Singapore Office Carbon Emissions Calculator

LI ZHUOYANG

2024/11/12

Overview

This project is a web application designed to calculate and visualize carbon emissions for office buildings in Singapore. It includes two calculators:

- **Standardized Calculator**: Calculates total carbon emissions based on user inputs.
- **Prediction Calculator**: Predicts emissions using a machine learning model.

Users can input data manually or upload CSV files. The results include total emissions, data visualizations comparing emissions to industry averages, and a personalized report generated by OpenAI's API.

Display Adaptation (1080p)

This application has been designed to deliver the best visual and interactive experience on **24-inch displays with 1080p resolution**. To ensure compatibility:

- **Target Resolution:** 1920×1080 (Full HD).
- **Scaling Recommendations:** Adjust the browser's zoom level based on your display settings:
  - For a 24-inch monitor: Use a zoom level of 100%.
  - For smaller screens or higher resolutions: Adjust the zoom level to fit the layout properly.
- For unsupported resolutions or devices, resizing or scaling the browser window may help achieve better compatibility.

Technologies Used

Frontend

- HTML: `index.html`
- CSS: `index.css`
- JavaScript: `script.js`
- Visualization Tool: Chart.js

Backend

- **OpenAI API**: `https://api.openai.com/v1/chat/completions`, used for generating personalized reports.
- **Local Machine Learning API**: After installing Flask and Flask-CORS, the API can be started from the `5105_ml` directory by running `python app.py` in the terminal. The `app.py` script processes the `best_rf_model.pkl` file, converting it into a JSON format readable by JavaScript for frontend integration.

Setup Instructions

Important Notes

Due to differences in screen resolution and scaling settings across devices, users may need to manually adjust the browser zoom level for optimal display. For a 24-inch 1080p monitor, a zoom level of 90% is recommended.

Frontend

1. Clone the repository:

   ```
   git clone <repository_url>
   cd <repository_folder>
   ```

2. Open the `index.html` file in a browser.

Backend API

1. Navigate to the `5105_ml` folder.
2. Install required dependencies:

   ```
   pip install flask
   ```

   ```
   pip install flask-cors
   ```

3. Start the local API server:

   ```
   python app.py
   ```

4. Ensure the `best_rf_model.pkl` file is in the same directory as `app.py`.

Connecting Frontend and Backend

Update the API endpoint in `script.js` to match the local backend server, e.g., `http://127.0.0.1:5000`

How to Use

1. Input data manually or upload a CSV file.
2. View the results:
   - Total carbon emissions.
   - Visualizations comparing emissions to industry averages.
3. Generate a personalized report based on user data and emissions results.

Folder Structure

```
index.html             # Frontend HTML file
index.css              # Frontend CSS file
script.js              # Frontend JavaScript file
5105_ml/               # Backend folder
   app.py             # Flask application script
   best_rf_model.pkl # Machine Learning model
README.md              # Project documentation
```

Future Improvements

- Add more prediction models for higher accuracy.
- Enhance frontend design for better user experience.
- Deploy backend API to a cloud service for wider accessibility.