**Project Report: IMPLEMENTING ATATIATICAL FORECASTING IN NPCYF**

**Problem Statement**

The task at hand is to develop a time series forecasting application using the ARIMA (AutoRegressive Integrated Moving Average) model. Time series forecasting is crucial in various fields such as finance, economics, and environmental science, where predicting future values based on historical data is essential for decision-making. The application allows users to upload their own time series data in CSV format, check for stationarity, and fit an ARIMA model to forecast future values, providing a user-friendly interface for non-technical users.

**Work Approach**

1. **Research and Understanding**:
   * Studied the ARIMA model and its components (p, d, q) to understand how it works for time series forecasting.
   * Reviewed the requirements for checking stationarity and the differencing process.
2. **Environment Setup**:
   * Set up a Python environment using virtual environments to manage dependencies.
   * Installed necessary libraries: Pandas, NumPy, Streamlit, Statsmodels, and Matplotlib.
3. **Application Development**:
   * Developed functions to check for stationarity using the Augmented Dickey-Fuller test.
   * Implemented a function to apply differencing to the data to achieve stationarity.
   * Created a function to fit the ARIMA model based on user-defined parameters.
   * Built the Streamlit application interface for user interactions, including file upload, column selection, and parameter input.
4. **Testing and Validation**:
   * Tested the application with various datasets to ensure functionality and accuracy of the forecasting.
   * Validated the model's performance by comparing forecasted values with actual data.
5. **Documentation**:
   * Created user and technical manuals to guide users and developers in using and understanding the application.

**Work Products and Deliverables**

* **Software Programs**:
  + A Streamlit application for time series forecasting using ARIMA.
* **Datasets**:
  + Sample CSV datasets for testing the application (not included in the repository).
* **Documentation**:
  + User Manual
  + Technical Manual
* **GitHub Repository**:
  + GitHub Repository Link ([Project link](https://github.com/users/SUBHASISHALDER/projects/3?pane=issue&itemId=85720829))

**User Manual**

**How to Run the Application**

1. **Install Required Libraries**: Ensure you have Python installed. Create a virtual environment and install the required libraries using the following command:



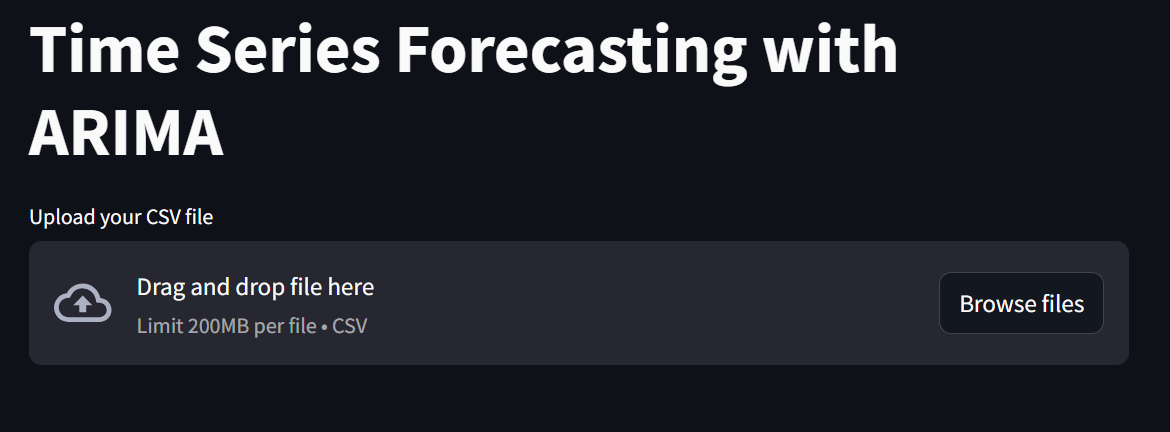
1. **Run the Application**: Navigate to the directory containing the application code and run:

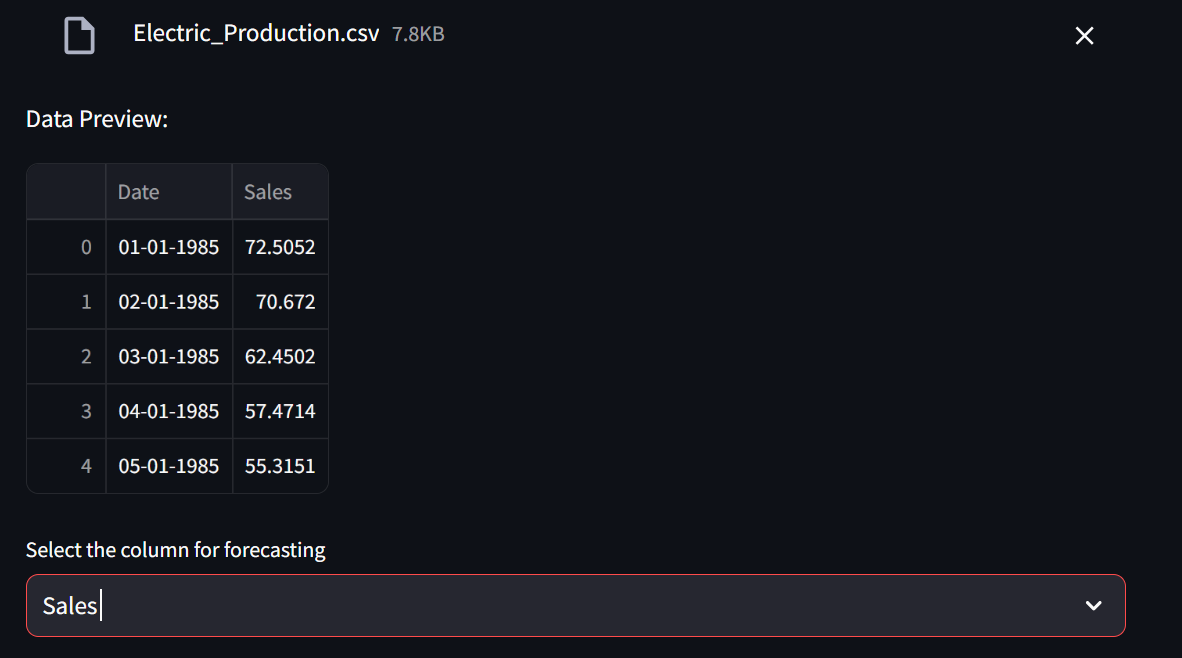


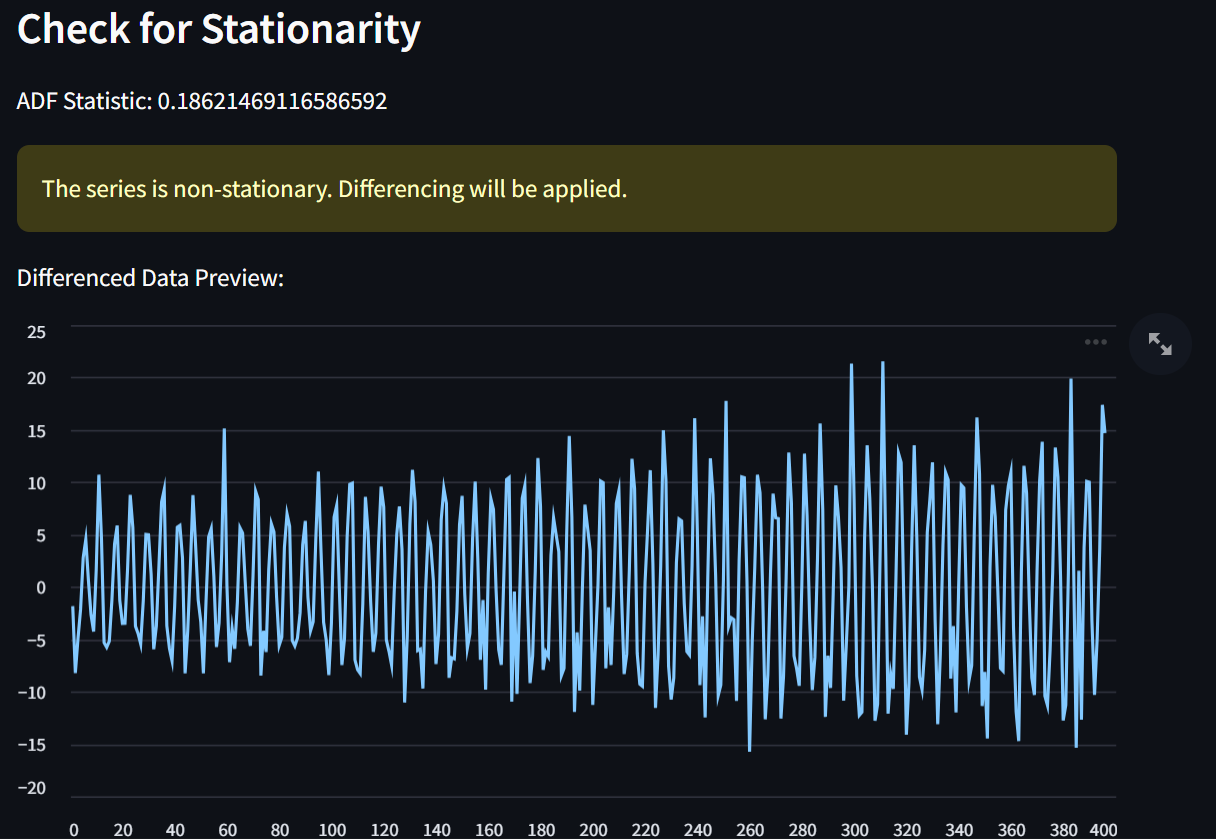
1. **Upload Your Data**:
   * Open your web browser and go to **http://localhost:8501**.
   * Click on the "Upload your CSV file" button to upload your time series data.
2. **Select Column for Forecasting**:
   * Choose the column you want to forecast from the dropdown menu.
3. **Check for Stationarity**:
   * The application will automatically check if the series is stationary. If not, it will apply differencing.
4. **Fit the ARIMA Model**:
   * Input the ARIMA parameters (p, d, q) and click the "Fit Model" button.
5. **Forecasting**:
   * Specify the number of steps to forecast and view the forecasted values.
6. **Visualize Results**:
   * The application will display a plot comparing historical data with forecasted values.

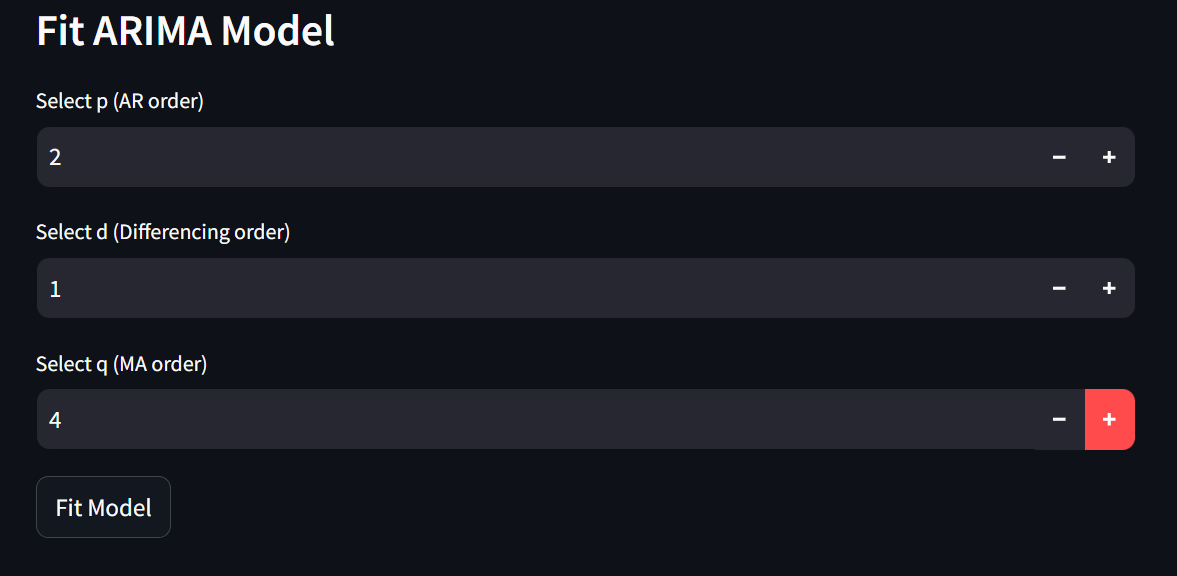
**Screenshots**

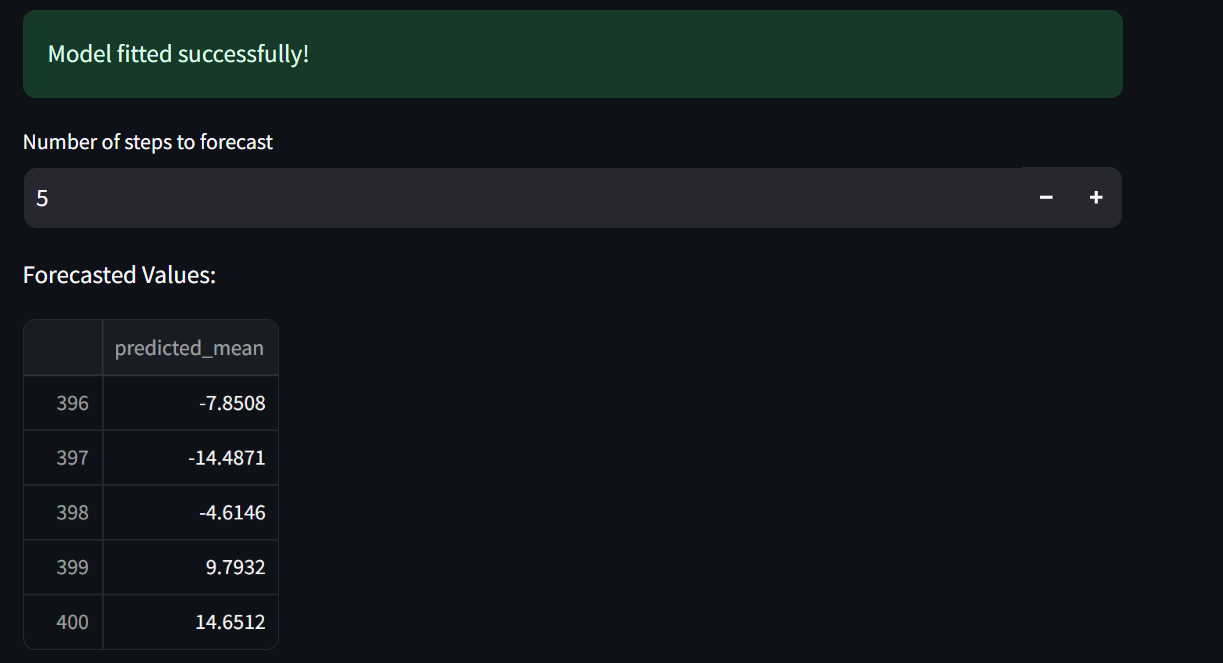
*(Include screenshots of the application interface at various stages, such as uploading a file, selecting columns, and displaying results.)*

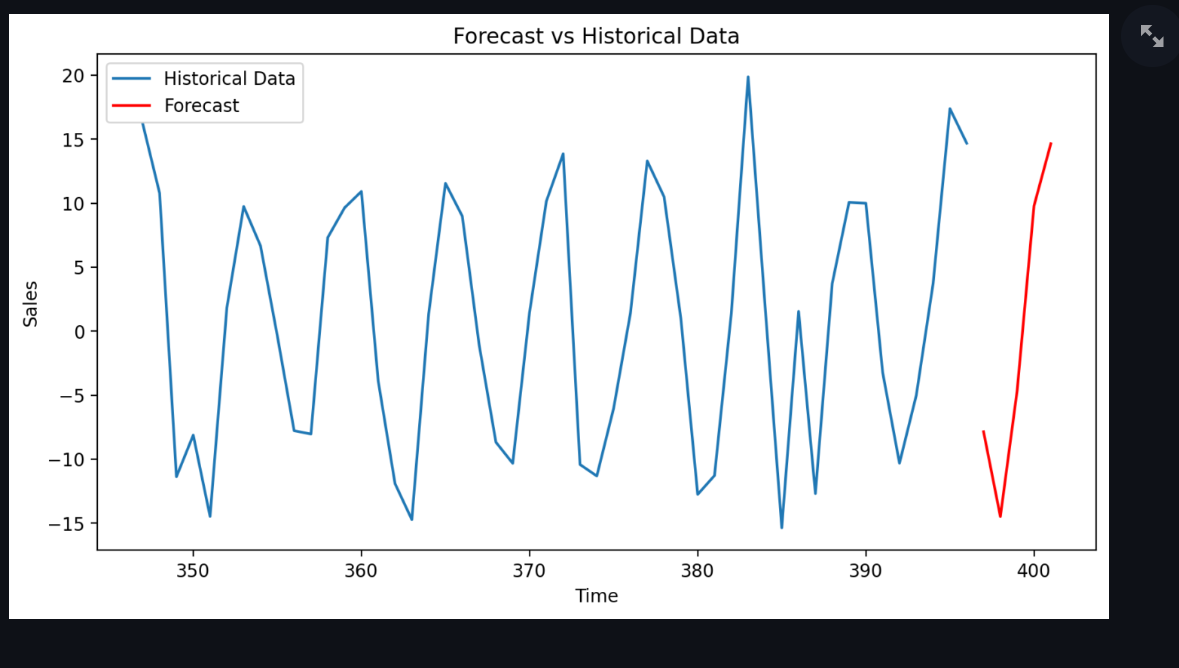
****

****

****

****

****

****

**Technical Manual**

**Datasets**

* The application requires a CSV file with a date column and at least one numerical column for forecasting. The date column should be named "Date".

**Software Libraries Used**

* **Pandas**: For data manipulation and analysis.
* **NumPy**: For numerical operations.
* **Streamlit**: For creating the web application interface.
* **Statsmodels**: For statistical modeling, including ARIMA.
* **Matplotlib**: For plotting graphs and visualizations.

**Program Documentation**

**Functions**

1. **check\_stationarity(data)**:
   * **Purpose**: Checks if the time series data is stationary using the Augmented Dickey-Fuller test.
   * **Returns**: p-value of the test.
2. **difference\_data(data)**:
   * **Purpose**: Applies differencing to the data to help achieve stationarity.
   * **Returns**: Differenced data.
3. **fit\_arima\_model(data, order)**:
   * **Purpose**: Fits the ARIMA model to the provided data using the specified order (p, d, q).
   * **Returns**: Fitted ARIMA model.
4. **main()**:
   * **Purpose**: Contains the core logic of the Streamlit application, handling