README File

How to run the Python file

Step 1. Download and install Python 3.7 or higher
Step 2. Create a virtual environment (this is only optional)
   a. Open a terminal
   b. Find the folder where the Python code will be
   c. Use the command [python -m venv venv] or [python3 -m venv venv] depending on the version

Step 3. Install the required libraries.
   a. In this case, run the code [pip install streamlit pandas plotly numpy]

Step 4. Save the code into the folder.
Step 5. Run the code on the terminal.
   a. Navigate to the folder where sunsynk_dashboard.py is saved
   b. Run the command [streamlit run sunsynk_dashboard.py]
   c. This will open the browse at http://localhost:8501

Step 6. Then upload a CSV file.
   a. On the webpage, there is a section on the sidebar which you can import the .scv file into

Step 7. Then follow all the options on the sidebar (this allows you to select columns, choose plot types and adjust settings for example).
Step 8. Stop the app
   a. When you are done press [Ctrl + C] in the terminal to stop the app.

Video demo showing the dashboard and visualisation

The video demo will be included in the zip file saved as demo.mp4

Overview

1. **Data Upload and Preprocessing:**
   - Users can upload a CSV file to load data into the app.
   - Any Timestamp columns used are then converted to datetime for easier analysis.
   - *Why this code: The file_uploader widget makes file import simple, and pd.to_datetime ensures proper handling of timestamp data.*
2. **Column Selection for Plotting:**
   - Users can use search filters to filter numeric and datetime columns for the X and Y axes. This allows easy selection of relevant data for visualisation.

- *Why this code: The text input and multi-select dropdowns allow easy column selection, streamlining data visualisation.*

3. **Plot Type Selection:**
    - Users can choose between scatter plots or line graphs to visualise the data.
    - Scatter plots can be used for distribution, and line graphs for trends.
    - *Why this code: Offering both plot types provides flexibility—scatter for distribution and line for trends.*
4. **Colour Customisation:**
    - Users can customise plot colours to differentiate between data series.
    - *Why this code: Colour pickers enable easy visual distinction, improving plot clarity.*
5. **Forecasting:**
    - Users can select a numeric column to forecast and a timestamp column as the feature.
    - Forecasting uses random values for now but can be enhanced later.
    - *Why this code: The random values provide a placeholder for future forecasting models.*
6. **Interactive Visualisation:**
    - Plotly is used for interactive charts, enabling users to explore data dynamically.
    - *Why this code: Plotly's interactive features allow users to engage with data, zoom, and explore trends.*
7. **User-Friendly Interface:**
    - A clean layout with clear labels guides users through uploading, selecting columns, and customising the plots.
    - *Why this code: The simple interface ensures ease of use and enhances the user experience.*

This design ensures flexibility, interactivity, and user control for easy data exploration and forecasting.

Overview and analysis (concerning the database)

From looking at the first 10 rows of this dataset, the dataset provides key information about the performance of an electrical system. It shows how much active (working) and reactive (supporting) power the system is using, as well as the voltage, current, and battery health. The data shown and the data that is visualised and developed by the dashboard suggest the system is running well: power levels are stable, voltage and current are within safe limits, and the battery is in good condition with no problems or warning signs.

This data would fit well into my program web dashboard as it helps track the system's performance. It can be used to monitor important things like power usage, battery levels, and overall system health. Since the system isn't showing any faults or problems, the data indicates everything is running smoothly, making it a good match for ensuring the system stays reliable and safe in the future. In which with use of the forecast feature used in my program can show this.