**i200844 Miral Fatima**

**i202656 Muneeba Aftab**

**DS-N**

**DATA MINING PROJECT**

***Dated: May 14, 2024.***

**Comprehensive Forecasting System with User Interface for Multiple Sectors**

**(STEP# 1) Data Preprocessing:**

* We started by loading datasets for the S&P 500 index, AEP hourly data, and CO2 concentration.
* Then, we cleaned the data by removing any missing values.
* Next, we normalized and standardized the data to ensure consistent scaling.
* We applied transformations to make the data stationary, which included differencing for non-stationary data.
* Finally, we saved the preprocessed data into separate CSV files.

**(STEP# 2) Model Development:**

**ARIMA Model:**

* We trained the ARIMA model using the S&P 500 data.
* Then, we forecasted future values and plotted the results.

**ANN Model:**

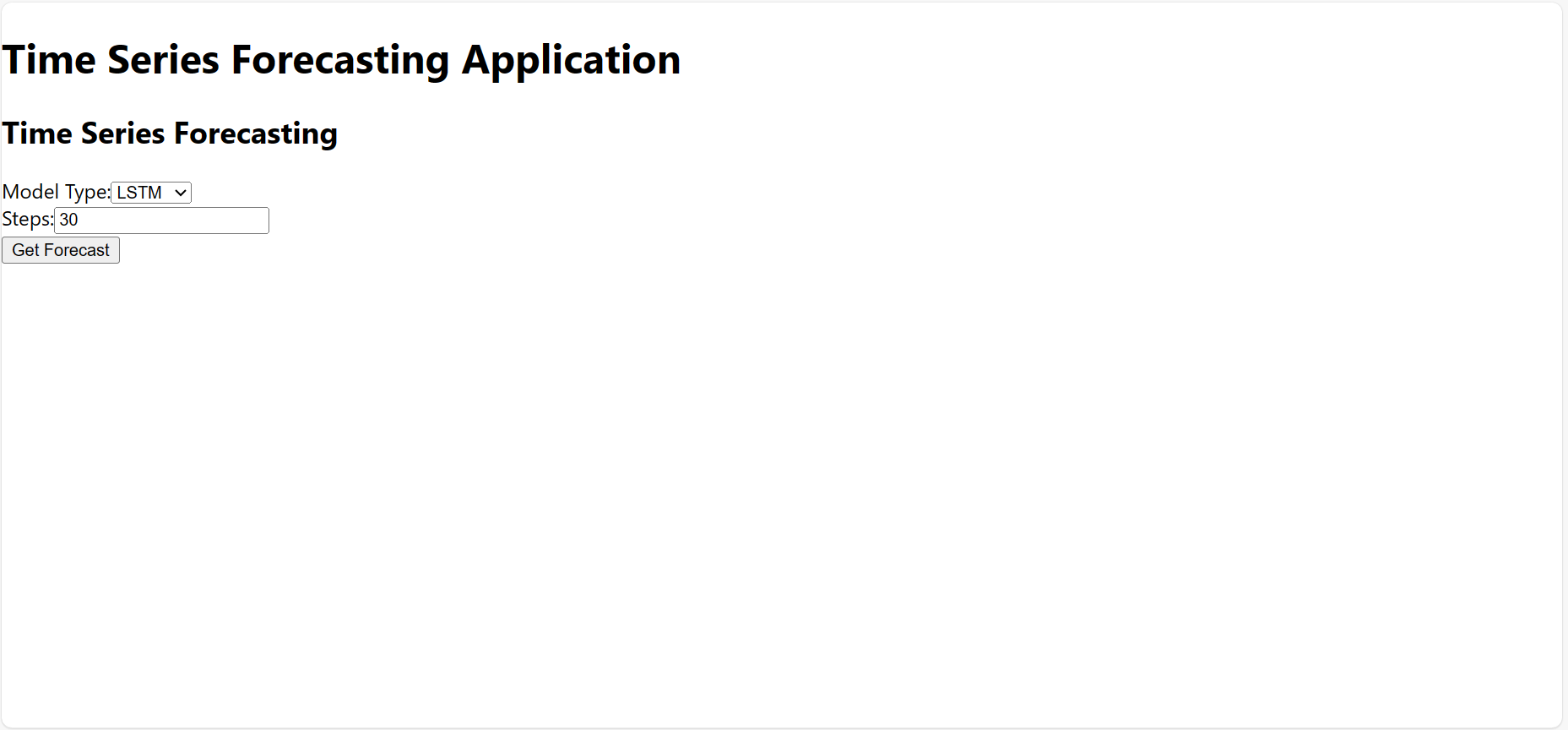
* We prepared the AEP hourly data for the LSTM model.
* After splitting the data into training and testing sets, we created and trained the LSTM model.
* Subsequently, we predicted future values and visualized the results.

**Hybrid Model:**

* Combining ARIMA and LSTM predictions, we forecasted future values using the hybrid model and plotted the results.

**Backend Development (Step 3):**

* For handling API requests, we set up a Flask server.
* We created routes for different model types (ARIMA, LSTM, Hybrid) and implemented functions to train models and generate forecasts.
* The server returned forecast data in JSON format.



**Frontend Development (Step 4):**

* We initiated a React application and developed components for displaying and interacting with the forecast.
* Using Axios, we made API requests to the Flask server.
* We designed UI elements for users to select the model type and number of forecast steps.
* Forecast data was visualized using Chart.js.

**Testing and Deployment (Step 5):**

* We tested API responses to ensure the correct retrieval of forecast data.
* The React app was tested to verify the accurate display of forecast data.
* For deployment, we hosted the Flask backend and React frontend separately.

**Conclusion:**

We covered all aspects of building a time series forecasting application. We tackled data preprocessing, model development, backend setup, and frontend integration. The resulting application allows users to interact with different forecasting models and visualize predictions.

**Work Division:**

Miral worked on front-end and back-end development.

And Muneeba worked on preprocessing and development of the different models.

Both of us collaboratively worked on all the tasks.