# DSC 102: Systems for Scalable Analytics

Programming Assignment 1

Released: 23 October 2023, Due: 13 November 2023

## 1 Introduction

In this assignment, you will continue using the Dask library to explore task parallelism on multiple machines. You will be performing feature explorations, data consistency checks, and computing several descriptive statistics about the dataset.

## 2 Dataset Description

You are provided with the Amazon Reviews dataset with the *reviews* and *products* tables as CSV files. The schemas are provided in Table 1.

| (A) Column name | Column description | Example |
|---|---|---|
| reviewerID | ID of the reviewer | A32DT10X9WS4D0 |
| asin | ID of the product | B003VX9DJM |
| reviewerName | name of the reviewer | Slade |
| helpful | helpfulness rating of the review | [0, 0] |
| reviewText | text of the review | this was a gift for my friend who loves touch lamps. |
| overall | rating of the product | 1 |
| summary | summary of the review | broken piece |
| unixReviewTime | unix timestamp of review | 1397174400 |
| reviewTime | time of the review (raw) | 04 11, 2014 |

| (B) Column name | Column description | Example |
|---|---|---|
| asin | ID of the product | 143561 |
| salesRank | sales rank information | {'Movies & TV': 376041} |
| imUrl | url of the product image | http://g-ecx.images-amazon.com / 31mC.jpg |
| categories | list of categories the product | [['Movies & TV', 'Movies']] |
| title | name of the product | Everyday Italian (with Giada de Laurentiis) |
| description | description of the product | 3Pack DVD set - Italian Classics |
| price | price in US dollars | 12.99 |
| related | related products (also bought, also viewed, bought together, buy after viewing) | {'also_viewed': ['B0036FO6SI', '000014357X'],'buy_after_viewing': ['B0036FO6SI', 'B000KL8ODE']} |
| brand | brand name | Big Dreams |

Table 1: (A) *Reviews* table and (B) *Products* table

## 3 Tasks

You will compute several descriptive statistics for both *reviews* and *products* table as follows:

Q1. Get percentage of missing values for all columns in the *reviews* table.

Q2. Get percentage of missing values for all columns in the *products* table.

Q3. Find Pearson correlation coefficient between the price and rating of the products.

Q4. Find mean, standard deviation, median, min, and max for the price column in the *products* table.

Q5. Find number of products for each super-category (the first entry in the "categories" column in the products table). Output categories should be sorted in non-increasing order in the number of products. Categories

with same number of products can appear in any order.

Q6. Check (return 1 or 0) if there are any dangling references from product ids in the *reviews* table to *products* table. Return 1 if there are dangling references and 0 otherwise.

Q7. Check (return 1 or 0) if there are any dangling references from product ids in the "related" column to the "asin" column of the *product* table. Return 1 if there are dangling references and 0 otherwise.

# 4 Extra Credits (+15 points)

For Question 6, if you are able to come up with a solution that efficiently checks the presence of dangling pointer by parallelizing the check across partitions of the reviews dataframe, you will receive an additional 15 points. **Hint:** First try to parallelize the check. You may use methods such as `map_partitions()` to achieve this. Second, you must stop checking when you find a dangling pointer in any of the partitions. For this you may use the `to_delayed()` and `as_completed()` methods. Iterating over the dataset using a for loop is not a parallel solution and will not fetch extra credits.

**Note:** Extra credits will be awarded only if solutions to all questions are correct and fall within provided runtimes.

# 5 Deliverables

1. Follow the instructions in the provided starter ipython notebook and write your code. Instructions to setup the AWS instances and download data for this assignment is in the `pa1_setup.pdf` file.

2. We have shared with you the "development" dataset and expected results on this dataset. Our code's runtime on 1 (runs dask scheduler & jupyter-notebook) + 1 (runs dask workers) = 2 instances and 1 (runs dask scheduler & jupyter-notebook) + 4 (runs dask workers) = 5 instances are roughly 550s and 155s respectively. If you attempt the extra credit and successfully implement a parallelized solution for Q6, the expected runtime on 2 and 5 instances is less than 550s and 195s respectively. You can use this to validate your results and debug your code. The final evaluation will happen on separate held-out test sets. The runtime and the speedup numbers will be different for the held-out test set.

3. Submit your source code as `<TEAM_ID>.ipynb` on Canvas. Make sure that your code is writing results to `my_results_PA1.json`.

# 6 Hints

1. We advise you to start using 5 instances so that you can test your code faster. This aligns with the "fail-fast" philosophy. Once your code is working you can measure the runtime on two and five instances and make further optimizations.

2. What are dangling references?
If there is a value $V$ in column $X$ of table $A$, but this value is missing in column $X$ of table $B$, we say that $V$ is a dangling reference of $X$ from $A$ to $B$.

For example consider the below two tables.

| ID | X |
|---|---|
| 1 | 21 |
| 2 | 32 |
| 3 | 1 |

| ID | Y |
|---|---|
| 1 | 4 |
| 3 | 5 |

Here, ID *2* is a dangling reference from the first table to the second.

3. Worker logs will be output to the terminals where you started your workers and may not appear in jupyter-notebook. You will need to check the terminal output for your workers to debug any errors.