**ICT337**

**Big Data Computing in the Cloud**

**July 2024**

**ECA**

|  |  |
| --- | --- |
| Name: | YANG XIAN WEI SHAWN |
| T-Group | T01 |
| Date Submitted |  |

**IMPORTANT NOTE**

**ECA Submission Deadline: Tuesday, 05 November 2024 12:00 pm**

***ECA Submission Guidelines***

*Please follow the submission instructions stated below:*

This ECA carries 70% of the course marks and is a compulsory component. It is to be done

individually and not collaboratively with other students.

**Submission**

You are to submit the ECA assignment in exactly the same manner as your tutor-marked

assignments (TMA), i.e. using Canvas. Submission in any other manner like hardcopy or any

other means will not be accepted.

Electronic transmission is not immediate. It is possible that the network traffic may be

particularly heavy on the cut-off date and connections to the system cannot be guaranteed.

Hence, you are advised to submit your assignment the day before the cut-off date in order to

make sure that the submission is accepted and in good time.

Once you have submitted your ECA assignment, the status is displayed on the computer screen.

You will only receive a successful assignment submission message if you had applied for the

e-mail notification option.

**ECA Marks Deduction Scheme**

Please note the following:

1. Submission Cut-off Time – Unless otherwise advised, the cut-off time for ECA submission will be at 12:00 noon on the day of the deadline.

All submission timings will be based on the time recorded by Canvas.

1. Start Time for Deduction – Students are given a grace period of 12hours.

Hence calculation of late submissions of ECAs will begin at 00:00 hrs the following day (this applies even if it is a holiday or weekend) after the deadline.

1. How the Scheme Works – From 00:00 hrs the following day after the deadline, 10 marks will be deducted for each 24-hour block.

Submissions that are subject to more than 50 marks deduction will be assigned zero mark.

For examples on how the scheme works, please refer to Section 5.2 Para 1.7.3 of the Student Handbook.

Any extra files, missing appendices or corrections received after the cut-off date will also not

be considered in the grading of your ECA assignment.

**Plagiarism and Collusion**

Plagiarism and collusion are forms of cheating and are not acceptable in any form of a student’s work, including this ECA assignment.

You can avoid plagiarism by giving appropriate references when you use some other people’s ideas, words or pictures (including diagrams).

Refer to the American Psychological Association (APA) Manual if you need reminding about

quoting and referencing.

You can avoid collusion by ensuring that your submission is based on your own individual effort.

The electronic submission of your ECA assignment will be screened through a plagiarism

detecting software.

For more information about plagiarism and cheating, you should refer to the Student Handbook.

SUSS takes a tough stance against plagiarism and collusion.

Serious cases will normally result in the student being referred to SUSS’s Student Disciplinary Group.

**(Full marks: 100)**

**Question 1**

**Spark Directed Acyclic Graph (DAG)**, **Resilient Distributed Dataset (RDD)** and **DataFrame**, are three key concepts that play significant role in providing efficient and fault-tolerant distributed data processing.

**Question 1(a)**

Apache Spark supports large-scale parallel data processing.

Discuss in detailed on the notion of Spark job execution process in a cluster environment.

(6 marks)

**ANS:**

**References:**

**Question 1(b)**

Elaborate how the Spark’s DAG works in Spark framework.

(4 marks)

**ANS:**

**References:**

**Question 1(c)**

Both RDD and DataFrame are different data storage strategies used in Spark framework.

Discuss in detailed on the concept of Spark RDDs and DataFrame.

(10 marks)

**ANS:**

**References:**

**Question 2**

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 is**:**

**airbnb\_data.csv**.

Note that this data file can be downloaded from ICT337 Canvas webpage.

**Question 2(a)**

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

* Read the “airbnb\_data.csv” and store the content using Spark DataFrames.

Show the content, schema and DataFrame dimension.

* Find any missing data from the DataFrame.

If so, drop the corresponding rows.

Show the details of before and after the missing value clean up.

* Find the basic statistics associated with each data column.

(6 marks)

**ANS:**

|  |
| --- |
|  |

**Question 2(b)**

Find the **Top Ten (10)** Airbnb’s neighbourhood\_group from the highest to lowest average price.

Show the DataFrame content and visualize the result in a plot accordingly.

Repeat the computation for the neighbourhood category.

Note that for the ease of using Matplotlib/Seaborn plotting, you may use “toPandas()” to convert a Spark DataFrame into Pandas DataFrame.

(6 marks)

**ANS:**

|  |
| --- |
|  |

**Question 2(c)**

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

* Append a new column of “popularity\_index” to the DataFrame.

The popularity index is defined as the percentage of the total number of reviews of a given host over the sum of reviews across all hosts.

Show the Top Ten (10) most popular host (i.e., [host\_id, popularity\_index]).

* Find the Top Ten (10) most popular neighbourhood based on the host’s popularity index.

Show the content and visualize the result in a plot.

(6 marks)

**ANS:**

|  |
| --- |
|  |

**Question 2(d)**

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

* Find the available room type from the input data.
* Find the average price for a given neighbourhood and room type.

Organize your DataFrame in terms of these columns: [neighbourhood, room type 1, …, room type N].

Sort the output by ascending alphabetical order of neighbourhood name.

* Visualize the average price for the above Top Twenty (20) neighbourhood by different room types.

Note that you may use a side-by-side bar chart.

(6 marks)

**ANS:**

|  |
| --- |
|  |

**Question 2(e)**

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

* Find the average price for a given neighbourhood\_group and room type.

Organize your DataFrame in terms of these columns:

[neighbourhood\_group, room\_type, average price], from highest to lowest pricing.

Visualize the result in a plot.

* Find the total number of host listing in a given neighbourhood group, sorted by highest to lowest number of listing.

Visualize the result in a plot.

* Find out who are the Top Ten (10) hosts (i.e., host\_name) that have the highest number of listings.

Show the DataFrame result and also visualize in a plot.

* Find the Top Twenty (20) hosts with their respective advertised room type that have the highest average number of review.

Show the DataFrame and visualize in a plot.

(16 marks)

**ANS:**

|  |
| --- |
|  |

**Question 3**

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 are:

**(i) 5-node-graph.txt, (ii) 20-node-graph.txt, (iii) 40-node-graph.txt.**

Note that these data files can be downloaded from ICT337 Canvas webpage.

Based on PySpark framework, we like to implement Dijkstra’s algorithm (<https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm>) so as to compute shortest path from a given source node to every other destination nodes in a weighted graph.

**Question 3(a)**

Read the 5-node-graph.txt input file and parse the input to RDD structure of:

**(node\_ID,(distance, list of neighbors with associated weight, path)).**

Show the RDD content.

(3 marks)

**ANS:**

|  |
| --- |
|  |

**Question 3(b)**

Based on the input from Question 3(a), design and implement an iterative Dijkstra’s algorithm using infinite while loop.

The final output from the Dijkstra computation should be:

**A list of (node\_ID, (shortest path distance, path traversal)).**

We assume that the source node is node\_ID = 1 and therefore an example of path traversal output is:

**“1→X→ … →Y→D”,**

Where X, Y are intermediate nodes and D is destination node.

(8 marks)

**ANS:**

|  |
| --- |
|  |

**Question 3(c)**

Based on your program in Question 3(b), explain in detailed on how the shortest path computation works by showing the results of each iteration step.

Also, explain what is the condition to break from the infinite while loop.

(8 marks)

**ANS:**

|  |
| --- |
|  |

**Question 3(d)**

Show the number of iterations to complete the shortest path computation for 5-node graph.txt.

Also, show the final output as:

A list of **(node\_ID, (shortest path distance, path traversal)),** sorted by ascending node\_ID.

(4 marks)

**ANS:**

|  |
| --- |
|  |

**Question 3(e)**

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

* Find the **Top Three (3)** furthermost node and its path & distance.

Sort the result by descending distance.

* Find the destination node(s) that have the most number of traversal hops in the path.

Show the detailed output path and distance.

* Find the set of node(s) that are not reachable from source node (node\_ID=1).

Sort the result by ascending node\_ID.

(9 marks)

**ANS:**

|  |
| --- |
|  |

**Question 3(f)**

Repeat the above computation for the scenarios of:

**“20-node-graph.txt” and “40-node-graph.txt”**

And show the respective results as in Question 3(d) and Question 3(e).

(8 marks)

**ANS:**

|  |
| --- |
|  |

**----- END OF ECA PAPER -----**