Step-by-Step Tutorial: Setting Up and Running the Subjective Wellbeing Platform

This guide will help you set up, configure, and run the **Subjective Wellbeing Platform**, which includes:

- **Interactive Dashboard**: Explore trends and predictions for wellbeing indicators.
- **Interactive Map**: Visualize wellbeing data geographically.
- **Data Viewer**: Browse and filter raw survey data interactively.

---

1. Install Required Libraries

Why This Step Is Important

The platform relies on Python libraries for:

1. **Backend Development**: Flask enables routing and web server capabilities.
2. **Data Visualization**: Dash and Plotly are used for creating interactive graphs.
3. **Geographic Mapping**: Folium generates interactive maps with heatmaps and markers.
4. **Data Processing**: Pandas processes and cleans the dataset.

Steps

Run the following command to install all the required libraries:

In [ ]:

```
!pip install flask dash pandas folium plotly dash-bootstrap-components
```

2. Set Up the Project Directory

Why This Step Is Important

Organizing your files ensures:

- All components of the project (code, templates, data) are in the correct place.
- The application can locate datasets, templates, and scripts without errors.

Steps

1. Create a folder for your project:
   `mkdir subjective_wellbeing_platform`
2. Inside this folder, organize the files and directories as follows:
   ```
   subjective_wellbeing_platform/
   ├── app.py                    # Main Flask application
   ├── dash_app.py               # Dash app for the interactive dashboard
   ├── map.py                    # Folium map generation logic
   ├── templates/                # HTML templates for Flask
   │   ├── index.html            # Home page
   │   ├── dashboard.html        # Embedded Dash app page
   │   ├── map.html              # Interactive map page
   │   ├── data.html             # Data viewer page
   ├── static/                   # Static assets (CSS, JS, images, etc.)
   ├── data/                     # Data storage
       ├── processed/            # Processed datasets
       │   ├── subjective_wellbeing_cleaned.csv
       ├── predictions/          # Predictions data
           ├── predictions.csv
   ```
3. Place your cleaned dataset (`subjective_wellbeing_cleaned.csv`) in the `data/processed/` folder.

3. Run the Flask Application

Why This Step Is Important

Flask acts as the backend server, rendering HTML pages and hosting the dashboard and map functionalities.

Steps

1. Open your terminal or code editor.
2. Navigate to the project folder:
   `cd subjective_wellbeing_platform`
3. Run the Flask application:
   `python app.py`
4. Once the server starts, you'll see an output like this:
   `* Running on http://127.0.0.1:2022/ (Press CTRL+C to quit)`
5. Open the URL (`http://127.0.0.1:2022/`) in your browser to access the platform.

4. Explore the Platform

The platform includes three key features: the **Dashboard**, **Interactive Map**, and **Data Viewer**. Here's how to use each:

4.1 Interactive Dashboard

Purpose

The dashboard visualizes wellbeing trends over six years and provides predictions for 2024–2025.

Steps to Access

1. On the home page, click "Dashboard" or navigate to:
   `http://127.0.0.1:2022/dashboard-page`

How to Use

1. Use the dropdown filters at the top of the page to select:
   - **Subtopic**: E.g., Personal Health, Community Connection.
   - **Age Group**: E.g., 18–24, 25–34.
   - **Gender**: Male or Female.
   - **Suburb**: Choose a location to narrow the analysis.
2. The line graph updates dynamically based on your selections:
   - **X-axis**: Year.
   - **Y-axis**: Wellbeing percentage.
   - Predictions for 2024 and 2025 are shown as distinct points.
3. Reset filters by selecting "All" in the dropdown menus or refreshing the page.

4.2 Interactive Map

Purpose

The map highlights wellbeing data geographically using heatmaps and cluster markers.

Steps to Access

1. On the home page, click "Map" or navigate to:
   `http://127.0.0.1:2022/map`

How to Use

1. Use the filters to refine the map:
   - **Subtopic**: Select an indicator (e.g., Satisfaction with Safety).
   - **Year**: View data for a specific year or "All".
2. The map features:
   - **Heatmap**: Displays intensity using color gradients (red for high, blue for low).
   - **Markers**: Clickable points showing details for each location.
3. Click "Update Map" to apply filters.

4.3 Data Viewer

Purpose

The data viewer displays the raw dataset in a table format with interactive filters.

Steps to Access

1. On the home page, click "Data Viewer" or navigate to:
   `http://127.0.0.1:2022/data`

How to Use

1. Use dropdown filters to narrow down the data:
   - **Subtopic**: Filter by wellbeing indicators.
   - **Category**: Select Age Group, Gender, or Suburb.
   - **Year**: Choose a specific year or "All".
2. Filtered data will appear dynamically in the table.

5. Troubleshooting

Common Issues and Fixes

1. **Flask App Not Starting**:

   - Ensure you are in the correct directory (`cd subjective_wellbeing_platform`).
   - Check Python is installed and dependencies are installed:
     `pip install flask dash pandas folium plotly dash-bootstrap-components`
2. **Page Not Loading**:

   - Verify the Flask app is running. If not, restart it using:
     `python app.py`
3. **Slow Loading**:

   - For large datasets, delays may occur. Optimize the dataset or preprocess it to include only relevant columns.
4. **Filter Errors**:

   - Ensure your dataset contains consistent column names.
   - Check for typos in the dropdown selections.

6. Advanced Tips

1. **Adding New Data**:

   - Add new cleaned datasets to the `data/processed/` folder.
   - Restart the Flask app to load the new data.
2. **Customizing Filters**:

   - Modify filter logic in `dash_app.py` or `map.py` to include additional categories or metrics.
3. **Deploying the Platform**:

   - Use deployment platforms like Heroku or AWS to make the platform accessible online.
4. **Exporting Data**:

   - Implement an "Export Data" button in the Data Viewer to allow users to download filtered datasets as CSV files.