

# ACT Visualization

## Overview

This project is performing text analysis on the data from the [Artificial Intelligence Act](#) website. I manged several visualizations using NLP techniques and the findal output is available on [Github Pages](#)

## Dockker

To build the docker image, run the following command:

```
docker build -t act-visualization .
```

To render the report, run the following command:

```
docker run --rm -v $(pwd)/docs:/app/docs act-visualization
```

The rendered report will be available in the docs directory.

```
open docs/index.html
```

## Installation

### 1. Clone the Repository

```
git clone https://github.com/kkasra12/act_visualization.git  
cd act_visualization
```

### 2. Install Dependencies

Ensure you have Python installed. Install the required Python packages using:

```
pip install -r requirements.txt
```

### 3. Install Quarto

Quarto is required to render reports and visualizations. Install it by following the instructions for your operating system:

- **Windows:** Download and install from [Quarto's website](#)
- **macOS:** Use Homebrew:

```
brew install quarto
```

- **Linux:** Use the following commands:

```
sudo apt install quarto # Debian/Ubuntu  
sudo dnf install quarto # Fedora
```

Verify the installation:

```
quarto --version
```

#### 4. Set Up OpenAI key (Optional)

Go to OPENAI website and get an API key. Then copy is to the `config.py` file.

```
API_KEY='your_api_key_here'
```

Note: This step is optional. If you don't have an API key, you can still run the project without it and it will use the existing data.

## Project Structure

- `classifier.py`: Script for classifying genomic data.
- `scraper.py`: Script for scraping relevant data.
- `node.py`: Contains node definitions used in data processing.
- `requirements.txt`: Lists the Python dependencies.
- `index.qmd`: Quarto file containing the main analysis and visualizations.
- `config.py`: Configuration file for storing API keys and other settings.
- `images/`: Directory containing images used in the project.
- `tests/`: Directory containing test scripts.
- `.github/workflows/`: Contains GitHub Actions workflows for CI/CD.

## Test

to run the tests, run

```
python -m pytest
```