Assignment #5

Submission:

Submit your assignment as a single ZIP file on Canvas with the name "HW5_YourLastName_FirstName"

Deliverables:

Your assignment submission ZIP file (("HW5_YourLastName")) must have the following items in it:

1. Your **ZIP** file must have two directories: **src** and **doc**
2. Your **src** directory must have 3 subdirectories: **React**, **Flask**, **Forecasting (Tensorflow_LSTM, Prophet, StatsModel)**
3. Your **Readme** files must have the detailed steps to install, deploy, and run every microservice
4. Use **Panopto** to create 15 minutes **video** for a live-demo of your application
5. All source code must be stored under **src** directory
6. Final Report (PDF Document) for the comparative analysis of the experimental results obtained for the 3 models: TF/Keras/LSTM, Prophet, StatModel. It is expected that your recommendation for the best time-series forecasting model shall be based on your comparative analysis of the experimental results.
7. Documentation and video must be stored under **doc** directory

Requirements:

Part I:

Reuse, modify, and fine-tune the tutorials and source code discussed in the class in your implementation of the following requirements:

1) Use **Python/GitHub API** to retrieve information of the past 2 months for the following repositories:
    1. https://github.com/langchain-ai/langchain
    2. https://github.com/langchain-ai/langgraph
    3. https://github.com/microsoft/autogen
    4. https://github.com/openai/openai-cookbook
    5. https://github.com/elastic/elasticsearch
    6. https://github.com/milvus-io/pymilvus/
2) A Line Chart to plot the issues for every Repo
3) A Bar Chart to plot the issues created for every month for every Repo
4) A Bar Chart to plot the stars for every Repo
5) A Bar Chart to plot the forks for every Repo
6) A Bar Chart to plot the issues closed for every week for every Repo
7) A Stack-Bar Chart to plot the created and closed issues for every Repo
8) Use **TensorFlow/Keras LSTM** package to forecast the following for every repo
    1. The day of the week maximum number of issues created
    2. The day of the week maximum number of issues closed
    3. The month of the year that has maximum number of issues closed
    4. Plot the created issues forecast
    5. Plot the closed issues forecast
    6. Plot the pulls forecast
    7. Plot the commits forecast
    8. Plot the branches forecast
    9. Plot the contributors forecast
    10. Plot the releases forecast.
9) Re-implement the above 10 requirements using **Facebook/Prophet:**
10) Re-implement the above 10 requirements using **StatsModel**
11) Deploy to Google Cloud your microservice for the requirements you implemented above as follows:
    i. Use **Docker** and **Google Cloud** to create and deploy the microservices for your application; you need to follow the same process and tutorials demonstrated during class lectures.
    ii. Use **Python** and **Flask** for **Back-End**
    iii. Use **React** and **JavaScript** for **Front-End**

Part II:

Use **Python/GitHub API** to retrieve information of the **past 2 months** for the Repos listed above.

Reuse/Modify/Fine-tune the IPYNB scripts presented in class to create vector embeddings for the issues you retrieved and store them in ElasticSearch (Push, Pull, and Readme directories). You MUST preserve the same directory structure presented in the class.

Create an IPYNB script with the name Analytics.ipynb that used the ElasticSearch index and vector embeddings to answer the following

1) Create a **Bar Chart** to plot the number of issues created for every repo for every day of the week; that is total number of issues created on Monday, Tuesday, Wednesday …, Sunday for EVERY Repo name.

2) Use vector embeddings, score, and semantic search to identify and list the **Top 5 most similar issues** for **every repo** listed above.