ICT337
Big Data Computing in the Cloud

Tutor-Marked Assignment

July 2023 Presentation

TUTOR-MARKED ASSIGNMENT (TMA)

This assignment is worth 18 % of the final mark for **Big Data Computing in the Cloud.**

The cut-off date for this assignment is **Thursday, 19 Oct 2023, 2355 hours.**

Note to Students:

You are to include the following particulars in your submission: Course Code, Title of the TMA, SUSS PI No., Your Name, and Submission Date.

Question 1

(a) Discuss in detail on the concept of PySpark Resilient Distributed Datasets (RDD) and PySpark DataFrames.

(10 marks)

(b) Use a table to highlight the main differences between RDD and DataFrames.

(5 marks)

Question 2

(a) Explain in detail on the setup and configuration process of Apache Spark with Spark/Hadoop package in your local machine. Demonstrate the non-interactive execution of built-in "pi.py" example program.

(10 marks)

(b) Explain in detail on the key logic of pi.py program in Figure 1.

```python
from __future__ import print_function

import sys
from random import random
from operator import add

from pyspark.sql import SparkSession

if __name__ == "__main__":
    """
        Usage: pi [partitions]
    """
    spark = SparkSession\
        .builder\
        .appName("PythonPi")\
        .getOrCreate()

    partitions = int(sys.argv[1]) if len(sys.argv) > 1 else 2
    n = 100000 * partitions

    def f(_):
        x = random() * 2 - 1
        y = random() * 2 - 1
        return 1 if x ** 2 + y ** 2 <= 1 else 0

    count = spark.sparkContext.parallelize(range(1, n + 1), partitions).map(f).reduce(add)
    print("Pi is roughly %f" % (4.0 * count / n))

    spark.stop()
```

Figure 1: Snapshot of the pi.py example program

(5 marks)

Question 3

In your local machine's Spark setup, develop a PySpark program using **Spark DataFrame APIs** to perform the following tasks. Show your full PySpark program and provide screenshots for all key steps where applicable.

Data sources used in this question are: (i) flights_data.csv, (ii) planes_data.csv. Note that these data files can be downloaded from ICT337 Canvas webpage.

(a) Perform the following tasks and show the results in each step:
- Read the "flights_data.csv" file and store the content using Spark DataFrame. Show the content, number of occurrences and schema.
- Find any missing data from the DataFrame and drop the corresponding rows. Show the content and the number of occurrences.

(4 marks)

(b) We like to carry out basic data analysis. Perform the following tasks and show the results in each step:
- Find the number of flights per year, month. Sort the results from highest to lowest counts.
- Find the number of flights per day. Sort the results from highest to lowest counts.
- Find the number of flights made by a given carrier. Also, compute the corresponding percentage. Sort the results from highest to lowest counts.
- What are the origin airports and the corresponding number of trips from the airports. Sort the results from highest to lowest counts. Repeat this computation for destination airports as well.
- Find the Top **TEN** (10) planes (i.e., tailnum) that made the most flights. Sort the results from highest to lowest counts.

(7 marks)

(c) Find the number of flight departure per hour. Sort the results from highest to lowest counts. Next, we also analyze departure delay by performing the following tasks:
- Departure delay is characterized by either positive or negative values. Find the average positive departure delay for a given carrier. Sort the results from highest to lowest values. Also, compute the average departure delay.
- Find the average departure delay per month. Sort the results from highest to lowest values.
- Find the average departure delay per hour. Sort the results from highest to lowest values.
- Repeat the above for negative departure delay.

(9 marks)

(d) We now examine travel distance, speed and flight air time. Perform the following tasks and show the results in each step:
- Find the average, minimum and maximum flight distance of a given carrier. Sort the results by average distance from highest to lowest.
- Create a new column, called "flight_speed (miles per hour)" to the existing DataFrame. Note that the unit of flight distance is miles and flight air time is minutes.

- Find the average, minimum and maximum flight speed of a given carrier. Sort the results by average speed from highest to lowest.
- Find the shortest flight from "PDX" in terms of distance. Find the longest flight from "SEA" in terms of flight duration.
- Find the average flight duration of carrier "UA" and originated from "SEA". Also, compute the total flight duration in hours.

(7 marks)

(e) We now combine the existing DataFrame with plane information. Perform the following tasks and show the results in each step:
- Read the "planes_data.csv" file and store the content using Spark DataFrame. Show the content, number of occurrences and schema.
- Delete the "speed" column and change the name of "year" column to "plane_year".
- Perform an inner join with the existing flight DataFrame based on the key of talinum. Find any missing data from the resultant DataFrame and drop the corresponding rows. Show the content and the number of occurrences.
- Find the Top **TWENTY (20)** planes (e.g., carrier, model, plane_year) that made the most number of trips. Sort the results from highest to lowest counts.
- Find the Bottom **TWENTY (20)** planes (e.g., carrier, model, plane_year) that made the least number of trips. Sort the results from the lowest to highest counts.
- Repeat the above Top **TWENTY (20)** and Bottom **TWENTY (20)** computation using **PySpark SQL approach**.

(8 marks)

Question 4

In your local machine's Spark setup, develop a PySpark program using **PySpark RDD APIs** to perform the following tasks. Show your full PySpark program and provide screenshots and results for all key steps where applicable.

Data sources used in this question is: (i) grocery_data.csv. Note that this data file can be downloaded from ICT337 Canvas webpage.

(a) In the "grocery_data.csv" file, each line represents a single transaction (i.e., grocery items in a basket). Perform the following tasks and show the results in each step:
- Read the "grocery_data.csv" file and store the content using Spark RDDs.
- Strip off any trailing spaces and change the grocery item's name to lower case. Show the content and find the total number of transactions.
- Find the most number of items in a basket among all transactions performed. Show the basket content and number of items.
- Find the unique number of items.

(3 marks)

cleansed

(b) Perform the following tasks and show the results in each step:
- Find the Top **TWENTY (20)** most frequently purchased items. Show the item name, number of occurrences, and compute the corresponding percentage.
- Find the Bottom **TWENTY (20)** least frequently purchased items. Show the item name, number of occurrences, and compute the corresponding percentage.

(4 marks)

(c) We like to perform simple Market Basket Analysis (https://en.wikipedia.org/wiki/Association_rule_learning) so as to better understand the customer purchasing pattern (i.e., which products that are likely to be purchased together). Perform the following tasks and show the results in each step:
- Assign an index for each transaction (i.e., start from 0, 1, …, etc.). RDD
- Find all possible **combination** of having 2 grocery items (i.e., item pair X & Y, transaction index). Sort the item pair by name. Show the content and number of records. RDD → Base
- Find a list of transaction indices that are associated with a given item pair. Show the content and number of records.
- Find the total number of times a given item pair X & Y occurs. Sort the counting from highest to lowest. Show the content and number of records.

(6 marks)

(d) Support metric indicates how popular an item pair is, which is measured by the proportion of transactions in which a given item pair appears. Formally, Support is defined as:

combi_2item

$$Support = \frac{Frequency\ (X,Y)}{N}$$

Figure 2: Definition of Support

Note that "*Frequency(X,Y)*" refers to the number of times an item pair X & Y appears, while "*N*" refers to the total number of transactions.

Perform the following tasks and show the results in each step:
- Compute the Support metric for the item pair in Question 4(c). Show the content in terms of ((item X, item Y), (occurrence count, Support percentage)), as well as the total number of records. ! should use combined_2item
- Find the Top **TWENTY (20)** item pair sorted by occurrence count of item pair X and Y, from highest to lowest.
- Find the Bottom **TWENTY (20)** item pair sorted by occurrence count of item pair X and Y, from lowest to highest.

(6 marks)

(e) Confidence metric indicates how likely an item Y is purchased together if item X is purchased. Formally, Confidence(X,Y) is defined as:

$$Confidence = \frac{Frequency\ (X,Y)}{Frequency(X)}$$

Figure 3: Definition of Confidence

Perform the following tasks and show the results in each step:

- We now compute the term *Frequency(X,Y).* Assign an index for each transaction (i.e., start from 0, 1, …, etc.).
- Find all possible sorted **permutation** of having 2 grocery items (i.e., item X and Y pair, transaction index). Then, find a list of transaction indices that are associated with a given item pair. Show the content and number of records.
- Find the total number of times the above item pair X & Y occurs. Sort the counting from highest to lowest. Show the content and number of records.
- Next compute the term *Frequency(X).* Show the total number of times a given item occurs. Sort the counting from highest to lowest.

(10 marks)

(f) Perform the following tasks and show the results in each step:

- Compute the Confidence metric based on the output from Question 4(e). Show the content in terms of ((item X, item Y), (occurrence count of item pair X and Y, occurrence count of item X, Confidence percentage)), as well as the total number of records.
- Find the Top **TWENTY (20)** item pair sorted by occurrence count of item X, and occurrence count of item pair X and Y, from the highest to lowest.
- Find the Bottom **TWENTY (20)** item pair sorted by occurrence count of item X, from the lowest to highest.

(6 marks)

---- END OF ASSIGNMENT ----