Comprehensive Guide to Running the Streamlit Demand Forecasting Application

This document provides a detailed, step-by-step guide to setting up, running, and interacting with the Streamlit Demand Forecasting Application. This application is designed to help users analyze historical demand data, generate forecasts, and optimize supply chain parameters such as lead time and buffer stock.

1. Introduction

The Streamlit Demand Forecasting Application is a powerful tool for businesses to predict future demand for their products and manage their supply chain effectively. It incorporates features for data upload, interactive forecasting, supply chain parameter adjustments, and detailed visualization of results. The application is built using Python and the Streamlit framework, making it highly interactive and user-friendly.

2. Prerequisites

Before you can run the Streamlit application, ensure you have the following software installed on your system:

- **Python 3.7 or higher**: The application is developed in Python. You can download Python from the official website: https://www.python.org/downloads/
- **pip**: Python's package installer. It usually comes bundled with Python installations. You can verify its installation by running `pip --version` in your terminal.

3. Recommended File Structure

To keep your project organized, it is recommended to place the Streamlit application file (`demand_forecasting_app.py`) in a dedicated project directory. You can also create a `data` subdirectory to store your CSV input files.

```
/your_project_directory/
├── demand_forecasting_app.py
```

```
    └──── /data/
        └──── your_demand_data.csv
```

4. Installation of Dependencies

The Streamlit Demand Forecasting Application relies on several Python libraries. You need to install these libraries using `pip`. It is highly recommended to use a virtual environment to manage your project dependencies, which helps in avoiding conflicts with other Python projects.

4.1. Create a Virtual Environment (Recommended)

Open your terminal or command prompt, navigate to your project directory, and run the following commands:

```
python3 -m venv venv
source venv/bin/activate  # On Windows, use `venv\Scripts\activate`
```

This will create a virtual environment named `venv` and activate it. You will see `(venv)` prefix in your terminal prompt, indicating that the virtual environment is active.

4.2. Install Required Python Libraries

With your virtual environment activated, install the necessary libraries using the following command:

```
pip install streamlit pandas numpy plotly scikit-learn
```

These libraries include:

- **streamlit**: For building the web application interface.
- **pandas**: For data manipulation and analysis.
- **numpy**: For numerical operations.
- **plotly**: For creating interactive charts and visualizations.
- **scikit-learn (sklearn)**: For machine learning models used in forecasting.

5. Running the Application

Once all dependencies are installed, you can launch the Streamlit application. Make sure your virtual environment is activated and you are in the directory containing `demand_forecasting_app.py`.

5.1. Launch Command

Execute the following command in your terminal:

```
streamlit run demand_forecasting_app.py
```

5.2. Expected Output

Upon successful execution, Streamlit will launch a local server and provide you with URLs to access the application in your web browser. You will typically see output similar to this:

```
 You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.X:8501
```

Open your web browser and navigate to the `Local URL` (e.g., `http://localhost:8501`). This will open the Streamlit Demand Forecasting Application interface.

6. Interacting with the Web Interface

The application's interface is designed for intuitive interaction, allowing you to upload data, configure forecasts, and visualize results.

6.1. File Upload Functionality

On the left sidebar, you will find a "Data Upload" section with a "Choose a CSV file" button. Click this button to upload your demand data in CSV format. The application expects your CSV file to contain at least the following columns:

- `WEEK`: Date of the week (e.g., `YYYY-MM-DD` format).
- `UNITS`: Number of units sold.
- `SALES`: Sales revenue.

Optionally, your CSV can also include:

- `STORE`: Identifier for the store.
- `PRODUCT`: Identifier for the product.
- `BASE_PRICE`: Base price of the product.
- `PRICE`: Actual selling price.
- `FEATURE`: Indicator for promotional features.
- `DISPLAY`: Indicator for product display.
- `INVENTORY`: Inventory levels.
- `VISITS`: Number of customer visits.

If required columns are missing, the application will display an error message. A sample data format is displayed on the main page when no file is uploaded.

6.2. Sidebar Controls

The sidebar is your primary control panel for configuring the forecast and supply chain parameters:

- **Forecast Configuration**:
    - **Select Target Variable**: Choose between `UNITS` or `SALES` as the variable to forecast.
    - **Forecast Periods (weeks)**: Adjust the number of weeks into the future you want to forecast.
- **Data Filters**:
    - **Select Store**: If your data includes a `STORE` column, you can filter the data to analyze specific stores.
    - **Select Product**: If your data includes a `PRODUCT` column, you can filter the data to analyze specific products.
- **Supply Chain Parameters**:
    - **Lead Time (weeks)**: This slider allows you to specify the lead time in weeks, which is the duration between placing an order and receiving delivery. The application will adjust the forecast timeline based on this parameter.
    - **Buffer Stock (%)**: This slider enables you to set a percentage-based buffer stock. This acts as a safety stock, adding a percentage of the forecasted demand to the total requirement.
    - **Buffer Stock (Quantity)**: This input field allows you to add a fixed quantity as a buffer stock, independent of the forecasted demand.

As you adjust these parameters, the application will dynamically update the forecasts and visualizations in real-time.

6.3. Interactive Charts

The main section of the application displays interactive charts, primarily focusing on forecast visualization:

- **Original vs Adjusted Forecasts**: This chart compares the initial demand forecast with the forecast adjusted by your specified lead time and buffer stock parameters. It also includes historical actual data for context.
- **Impact Analysis**: This section provides key metrics and a bar chart illustrating the impact of your supply chain adjustments on the total forecasted demand. You will see the total original forecast, total adjusted forecast, and the quantitative and percentage impact of the adjustments.

These charts are built using Plotly, allowing for interactive features like zooming, panning, and hovering over data points to view detailed information.

6.4. Data Table Displays

Below the charts, you will find detailed forecast data presented in interactive tables, organized into tabs:

- **Original Forecast**: Displays the week-by-week breakdown of the unadjusted demand forecast.
- **Adjusted Forecast**: Shows the week-by-week breakdown of the forecast after applying lead time and buffer stock adjustments, including the buffer stock components.
- **Comparison**: Provides a side-by-side comparison of the original and adjusted forecasts, along with the difference and percentage change between them.

These tables provide a granular view of the forecast data, allowing for detailed analysis.

6.5. Model Performance Metrics

If historical data is available, the application will also display key model performance metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics help in evaluating the accuracy of the forecasting model.

7. Troubleshooting Common Issues

Here are some common issues you might encounter and how to resolve them:

- **Missing Dependencies**: If you encounter `ModuleNotFoundError` or similar errors, it means one or more required Python libraries are not installed. Ensure you have activated your virtual environment (if used) and run `pip install streamlit pandas numpy plotly scikit-learn`.
- **Incorrect File Format**: The application expects a CSV file with specific column names (`WEEK`, `UNITS`, `SALES` are mandatory). If your file has different column names or is not in CSV format, you will receive an error. Ensure your CSV file is correctly formatted and contains the necessary columns.
- **Port Conflicts**: If you receive an error indicating that the port (default 8501) is already in use, it means another application is using that port. You can try launching Streamlit on a different port using the `--server.port` flag: `bash streamlit run demand_forecasting_app.py --server.port 8502`
- **Data Preprocessing Errors**: The `WEEK` column must contain valid dates that can be converted to datetime objects. If you see errors related to date conversion, check the format of your `WEEK` column in the CSV file.
- **Not Enough Data Points for Forecasting**: The application requires a minimum number of data points (at least 10 complete records) to train the forecasting model. If your uploaded dataset is too small, you will receive an error.

8. Sample Data Structure

For your convenience, here is an example of the expected CSV data structure. You can create a CSV file with similar columns and data to test the application.

WEEK	STORE	STORE_TYPE	PRODUCT	CATEGORY	UNITS	SALES	BASE_PRICE
2024-01-01	Store_A	Type_1	Product_X	Category_1	100	1000	10
2024-01-08	Store_A	Type_1	Product_X	Category_1	150	1500	10
2024-01-15	Store_A	Type_1	Product_X	Category_1	120	1200	10
2024-01-22	Store_A	Type_1	Product_X	Category_1	130	1300	10
2024-01-29	Store_A	Type_1	Product_X	Category_1	110	1100	10

This table represents a minimal set of columns. Your actual data may include more columns as mentioned in Section 6.1.

9. Conclusion

By following these instructions, you should be able to successfully set up and run your Streamlit Demand Forecasting Application. The interactive interface allows for flexible data analysis and supply chain optimization, providing valuable insights for your business. If you encounter any further issues, refer to the troubleshooting section or consult the Streamlit documentation for more advanced help.