Monthly Blog 6

Team 22
Jan. 24 - Feb. 24

Once the elevator pitch is finished, we started the code implementation part. Unlike the previous part, we did not provide as much visual progress as before to show (like speech, slides), so we reduce the frequency to write the report to once a month, and focus more on the actual code implementation part.

0 Emergency

No Access to Actual Data

Before the code section began, we have encountered serious issue. Due to Volvo's protection of data privacy, we were unable to access the real carbon emission data, which greatly affected the progress of our project. After discussed to our client from Avanade, they suggest that they could provide us the access to an Azure resource project with dummy data, so that we could build the backend based on it. In this case, once he successfully requests the access to the real data, we just need to transfer the resource group to the actual data.

1 Progress

In spite of the unavailability of real data, we have made great progress this month in each part of our system.

1.1 Front End Implementation

During this month, Ramit worked on the frontend part of our dashboard, by learning how to work with frontend infrastructure React, and investigate what kind of icons and charts we could make use of, he built the framework for the front-end and added the appropriate charts to the framework using react library such as the Nivo chart and Material-UI.

Here are some screenshot from the dashboard by the end of this period. In the current progress, we have completed:

- Charts used to present data: line chart, bar chart, pie chart, map.
- Functional buttons: Button to change themes(light / dark) , button to hide sidebar.
- Interact-able component: Hover over the chart to display specific data.

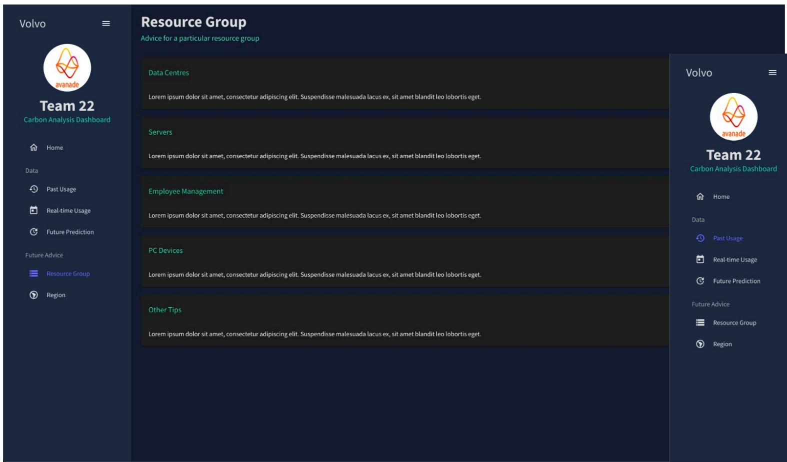

Figure 1: Screenshot of Resource Group Page

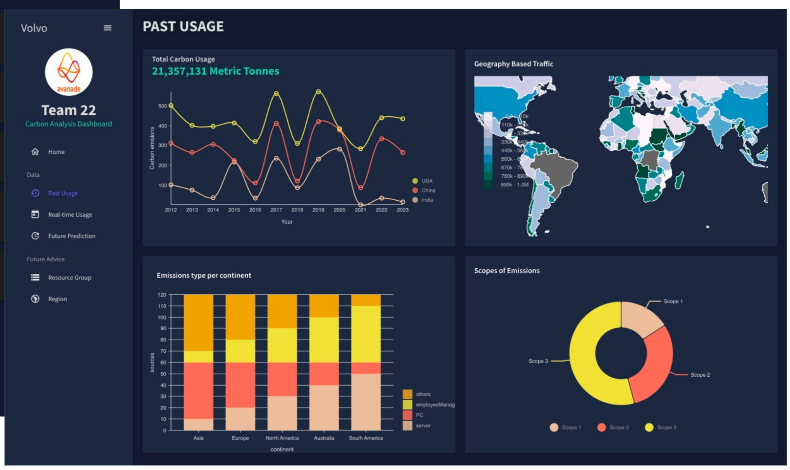

Figure 2: Screenshot of Past Usage Page

1.2 Back End Implementation

Sam is responsible for writing the backend programs. During this month he completed:

1. The Azure client.
   - To fetch carbon emission data from Azure resource Group.
   - Use a variety of functions to get corresponding data from given resource group.
2. Electricity Mapper Integration.
   - Fetch unprocessed information from Azure resource group, including the percentage of renewable / fossil-free energy use, the percentage of different energy types used.
   - Integrating and analysing metrics information from Azure, and calculating actual carbon emissions data to be displayed.
3. API Endpoints.
   - Provide nodes to the frontend application
   - Integrate the APIs from the previous two applications, use the App Routing provided by Flask to return data to the frontend via different URLs.

```python
@app.route("/locations")
@app.route("/locations/<resourceGroup>")
def get_locations(resourceGroup: str=None):
    if resourceGroup is not None:
        resources: list[GenericResourceExpanded] = azure_client.get_resources_in_group(resourceGroup)
    else:
        resource_groups: list[str] = azure_client.get_resource_groups()
        resources: list[GenericResourceExpanded] = []
        for resource_group in resource_groups:
            resources.extend(azure_client.get_resources_in_group(resource_group))

    locations = set(map(lambda resource: resource.location, resources))

    return {
        "value": sorted(list(locations))
    }
```

Figure 3: A piece of code in Endpoint

By using a client, backend-Azure interaction and frontend-backend interaction design pattern, the architecture of the backend is clearly outlined and easy to manage and maintain

1.3 Algorithm Investigation & Implementation

For this part, Tony and William have investigated the model can be used to generate prediction.

Linear regression, the one introduced in the algorithm literature review, was vetoed as the very early stage. Since we realised that the actual carbon emission data varies frequently based on the time of a day, it can't be simply treated as a linear equation.

After comparing different neural networks, we decided to use the LSTM neural network. Compared to other neural models, it is more suitable for processing and predicting important events with long intervals and delays in the time series, which we believe will be a well suite for our dataset. Then, over the following weeks, we constructed a model based on this network.

For more information, Tony will write a paper later to analyse all the algorithm we used in detail.

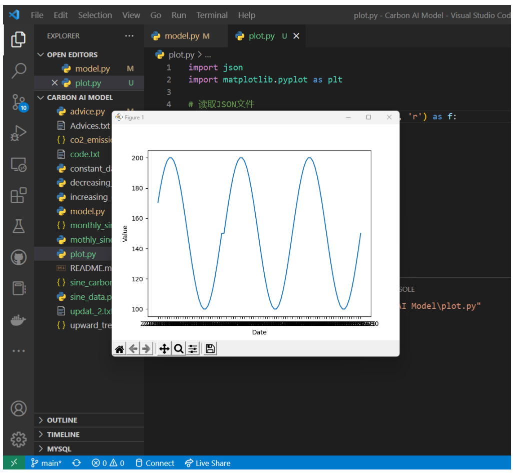

Figure 3:
Predictions of our LSTM model for the sin function