize([("Leeds", 40),("Leicester", 130),("Manchester", 0)])

distanceToLeedsRDD= sc.parallelize([("Manchester",40),("Leicester", 100),("Glasgow", 160)])

1. What is the outcome from the Pyspark code

val joinRDD = distanceToLondonRDD.join(distanceToManchesterRDD)

print(joinRDD.collect())

[("Leeds", (190, 40),

"Leicester", (100, 100),

"Glasgow",(320, 160 ),

"Manchester", (None, 40)]

1. What is the outcome from the following Scala code

joinRDD = distanceToLondonRDD.join(distanceToManchesterRDD)

print(joinRDD.collect())

[("Leeds", (190, 40),

"Leicester", (100, 130),

"Glasgow",(320, None ),

"Manchester", (None, 0)]

1. Write the Scala code to produce the following output

[('Leicester', (100, 100)),

('Manchester', (None, 40)),

('Glasgow', (320, 160))]

joinRDD = distanceToLondonRDD.join(distanceToLeedsRDD

1. Explain what following the statement will do?

distanceToLondonRDD= sc.parallelize([("Leeds", 190),("Leicester", 100),("Glasgow",320)])

\_\_will create a list of all the distances to london

1. What will the following statements do?

df = distanceToLeedsRDD.toDF()

newdf = df.withColumnRenamed("\_1", "City").withColumnRenamed("\_2", "Distance From London")

newdf.show()

\_\_\_will create a dataframe from distanceToLeedsRDD, then will rename the first column as city anf the second column as distant from london

Q16. It is claimed that SPARK can be faster than Hadoop for batch processing, by how much?

1. 10 times faster
2. 20 times faster
3. 100 times faster
4. 200 times faster

Q17. In Spark, what command is used to read in a text file?

1. sc.readText()
2. text.readFile()
3. sc.textFile()
4. spark.textFileRead()

Q18. If an RDD is described as being immutable it means that the RDD can be updated in the same area of memory as it was previously

1. TRUE
2. FALSE