Live Dashboard-esque Tool Report

Omer Feyzullahoglu
Binghamton University
Binghamton, New York, USA
ofeyzul1@binghamton.edu

ABSTRACT

In this paper, I describe the road map that I followed while developing web based live dashboard-esque tool for social media data that collected and analyzed in the previous projects.

Keywords

Social media data, dashboard tool, python, Django,

1. INTRODUCTION

In order to work in the field of data analysis, it may not be enough to collect data, analyze and visualize it through various scripts. At the same time, it is necessary to show the results of the analysis dynamically and to develop accordingly. In this direction, this project aims to dynamically display the analysis results of the data collected and analyzed in previous projects and to create a data dashboard.

To achieve this goal, a web page was prepared using python, Django and MongoDB. Django is a Python programming language web application framework. The MVT (Model View Template) design pattern is used. Due to its quick development characteristic, Django is quite demanding. After gathering customer requirements, it takes less time to construct an application.(1)

2. DEVELOPMENT PROCESS

Before I started developing the dashboard, I had to learn a little bit of Django as I hadn't worked with Django before. After doing some research using various sources, especially YouTube, I created the Django project. I had previously determined that the whole process would go through a single page, so I created only one view-creating function and one home html page. Using various html tags, I divided the page into two columns and designed the left side to be static and the right side to be dynamic.

After the design was finished, I started working on the function that created the view. Here is the logic of the function:

- The static part takes the sentiment analysis results previously saved to the database by connecting to MongoDB and puts them in the relevant field.

- Dynamic part takes desired subreddit name with the help of html tag called “form”, then makes search in the MongoDB. After calculating number of, positive, negative, and neutral expressions of the “title” field that belongs to corresponding subreddit, it places on the web page. On the other hand, dynamic part also provides sentiment analysis results for tweets flagged as “possibly sensitive ”, to see results user need to enter “sensitive” to the input area.

3. HOW TO USE DASHBOARD

To use the dashboard, following steps should be followed in order:

- We open VS Code and clone the project from the relevant GitHub repository to our own computer.

- The credentials used to access the database were not uploaded to GitHub for security reasons. Therefore, the username and password I provided must be known in advance.

- After the username and password are written in the relevant field in the connection string which is placed in views.py file, the dashboard tool is ready to work.

- First of all, the server created locally must be run. In order to run the server, it is necessary to be in the relevant folder. If the terminal does not open in the relevant folder, go to the folder named "databoard" from the VS Code terminal and activate the server with the "python manage.py runserver" command.

- After the server starts, the web page is opened by clicking the link which is on the debug console.

- The left column of the table on the web page is static and the right column is dynamic. The name of the subreddit we want to see the results of sentiment analysis is written in the input field on the right and the OK button is clicked.

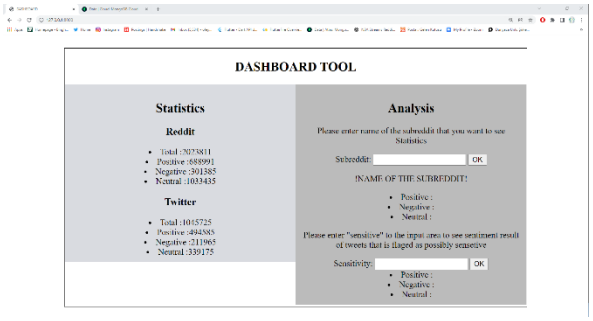

Figure 2:Home Screen

- It takes a little more time than usual to calculate the result because the processor power is not enough. Because of this Reddit results are limited as *120000* posts and Twitter results are limited as *5000* tweets.

4. REFERENCES

[1] Django Project. 2021. Meet Django Retrieved Dec, 16 2021 Retrieved from https://www.djangoproject.com/.