ENSF 612 Fall 2021: Quiz 2

Marks: 20

Duration: 48 hours

The quiz is worth 5% of course marks

Instructions:

1. This assignment must be completed individually. You cannot copy from others or consult with others. Any identification of such unauthorized actions may result in 0 grade for this assignment.
2. This is a take home assignment.
3. Each student will submit his/her solution in a PDF file and upload it in D2L under Quiz 1 Folder.

Question 1 (Marks 5).

1. Explain why Apache spark can often execute faster than Hadoop? (Marks 1)
    a. Results from a stage can be kept in memory and can be read from memory by the next stage.
2. True or False? An RDD can be changed after it is constructed. (Marks 1)
    a. False. An RDD is immutable. However, you can create another RDD from it.
3. True or False? Apache action is lazily evaluated. (Marks 1)
    a. No. Only Spark transformations are lazily evaluated
4. Briefly describe the advantage of using broadcast and accumulator variables in Apache spark? (Marks 2)
    a. Look at slides 5 – 11 in Lecture 16 – Spark Programming and ML

Question 2 (Marks 5).

Suppose we have a CSV file of questions from an online forum. We read the CSV file as follows.

df = sc.read.csv(“QuestionsWithAnswers.csv”)
df.show()
QuestionId,HasAcceptedAnswer,Score
1, False, 1
2, True, 30,
....

How can we get the following? (provide pyspark code)

1. The total score of questions with an accepted answer?
2. The difference of score questions with an accepted vs non-accepted answer?

1. Sum

df.where(“HasAcceptedAnswer = True”).select(“Score”).groupBy().sum().collect()[0][0]

2. Difference

df.where(“HasAcceptedAnswer = True”).select(“Score”).groupBy().sum().collect()[0][0] - df.where(“HasAcceptedAnswer = False”).select(“Score”).groupBy().sum().collect()[0][0]

Question 3 (Marks 4).

How can we make the following spark code efficient? (it's not optimized as it is now)

```
fileRDD = sc.textFile("BigLog.txt")
def filter1(record):
        ....
def filter2(record):
        ....
result1RDD = fileRDD.filter(filter1)
print(result1RDD.take(5))
result2RDD = fileRDD.filter(filter2)
print(result2RDD.take(5))
```

fileRDD is used twice in two filters. Since spark transformation is lazily evaluated, this will try to load fileRDD two times in the two filter operation. It's better to call persist() or cache() on the RDD.

Question 4 (Marks 6).

Consider that you have to create a data analytics pipeline for your company to help analyze large volume of consumer product company product purchasing records that are collected from different regions (e.g., North America, Europe, etc.). The pipeline should do the following based on the data from January to December of previous month:

1. Identify patterns of consumer product purchase (e.g., products they busy together).
2. Predict consumer spending for the next month (e.g., for January next year)
3. Connect the purchase of the same consumer across the globe, if the consumer was traveling while purchasing (you can assume that due to privacy the company may only have the consumer first and last time, his/her gender, his/her country of origination, and the last four digits of the credit card the person has used during purchase).

For each of the above use cases, draw and explain a pipeline as follows.

1. The type of algorithm to use (supervised or unsupervised)
2. The kind of data preprocessing that needs to be done
3. The kind of output that you can generate

For 1.

- Types of algorithm: unsupervised algorithms like Frequent Itemset Mining.
- Preprocessing.
    - Some items may be bought all the time by the consumers (e.g., paper towels). It's probably not important to use those into the pattern mining algorithm
    - Patterns for Christmas periods may not be similar to other periods. Thus pattern analysis may not be done on a quarterly basis.

For 2, supervised algorithms (e.g., algorithms that can use time series forecasting)

- Purchasing patterns in Christmas may not match the patterns in January.

- Thus, the preprocessing may need to avoid Christmas periods while predicting for January

For 3, rule-based or supervised algorithms or a combination of both may work.

- For data preprocessing, the consumer records may need to be cleaned (e.g., if there is any typo while recording the names)
-