Visualization of Machine States

The goal of this project is to visualize machine states based on power usage data to help users understand equipment utilization. The project involves analyzing pump data over a day, determining machine states, and creating intuitive visualizations to convey insights.

Data Analysis:

- The pump data is loaded from a CSV file using the pandas library.
- A function (`extract_psum`) is defined to safely parse JSON data within each row and extract the `Psum` value.
- The `extract_psum` function is applied to create a new column (`Psum`) containing the average power values.
- Rows where `Psum` couldn't be extracted are removed.
- The operating load is calculated as the mean of the top 10 `Psum` values.
- Machine states (`State`) are determined based on the calculated `Psum` and operating load.
- The 'fromts' column is converted from Unix timestamp to datetime format.
- The processed data is exported to a JSON file (`Processed_machine_states.json`).

Data Visualization:

- Visualizations are created to provide insights into machine states and power usage:
 - Average power draw over time is visualized using line charts.
 - Operational efficiency is represented by a pie chart showing the distribution of machines across different states as a percentage.
 - The number of machines in each state is displayed using a bar chart.

API Implementation

API Endpoints:

1. /process_data
    a. Method: GET
    b. Functionality:
        i. Fetches raw data from a predefined source.
        ii. Calculates 'Psum' value from 'metrics' column.
        iii. Classifies machine state based on 'Psum'.
        iv. Stores processed data.
    c. Response:
        i. Success message: "Data processed successfully"
    d. Example Usage:
        i. GET /process_data

2. /power_data
    a. Method: GET
    b. Functionality:
        i. Fetches data points with timestamps and 'Psum' values.
        ii. Accepts parameters for time range.
    c. Response:
        i. Success message: "Data fetched successfully"
        ii. Data format: JSON array with 'fromts' (timestamp) and 'Psum' (power draw).
    d. Example Usage:
        i. GET /power_data?start_timestamp=<start>&end_timestamp=<end>

Chart Visualization:

- Data from /process_data is used to generate charts.
- Bar chart shows the frequency of machine states.
- Pie chart illustrates the distribution of states.

Power Usage Insights:

- The API can integrate the calculation of average power usage for each machine state.
- It can either extend the /process_data endpoint to include this calculation or introduce a new endpoint specifically for power usage insights.

Enhancements

- Real-time Data Visualization:
    - Implement real-time updates and a live dashboard feature for operators to monitor machine states and power usage in real-time.
- Historical Data Analysis:
    - Extend APIs to support querying historical data for specific time ranges.
    - Develop additional chart visualizations, such as line charts or area charts, to analyze trends and patterns over time.
- Alarm and Alerting System:
    - Implement APIs to monitor machine states and trigger alerts based on predefined thresholds.
    - Develop visualizations and alerting mechanisms to notify operators of potential issues in real-time.
- Comparative Analysis:
    - Develop APIs to compare machine states and power usage across different machines or production lines.
    - Create visualizations, such as side-by-side bar charts or stacked area charts, to facilitate comparative analysis.
- Predictive Analytics:
    - Integrate machine learning models to forecast future machine states and power usage.