REPORT OF INTERSHIP PROGRAM 2024

On

" CO2 EMISIIONS ANALYSIS"

Mentorness

Submitted by:

M.Sivapriya

MIP-DA-14

CO2 EMISIIONS ANALYSIS

****Sources of CO2 Emissions****

- - **Fossil Fuels**: Burning coal, oil, and natural gas for energy and transportation.
- - **Industrial Processes**: Cement production, steel manufacturing, and chemical processes.
- - **Deforestation**: Trees absorb CO2; cutting them down releases stored carbon.
- - **Agriculture**: Livestock production and land use changes contribute indirectly.

****Trends Over Time****

- - **Historical Data**: Review of emissions data over decades to identify trends.
- - **Global vs. Local**: Comparison of emissions by country or region.
- - **Impact of Policy Changes**: Analysis of how regulations or incentives affect emissions.

****Measurement and Reporting****

- - **Carbon Footprinting**: Assessing the total greenhouse gas emissions of an individual, organization, or product.
- - **National Inventories**: Countries report emissions data to international organizations (e.g., UNFCCC).
- - **Life Cycle Assessments**: Evaluating emissions throughout the life cycle of a product, from raw materials to disposal.

****Impact Assessment****

- - **Climate Change**: Understanding how CO2 contributes to global warming and climate change.
- - **Air Quality**: Assessing the health impacts of elevated CO2 levels and associated pollutants.
- - **Ecosystem Effects**: Evaluating how changing CO2 levels affect biodiversity and ecosystems.

****Mitigation Strategies****

- - **Renewable Energy**: Transitioning to solar, wind, and other renewable sources to reduce reliance on fossil fuels.
- - **Energy Efficiency**: Improving efficiency in transportation, buildings, and industry.
- - **Carbon Sequestration**: Technologies that capture and store CO2 emissions.
- - **Reforestation and Afforestation**: Planting trees to absorb CO2 from the atmosphere.

Future Projections

- - **Models and Scenarios**: Using climate models to predict future emissions and their impacts.
- - **Policy Implications**: Analyzing how proposed policies could affect future emissions trajectories.

Public Awareness and Engagement

- - **Education Campaigns**: Raising awareness about the importance of reducing CO2 emissions.
- - **Community Initiatives**: Local efforts to promote sustainable practices and reduce carbon footprints.

Tools and Resources

- - **Emission Calculators**: Online tools to estimate personal or organizational carbon footprints.
- - **Government Reports**: National and international bodies provide emissions data and analysis.
- - **Research Studies**: Academic papers and studies on emissions trends and mitigation strategies.
-
- By analyzing these aspects, we can develop a better understanding of CO2 emissions and work towards effective solutions to mitigate their impact on the environment. If you have a specific area or dataset in mind for analysis, feel free to share!nline (updates on industry competition)

Source code:

```
# Byte-compiled / optimized / DLL files

__pycache__/
```

```
*.py[cod]
*$py.class

# C extensions
*.so

# Distribution / packaging
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
pip-wheel-metadata/
share/python-wheels/
*.egg-info/
.installed.cfg
*.egg
```

MANIFEST

PyInstaller

Usually these files are written by a python script from a template

before PyInstaller builds the exe, so as to inject date/other infos into it.

*.manifest

*.spec

Installer logs

pip-log.txt

pip-delete-this-directory.txt

Unit test / coverage reports

htmlcov/

.tox/

.nox/

.coverage

.coverage.*

.cache

nosetests.xml

coverage.xml

*.cover

*.py,cover

.hypothesis/

.pytest_cache/

```
# Translations
*.mo
*.pot

# Django stuff:
*.log
local_settings.py
db.sqlite3
db.sqlite3-journal

# Flask stuff:
instance/
.webassets-cache

# Scrapy stuff:
.scrapy

# Sphinx documentation
docs/_build/

# PyBuilder
target/

# Jupyter Notebook
```

```
.ipynb_checkpoints

# IPython
profile_default/
ipython_config.py

# pyenv
.python-version

# pipenv
#   According to pypa/pipenv#598, it is recommended to include Pipfile.lock in version control.
#   However, in case of collaboration, if having platform-specific dependencies or dependencies
#   having no cross-platform support, pipenv may install dependencies that don't work, or not
#   install all needed dependencies.
#Pipfile.lock

# PEP 582; used by e.g. github.com/David-OConnor/pyflow
__pypackages__/

# Celery stuff
celerybeat-schedule
celerybeat.pid

# SageMath parsed files
*.sage.py
```

```
# Environments
.env
.venv
env/
venv/
ENV/
env.bak/
venv.bak/

# Spyder project settings
.spyderproject
.spyproject

# Rope project settings
.ropeproject

# mkdocs documentation
/site

# mypy
.mypy_cache/
.dmypy.json
dmypy.json
```

```
# Pyre type checker

.pyre/
```