Analysis of the Relationship between Macroeconomic Indicators and the Health/Energy Sectors

Team: Kambhampati Sai Madhav, Cheeti Sricharan, Pendem Shiva Kumar

Section 1

Web App Link: https://dashboard-1.onrender.com/

Credentials:

Username: hello

Password: world

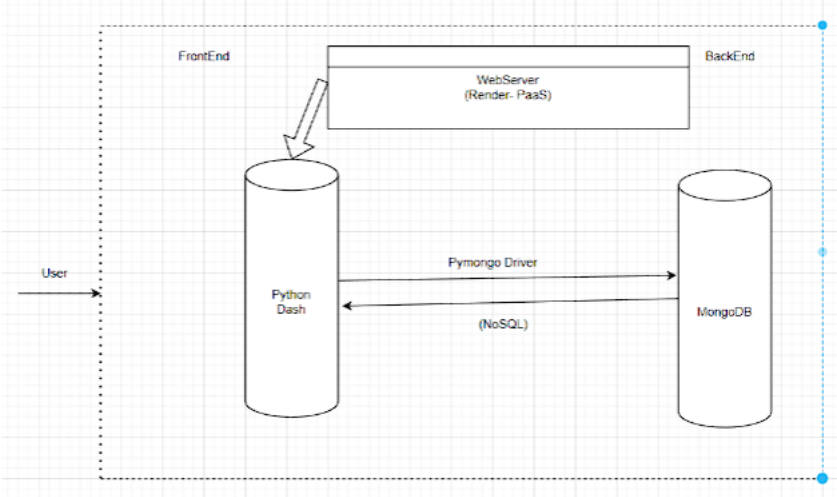
Github Link: https://github.iu.edu/sakamb/Time-Series-Analysis-Project

Section 2

Purpose:

The objective of this project is to analyze the relationship between macroeconomic indicators and the volatility of the health and energy industries. The findings can be useful for investors to diversify their portfolio by focusing on the macroeconomic factors that have historically influenced these industries. We will examine industry-specific ETFs, which track a group of companies belonging to the same industry. Additionally, we will use the various models to compute the volatility response variable. Understanding the impact of macroeconomic factors on these industries can guide investors in making informed decisions about hedging strategies.

How the project is built:



Web Application Architecture

We have successfully completed our project that involves storing data in MongoDB, using Python to build our backend, and designing a visualization dashboard that allows users to interact with multiple dashboards. The data was downloaded from Stooq and uploaded to

MongoDB. We used the pymongo driver to connect to MongoDB, and our APIs were designed to validate user input before retrieving data from the database, thus ensuring data security.

Our visualization dashboard was built using Python Dash, and we made use of the pre-built Bootstrap components available in the library. We hosted our application on Render, a free hosting platform, and used Dash callback functions to provide interactivity with our dashboard.

Overall, our project was successful, and we were able to provide investors with valuable insights into the health and energy industries. Users can interact with our dashboard with confidence, knowing that the data is secure, and the APIs are designed to prevent any unauthorized modifications.

Dataset:

For our project, we used a dataset obtained from **Stooq**, a financial data service provider. The dataset contained macroeconomic indicators such as inflation, unemployment, and GDP, as well as stock price data for companies in the health and energy industries(Vanguard in our case). We selected this dataset because it contained information that was relevant to our project objective, which was to analyze the relationship between macroeconomic indicators and the volatility of the health and energy industries. We also found the dataset to be of high quality, with no missing or erroneous data. We pre-processed the data using Python to ensure that it was in a format suitable for analysis, and then stored it in MongoDB, a NoSQL database.

Functionalities:

- Authentication: Our web app has authentication feature that requires users to log in with their credentials in order to access the Monitoring and Prediction Tabs. This ensures that only authorized users can view and interact with the data and predictions generated by the app.

- Menu Panel: The menu panel will contain two buttons: Monitoring Button and Prediction Button. The Monitoring Button will point to the Monitoring Tab, and the Prediction Button will point to the Prediction Tab.

- Monitoring Tab: The user will be able to view the Macro-Indicator Graph and ETFs graph. The Macro-Indicator Graph will have a dropdown list button that allows the user to choose different indicators such as GDP, Inflation, Employment and also ETF prices. The user can interact with the graphs and visualize the data according to their preferences.

- Prediction Tab: The 'Random Forest Regression Model' was utilized in the backend of our web app to predict the output of the two ETFs, namely VHT and VDE, which the user can select according to their preference.

Section 3

Contributions:

- Madhav, Sricharan, Shiva: Researched macroeconomic indicators to be used: This involves identifying and selecting relevant macroeconomic indicators that are useful in analyzing

and forecasting economic trends. These indicators can include factors such as GDP, inflation, employment rates, and more.

- Madhav, Sricharan: Have done the Transformations section

Transformation Section: Implemented a function which is designed to merge macroeconomic indicator data with daily data based on the specified regime ('PREVIOUS' or 'LAST') for merging. However, it's worth noting that there is no error handling for cases where the specified dates in daily_df do not exist in macro_ind_df, and the function assumes that the index of macro_ind_df represents dates in ascending order. The function assumes that macro_ind_df has a date-based index, and that the macroeconomic indicators are stored as columns in macro_ind_df with the same column names in daily_df for merging. Further error handling and data validation may be necessary depending on the specific use case

- Sai Madhav: After careful consideration and evaluation, I have arrived at a decision to utilize MongoDB and Python for the backend development of our web application. I have assessed the various options available and determined that MongoDB and Python best meet the requirements and objectives of our project. Also collaborated with Sricharan in designing the web application Architecture. Implemented the Implemented the Decision Tree, Random Forest Model.

- Sricharan: After thorough analysis and evaluation of different options, I have concluded that Dash and Plotly are the most suitable choices for building the frontend of our web application. Based on their functionalities and capabilities, I am confident that these tools can effectively meet our project's goals and requirements. Also collaborated with Madhav in designing the web application Architecture. Implemented the Implemented the Linear Regression for the Feature Importance, Implemented the CNN model.

- Shiva Kumar: I am responsible for Decomposition for the VDE and VHT index Models: This task likely involved using time-series analysis techniques to break down the behavior of the VDE and VHT index models over time. Time-series analysis involves analyzing data points collected at regular intervals, such as daily, weekly, or monthly, to identify trends, seasonality, and other patterns. Decomposition of macroeconomic indicators like' GDP', 'CPIAUCSL'(Inflation) , 'UNRATE'(Employment) and more... Have done the correlation analysis for the VDE (Vanguard Energy ETF) and VHT (Vanguard Health ETF) ETF's. Have Implemented the LSTM model. And also, deploying the application to a hosting platform and ensuring its security. These tasks may include setting up the hosting environment on Render. I went on to design the app layout using Baslamiq.

What went well:

The UI is simple and clean.

The selection of MongoDB, Python, Dash, and Plotly for the backend and frontend of the application respectively proved to be the right decision. These technologies were able to meet our project's requirements and goals.

A well-defined project scope and roadmap were created, outlining each team member's responsibilities and the expected timeline for completion

For the Minimal Viable product of Analysis of the Relationship between Macroeconomic Indicators and the Health/Energy Sectors our approach is decent.

Further Improvements:

Our data is stored in MongoDB and we are fetching the data from MongoDB and using it for our web application. Instead of this, we can add an integration to get real time data and generate real time prediction.

And, Another quick way to test different time series models is with the help of a python module pycaret. This library has various types of models and one can easily use this package to compare the results of different models.

The UI can be made user interactive by adding additional features like the sentiment analysis of the ETF's over a particular time period. This way the user would have quick idea whether an ETF price has spiked or decreased.