Lab IST769 Unit J - Search

## Agenda

1. Your Questions
2. Go over Problem Set

## 1. Your Questions

Ask any questions you have here!

* Faced barriers while doing 1 and 2 as needed a lot of exploration into the elastic search to get latest data using timestamp.
* Couldn’t get the graphs to work on the last question
* Problems in providing last 5 minutes of tweets data
* Encountered difficulty using curl to efficiently query and retrieve accurate data, as well as experiencing challenges implementing filters effectively within the canvas while creating dashboards.
* Troubles with creating the right metrics and visualizations for both the map and dashboard in Kibana.
* Canvas in the 9th question. It is frustrating, I tried multiple ways but it does not work.
* Elaticsearch SQL was challenging. Getting the graphs right on canvas was challenging too.
* Is Canvas read-only and doesn't support realtime filter?

## 2. Go Over Problem Set J

1. Turn on tweets from the example **J-Elasticsearch** notebook. Search for tweets in the last 5 minutes from the (windows or linux) with **curl** (do the math yourself – don’t make this calculation automatic).

The >= did not work for me, so i did it this way

!curl -X GET 'http://elasticsearch:9200/tweets/\_search?pretty=true' -H 'Content-Type: application/json' -d '{\ "query": {"bool": {"must": [\ {"range": {"date": {"gte": "2024-04-09"}}}\ ,{"range": {"time": {"gte": "00:16:08"}}}\ ,{"range": {"time": {"lte": "00:21:08"}}}\ ]}}}'

2. Do the same thing from the Kibana UI for Elasticsearch. What must you do before you can do that and why? Turn off the tweets.

3. Write PySpark to load the 1,600 line weather data set into Elasticsearch under the index `weather` with default index type.

4. Use a **curl** command from the command line to hit the Elasticsearch API and demonstrate that there are 1,600 documents in the **weather** index.

5. Setup a **weather** index pattern in Kibana based on the **weather** index from Elasticsearch. Make sure you have a **geo\_point** based on lat/lon type and have selected a **@timestamp** field using the date field. Provide a screenshot including the fields in question.

emit(doc['latitude'].value, doc['longitude'].value)

6. Demonstrate your Kibana index pattern is functional. For the most recent entry get the weather for a City of your choice.

7. Create a Kibana map displaying the weather locations for the most recent weather data. Use any layer(s) of data you wish. Provide a screenshot of the map with data points on it.

8. Create a Kibana dashboard which when you select a city, will display the average day time and nighttime temperature for that city, in addition to a line chart of the average daily high and lows for all data on that city. Provide a screenshot of the dashboard in action.

9. Create a Kibana Canvas! Display at least 2 metrics and 2 charts. Decide which data you want to display and how you would like to present it. Provide a screenshot of the Canvas. You can do this for a set of cities or a single city.