Twitter Data Mining and Visualization

Project Description

a) Purpose

a. Learn how to use Docker
b. Work with APIs for web services
c. Become more familiar with Python, Kibana (optional) or any visualizations/graphs you want to create
d. Understand Async vs Sync
e. Understand how to use Elasticsearch Analysis
f. Understand data mining
g. Understand sentimental analysis

b) Motivation

Docker is everywhere, just like open source. Docker applications don't require any specific operating system to work. It is an application where you can download and start playing with it. I will get to explore various docker applications. The docker application will be the base for this project. I will use Twitter API to pull in tweets to verify that it works. Now that there's data, I will need to conduct a sentimental analysis on the tweets, which is the process of computationally identifying and categorizing opinions expressed in a piece of text. I will also get to learn how Elastic Search works by creating visualizations using the data obtain from Kibana.

c) Technical Specification

a. Tools: Docker
b. Platform: Docker
c. Database: Elasticsearch
d. Visualization: Kibana or using D3
e. Programming Languages: Python, JavaScript
f. IDE: VS Code or Sublime
g. Style guides to follow (PEP8 Standard and JavaScript style guide)

d) High-level functionalities

a. Explore Twitter API
b. Display tweets using Twitter API
c. Update the news feed with new tweets
d. Sentimental analysis for each feed (positive, negative, neutral)
e. Interact with Kibana Visualization
f. Text analysis using Elastic Search

e) Score of the project

a. Product Scope: Use Twitter API to pull data from the News Feed onto your website. Ability to display and keep the news feed updated. Data will then undergo a sentimental analysis which will rank the tweets with either (positive, neutral, or negative). Such data will then be analyzed using Elastic Search followed by displaying it using Kibana
b. Deliverables: Will be outlined below
c. What is not part of the project: JavaScript D3.
d. Additional Risks:
    i. Kibana doesn't work (then will have to use JavaScript D3)
    ii. Computer stops working
    iii. Constraints and assumptions: Limitations of Twitter API, and learning new technologies

f) Sketch of GUI

Weekly functional goals

1) Week 1: Docker Set up. Twitter API obtained. Display Tweets onto a website, with tweets being updated. And create two test cases for each screen
2) Week 2: Create a database to store the tweet data or store it locally. Conduct a sentimental analysis for each tweet, categorize each tweet with color identification, and display tweets that went through

3) Week 3: Elasticsearch API obtained. Filter, search, etc through the results. Save the filtered data and display them onto your website. Create visualizations using Kibana, and interact with visualization

Rubrics:

Week 1: April 12

Category	Total Allocated Score	Detailed Rubrics
Docker Set up	3	0: No Docker 3: Docker working
Twitter API Obtained	3	0: No Twitter API 1: Twitter Account Created 3: Twitter API obtained
Display tweets using Twitter API	4	0: No tweets display 2: Tweets display without API 4: Tweets displayed using API
Tweets updated automatically	5	0: No tweets 2: Static Tweets 5: Dynamic tweets display
Test1: Test if tweets are displayed	4	0: Didn't implement test 2: Test implement but failed 4: Test implement with a pass
Test2: Test if API Token is working	6	0: Didn't implement test 2: Test implement but failed -3: API included with submission 6: Test implement with a pass

Week 2: April 19

Category	Total Allocated Score	Detailed Rubrics
Database created	4	0: No Database 2: Incomplete database setup (i.e. missing table or columns) 4: Database working
Sentimental analysis	5	0: No code 1: Code that takes in text file instead of json 3: Partially process of data 5: completed function
Categorizing each tweet with color	3	0: No tweets 3: Completed function
Display categorized tweets	3	0: No tweets 3: Categorized tweets display
Test1: Database is working	5	0: Didn't implement test 2: Test implement but failed 4: Test implement with a pass
Test2: Categorized tweets are displayed	5	0: Didn't implement test 2: Test implement but failed 5: Test implement with a pass

Week 3: April 26

Category	Total Allocated Score	Detailed Rubrics
Elastic Search API	3	0: No API 3: API working
Conduct Search and filtering of data	3	0: No Elastic Search 3: Completed

Save and display filtered data from Elastic Search to Website	3	0: No Data 1: Data Saved 3: Data display on website
Visualizations using Kibana	3	0: No visualization 3: Static Visualization
Interact with visualizations	3	0: No Visualization 1: Static Visualization 3: Completed Function
Test1: Display of data obtained from Elastic Search	4	0: Didn’t implement test 2: Test implement but failed 4: Test implement with a pass
Test2: Visualization is displayed on website	6	0: Didn’t implement test 2: Test implement but failed -1: per screen without a test 6: Test implement with a pass