Term: Fall 2024 **Subject:** Computer Science & Engineering (CSE) **Number:** 512

Course Title: Distributed **D**atabase **S**ystems (CSE 512)

Assignment 3

Due on Monday, November 25th at 11:59 pm

This assignment is designed to help you understand **search processing in big data** within the context of a **distributed system**, ElasticSearch. The objective is that you will gain a deeper understanding of the intricacies of search processing in the world of big data using the framework of ***ElasticSearch and Kibana*** and answer the questions by writing the necessary set of **Python** functions.

Please submit your code file and a pdf document with the results.

ASU Academic Integrity:

Students in this class must adhere to ASU's academic integrity policy, which can be found at https://provost.asu.edu/academic-integrity/policy. If you are caught cheating, you will be reported to the Fulton Schools of Engineering Academic Integrity Office (AIO). The AIO maintains a record of all violations and has access to academic integrity violations committed in all other ASU colleges/schools.

Below are the steps you need to follow to set the environment and fulfill this assignment:

1. Read the ElasticSearch and Kibana instructions first by following the link. https://discuss.elastic.co/t/dec-11th-2022-en-running-elasticsearch-and-kibana-v8-locally-macos-linux-and-windows/320174/2

   Make sure ElasticSearch and Kibana have the same version number!
   Download ElasticSearch and unzip by following the link.
   https://www.elastic.co/downloads/elasticsearch

Download Kibana and unzip by following the link.
https://www.elastic.co/downloads/kibana

2. Run ElasticSearch in the ElasticSearch directory.
   Save the generated password for the elastic user and the enrollment token for Kibana in a secure location. These values are shown only **once** when you **start Elasticsearch for the *first time***. Also, note that the **enrollment token for Kibana is only valid for the *next 30 min***!

   bin/elasticsearch Linux/MacOS

   bin\elasticsearch.bat Windows

   Run Kibana in the Kibana directory:

   bin/kibana Linux/MacOS

   bin\kibana.bat Windows

3. Go to the localhost link written in the terminal. **If your enrollment token is not valid, generate a new enrollment token** by running the following command from the Elasticsearch installation directory:

   ```
   bin\elasticsearch-create-enrollment-token.bat --scope kibana
   ```

   After completing Kibana setup, enter username and password to login Kibana.

   username: elastic
   password: password you saved at Elasticsearch section.

4. Install elasticsearch

   ```
   pip install elasticsearch
   ```

5. Download Amazon Reviews dataset (json file)

   https://www.kaggle.com/datasets/abdallahwagih/amazon-reviews

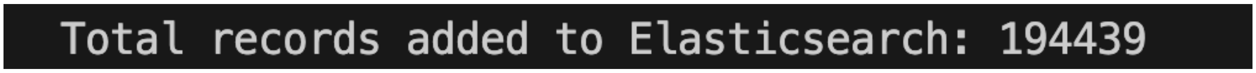
6. Download the assignment3.py file and implement the required Python functions explained below. Run your python file by using the instructions above.
7. Write a report and add results and screenshots from the Python project.

Make sure you run ElasticSearch first to build a connection!

Part 1 (10 points): **upload_product_data()**

Task: Create a Python function to upload the Amazon Product Reviews dataset into Elasticsearch.

- Load and parse the JSON dataset.
- Upload each product review as a document in Elasticsearch.
- Print the total number of records successfully uploaded.



```
Total records added to Elasticsearch: 194439
```

Part 2 (10 points): **top_five_rating_categories()**

Task: Implement a function to identify the top five rating categories in the reviews.

- Count the total reviews for each rating.
- Output: Display the rating categories and their document counts.

Part 3 (10 points): **reviews_with_keyword()**

Task: Implement a function that finds and displays reviews containing a specific keyword (or a list of keywords) within the review text. The function should take the keyword(s) as input and output each review text that contains the specified terms.

- Output: Display the review text for each matching review.

Part 4 (10 points): **reviews_by_reviewer()**

Task: Count how many reviews were written by a specific reviewer.

- Output: Display the total number of reviews by the specified reviewer.

Part 5 (20 points): **reviews_in_date_range()**

Task: Fetch reviews within a certain date range and display them.

- Input: Start and end dates.
- Output: Display review text, rating, and date for each review in the specified range.

Part 6 (15 points): **top_five_reviewers( )**

Task: Implement a function to retrieve the top five reviewers based on the total number of reviews they have written.

- Output: Display the reviewer names along with their total review counts.

Part 7 (15 points): **negative_reviews_with_keywords ( )**

Task: Retrieve reviews that have a rating of 1 or 2 stars and contain specific keywords (e.g., "poor," "bad").

- **Output:** Display review text and rating for each matching review.

Part 8 (10 points): Writing a report.
Write a report to explain:

- What is Elasticsearch and why do we use it?
- What are precision and recall? How to calculate them?
- What are Term Frequency (TF) and Inverse Document Frequency (IDF)?

Please add results and screenshots from the Python project.

---TheEND---