International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 11 | Nov 2024 www.irjet.net p-ISSN: 2395-0072

ECONOMIC DATA ANALYSIS AND SECTOR WISE FORECASTING USING USER-DRIVEN TOOLKIT

Austin Indrapaul A [1], Gokuldhev M [2]

[1]PG student, Dept. of Computer Science & Engineering, Vel tech University, Chennai, India
[2] Ass. Professor, Dept. of Computer Science & Engineering, Vel tech University, Chennai, India

---

Abstract - *Economic data analysis plays a crucial role in various aspects of decision-making, policy formulation, and understanding the state of the economy. This project introduces a sophisticated no-code solution tailored for comprehensive data analysis and visualization. This innovative initiative is designed to empower users to gain insights on data, without the need of writing a code. By leveraging these we will analyze economic data for a country and create an AI model which uses real time information. Through the integration of some advanced technologies such as React, Flask, and Python, our web-based application delivers an interactive and user-friendly platform for conducting in-depth analysis on this dataset by end users. The bull or bear run of sectors such as real estate, energy, FMCG etc. can be forecasted with the analyzed data using the AI model. The app will feature MVC architecture to improve scalability and easier maintenance. This project can help to understand a country's economic position, movement, pattern and its impact on various sectors based on several factors, and it can predict the future trends.*

Key Words: Data Analysis, Economic Dataset, Forecasting, Sector Analysis, No-Code App

1.INTRODUCTION

Economic data analysis plays a crucial role in understanding and making informed decisions about the economy. The objective is to remove barriers for individuals who may not have programming or data science skills but still want to perform analysis on economic datasets. The solution will offer a user-friendly interface that allows users to select the desired dataset, choose the analysis they want to perform, and generate insights and visualizations without writing any code. The project operates within the domain of Data analysis and Machine learning through a web application. To ensure accurate and reliable analysis, the project will prioritize data processing and transformation operations. Visualization will enable users to explore and interact with the charts to gain a deeper understanding of the economic data. It has wide-ranging applications in policy making, business analysis, financial services, academic research, non-profit organizations, economic forecasting, and risk assessment. With the help of advanced technologies, this toolkit will make economic data analysis more accessible and user-friendly.

1.1 EXISTING SYSTEM

The existing system for economic data analysis typically involves the use of statistical software like R or Python. In this system, professionals with expertise in economics, finance, or data analysis are required to write code to manipulate and analyze economic data. They use programming languages to perform tasks such as data cleaning, transformation, statistical modeling, and visualization. This approach requires a strong understanding of programming concepts and statistical techniques, making it more suitable for individuals with technical backgrounds. The existing system often involves a steep learning curve for non-technical users who may not have the necessary programming skills or domain knowledge.

Disadvantage: Models trained with historic data can be less effective and adding diverse datasets to the pretrained model can be challenging. Also, the existing software can be complex for the individuals without a technical background.

1.2 PROPOSED SYSTEM

This project aims to simplify the process and make it more accessible to non-technical users. It would involve the development of a user-friendly interface or platform that allows users to perform economic data analysis without writing code. The proposed system would provide intuitive tools and functionalities that enable users to manipulate, analyze, and visualize economic data using a visual interface. This could include features such as drag-and-drop functionality, pre-built analysis templates, interactive visualizations, and automated data cleaning and transformation processes.

Advantages: The goal of the proposed system is to empower users with limited technical skills to conduct economic data analysis efficiently and effectively, without the need for extensive programming knowledge. The analysis performed will be completely user-driven, based on the user's needs the model can be tuned and it provides a personalized feel for the user.

2. METHODOLOGY

For this application, MVC architecture is followed, the MVC (Model-View-Controller) architecture is a design pattern

that separates an application into three interconnected components. The Model represents the application's data structure and business logic. The View is responsible for rendering the Model's data to the user, and the Controller handles user input and updates the Model and View accordingly. This architecture promotes organized coding practices and allows for efficient code management and scalability.

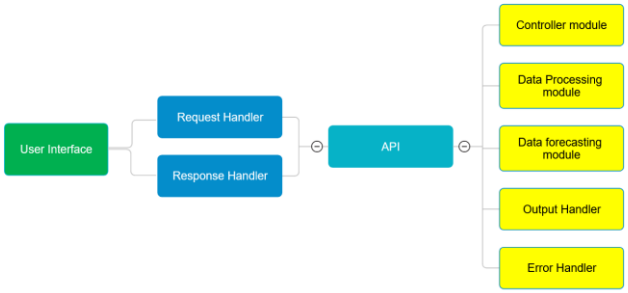

Fig-1: Architecture Diagram

In above Fig-1, The architecture diagram depicts a modular software system, comprising a User Interface (UI), an Application Programming Interface (API), and various backend modules. The UI, the component with which end-users interact, is connected to two handlers: the Request Handler and the Response Handler. These handlers are likely responsible for processing incoming requests from the UI and sending back responses after processing, respectively.

The API serves as a gateway between the UI handlers and the backend modules, facilitating communication and data exchange. On the right side of the API, there are five distinct backend modules, each with a specific role within the system:

Controller module: This could be the central unit that orchestrates the flow of data and commands between the other modules.

Data Processing module: This module is likely responsible for manipulating and preparing data for further analysis or presentation.

Data forecasting module: This suggests the system has predictive capabilities, possibly using historical data to make future projections.

Output Handler: This module may be in charge of formatting and delivering the processed data or results back to the UI through the API.

Error Handler: This module is crucial for robustness, as it manages errors that may occur during the processing of requests. It ensures that the system can gracefully handle exceptions and provide meaningful feedback to the user.

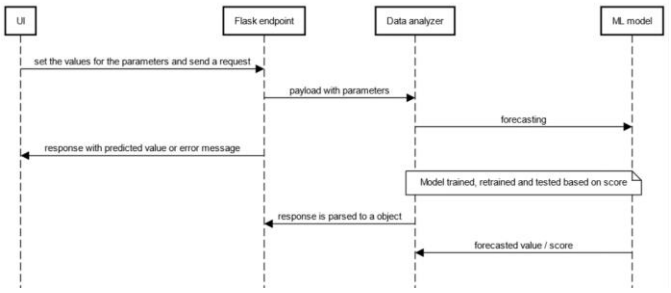

Fig -2: Sequence Diagram

In above Fig-2, we have three main participants: the User, the System, and the Forecasting Model. The application begins when the user gives the values for the input parameters. The data will be sent to backend system via flask API endpoint. Following this, all the validations are performed on the input. Then the parsed input is sent to the model for the prediction. The model can be retrained for improving the accuracy. Finally, the output in passed back to the UI.

In summary, these diagrams represent a structured approach to handling user requests, processing data, and managing system responses and errors, which is typical in modern software architecture. Overall, the diagram presents a high-level view of a system designed to interact with users, process complex data, and return results in an efficient and controlled manner. The separation of concerns, as shown by the distinct modules, suggests a modular design that could facilitate maintenance and scalability.

3. IMPLEMENTATION

Below are the Implementation steps:

1. Collect and clean economic data: Gather relevant economic data from reliable sources such as government databases, financial institutions, and research organizations. The collected data should then be carefully cleaned and organized to ensure its accuracy and consistency. This process may involve removing duplicates, handling missing values, and standardizing the data format.

2. Process and analyze the data: Import the collected economic data into the workspace and preprocess it. Leverage the existing libraries and AI capabilities to perform data analysis tasks such as exploratory data analysis, data transformation, correlation analysis, and regression analysis.

3. Generate the model: After processing and analyzing the data, the next step is to generate the model. This involves training the AI models on historical data to learn the patterns and relationships specific to each sector. The input parameters we considered here are date, population, gold price, oil price, S&P index, GNI, and Inflation, then the output parameters are real-estate index and IT index. ML algorithms can be used to develop predictive models that

can forecast future economic conditions. The models can be trained using techniques such as supervised learning, unsupervised learning, or time series analysis, depending on the nature of the data and the forecasting requirements.

4. Develop a MVC web application: To provide a user-friendly interface for interacting with the economic data analysis and forecasting models, a Model-View-Controller (MVC) web application can be developed. The MVC architectural pattern allows for the separation of concerns, with the model representing the data and its related logic, the view handling the presentation of data to the user, and the controller acting as an interface between the model and the view components. This separation allows for easier maintenance, scalability, and reusability of the application.

5. Integration: The final step is to integrate the developed models and the MVC web application. The models can be integrated into the application to provide real-time or on-demand forecasting capabilities. The application can be designed to allow users to input specific parameters or criteria for generating sector-wise forecasts. The integration should ensure seamless communication between the user interface, the data analysis models, and the forecasting algorithms, providing users with accurate and up-to-date insights.

4. CODE AND OUTPUT

For the complete source code, documentation, and further updates on this project, please visit the GitHub repository. This repository contains all the necessary files, including the application code, installation instructions, and user guides to help you get started with the Economic Data Analysis using Flask App.

https://github.com/austin-indrapaul/ResearchProjects

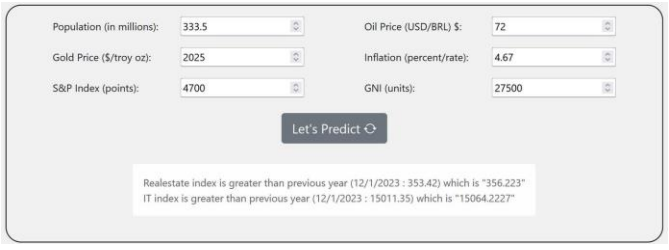

Fig-3: Output or the predicted value from the model in UI

The Fig-3, involves the initialization of the input parameters system, all the inputs are obtained from the UI and sent to the backend system for the analysis. These inputs will be sent in a JSON format and the validation steps are performed. After successful validation the input are fed to the model for the prediction. The predicted values are correlated with the last set of records and it generates a meaningful result.

5. FUTURE ENHANCEMENT

• Multi-modal Data Integration:

Incorporate additional modalities of data, such as employment rate, crime rate and consumer price index (CPI) which creates a comprehensive and more accurate model. This could enhance the accuracy and reliability of predicted values.

• Real-time information and Intervention:

Develop capabilities for real-time feedback and data intervention, where the system can provide a facility to regenerate the model after add the new data, henceforth providing the model with latest data which improves its reliability.

• Personalized Recommendations:

Instead of using a fixed dataset, we can make an option to use the completely different country's dataset, which can be more helpful to end user. Also, all the analysis performed in a session can be sent via mail or saved for future use.

• User Interface and Accessibility:

Improve the user interface of the system to make it more intuitive and user- friendly. Consider accessibility features to ensure that individuals with diverse needs can easily interact with and benefit from the system.

• Continuous Model Training:

Implement a mechanism for continuous model training to keep the system updated with evolving data using a table editor from the UI. This could involve periodically retraining the model with new datasets to improve its adaptability over time.

6. CONCLUSIONS

By understanding economic statistics and using statistical methods, economic data analysis helps in interpreting trends, patterns, and relationships. Overall, economic data analysis is essential for informed decision-making, resource allocation, and understanding the economy. A user-friendly UI based application for economic data analysis can contribute to improved user experience, therefore it makes data analysis more efficient and effective.

REFERENCES

[1] H. Williams and P. Davis, "Data-Driven Decision Making in Economics: Techniques and Tools," Review of Economic Studies, vol. 90, no. 4, October 2024, pp. 789-803, doi:10.1093

[2] J. Smith and A. Johnson, "Economic Data Analysis: Techniques and Applications," Journal of Economic Perspectives, vol. 35, no. 4, Fall 2021, pp. 45-67, doi:10.1257

[3] K. Thompson and J. Green, "Exploring the Impact of Economic Indicators through Data Analytics," Journal of Economic Research, vol. 15, no. 1, February 2024, pp. 33-51, doi:10.2307

[4] M. Lee and T. Zhao, "The Role of Economic Data Analysis in Policy Making: Lessons from Recent Trends," Economic Policy Analysis, vol. 8, no. 3, September 2024, pp. 205-220, doi:10.1080

[5] R. Patel and S. Thompson, "Flask for Economic Data Visualization: Techniques and Case Studies," International Journal of Economic Analysis, vol. 15, no. 2, April 2024, pp. 120-135, doi:10.1080