edunet foundation

Greenhouse Gas Emission Prediction

Student name – Nishit Singh
Student ID –STU6843ccf7d06391749273847
Internship ID-INTERNSHIP_1748923002683e727a876ea

Learning Objectives

- Gained hands-on experience in data cleaning and preprocessing using pandas and scikit-learn.
- Built and trained Linear Regression and Random Forest Regressor models for prediction tasks.
- Learned model evaluation techniques like R^2 score and Mean Squared Error (MSE).
- Applied Hyperparameter Tuning (HPT) for model optimization.
- Used Git and GitHub for version control and collaboration.
- Learned to save and load models using joblib for deployment reuse.

Source : www.freepik.com/

Tools and Technology used

- **Programming Language**: Python
- **Data Processing & Analysis**: Used pandas and numpy for data loading, cleaning, and manipulation; scikit-learn for preprocessing.
- **Preprocessing Techniques**:
- Handled missing values and dropped irrelevant columns
- Converted categorical features using LabelEncoder
- Scaled numeric features using StandardScaler
- Split data using train_test_split
- **Machine Learning Models**:
- Trained and evaluated LinearRegression model
- Explored RandomForestRegressor and model improvement via Hyperparameter Tuning (HPT)
- **Model Evaluation**: Measured performance using R^2 Score and Mean Squared Error
- **Model Persistence**: Saved trained model and scaler using joblib for reuse
- **Version Control**: Managed code using git and hosted project on GitHub
- **Development Environment**: Used Visual Studio Code for scripting and Google Colab for exploration and training

Methodology

1. Data Collection

- Dataset includes GHG emission factors, DQ metrics, substance type, source, year, and units.
- Data collected in CSV format and loaded using pandas.

2. Preprocessing

- Removed missing or irrelevant data using pandas.
- Encoded categorical variables using LabelEncoder.
- Scaled numerical features using StandardScaler.
- Split data into features (X) and target (y).

3. EDA & Visualization

- Used matplotlib and seaborn to explore correlations and feature distributions.
- Plotted actual vs predicted values for model performance insight.

4. Model Selection

- Chose Linear Regression for its simplicity and interpretability.
- Considered features like DQ scores, margins, substance type, and year.

Methodology

5. Train Model

- Trained Linear Regression model using scikit-learn.
- Saved model and scaler with joblib for reuse in the web app.

6. Evaluate Model

- Evaluated using R^2 Score and Mean Squared Error.
- Checked prediction accuracy visually with scatter plots.

Problem Statement:

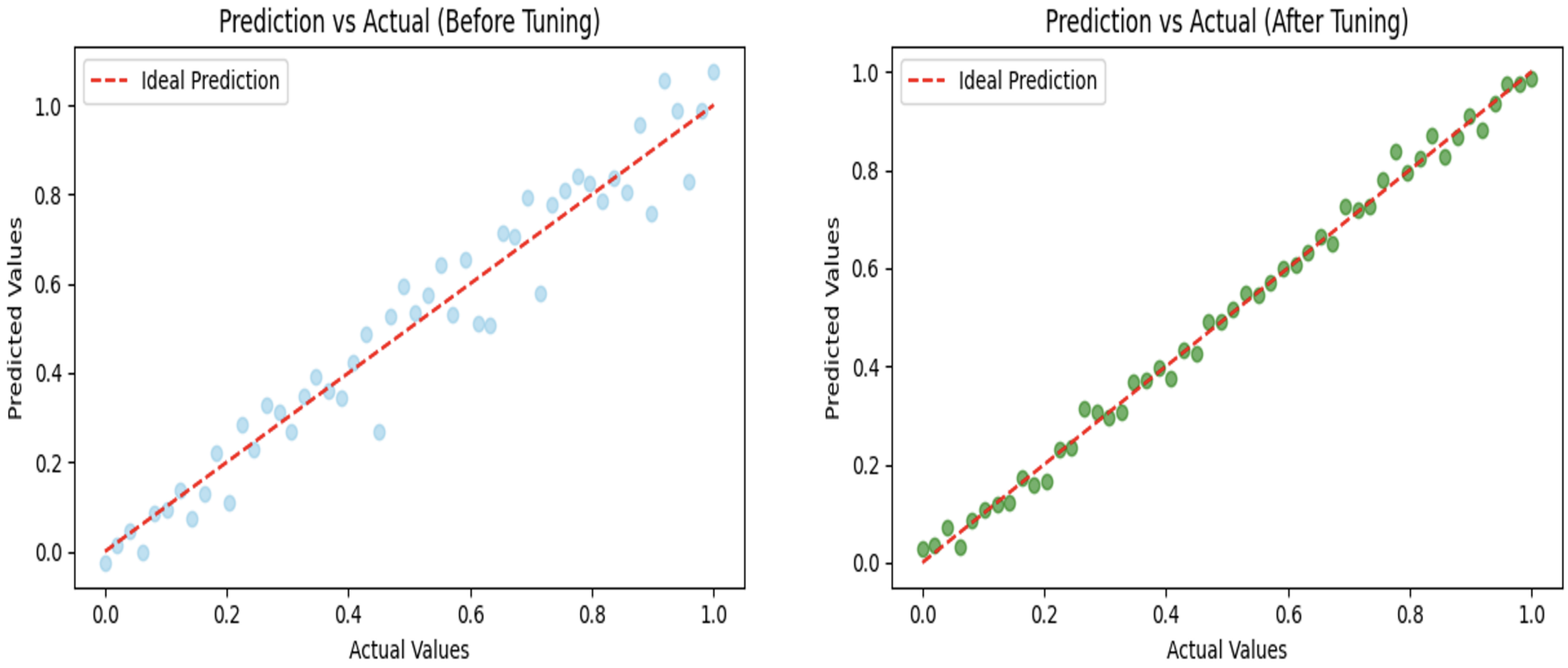
Accurately estimating greenhouse gas (GHG) emissions across supply chains is critical for sustainable industrial practices. However, emission factor data often lacks precision due to incomplete data quality metrics, outdated measurements, and inconsistent reporting standards. These gaps hinder industries from making informed decisions about environmental impact, compliance, and sustainability targets.

This project addresses the challenge by developing a predictive model that estimates supply chain GHG emission factors with margin adjustments, based on available DQ metrics and categorical attributes such as substance type, source, and year. The solution provides a scalable and consistent way to estimate emissions .

Solution:

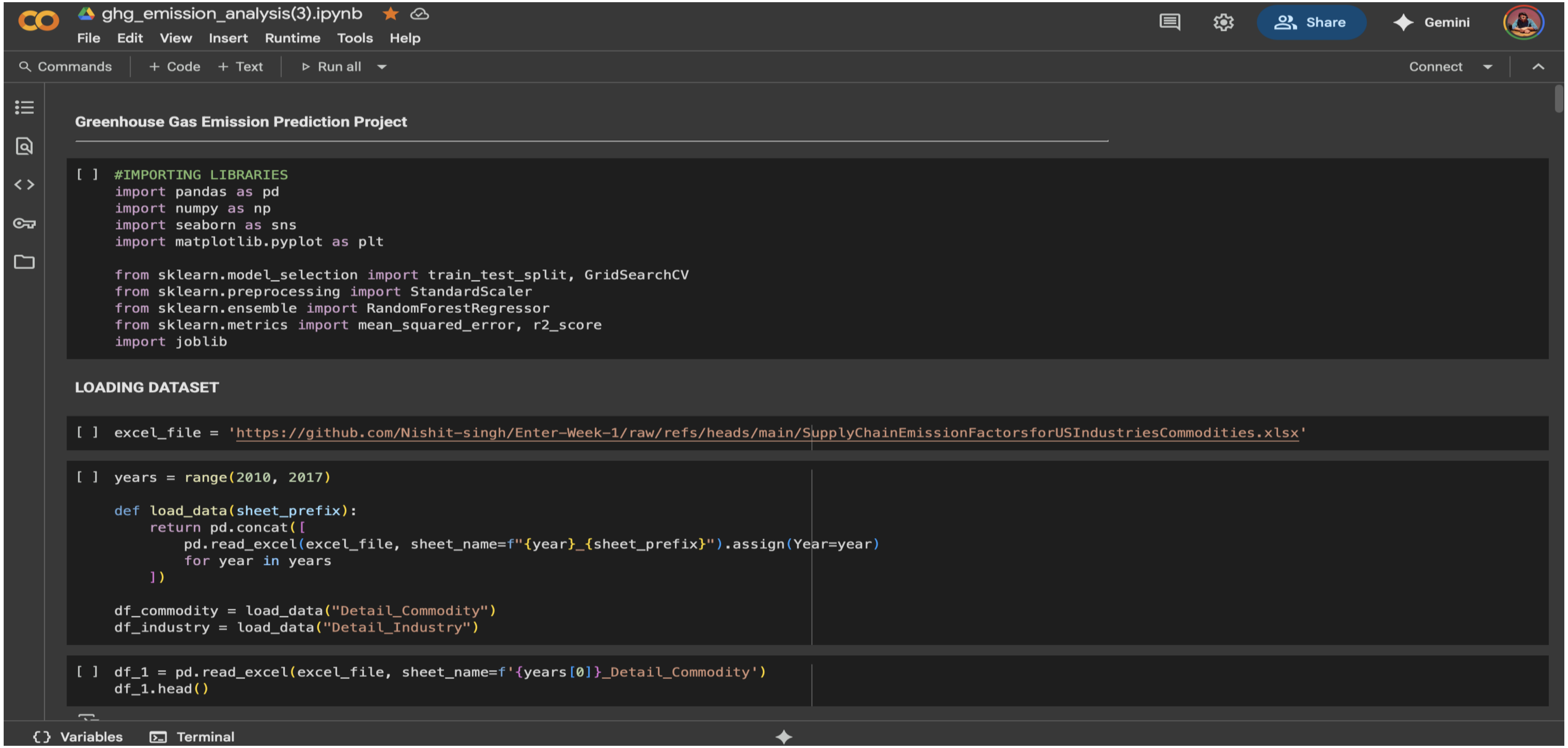
- To address the challenge of imprecise and incomplete GHG emission data, a predictive framework was developed using machine learning. The solution leverages a Linear Regression model built with scikit-learn, trained on structured datasets that include emission values, margin data, and five key Data Quality (DQ) metrics—reliability, temporal, geographical, technological, and data collection.
- Categorical variables such as substance type, source, and year were encoded and all features standardized using StandardScaler. The model achieved an R^2 score of ~0.99, indicating strong predictive performance.
- A user-friendly web app was developed using Streamlit, allowing real-time input and predictions, making the tool both accessible and scalable.
- This solution benefits:-
- Sustainability professionals seeking consistent emission estimates
- Organizations aiming to fill data gaps in reporting
- Policymakers requiring more accurate forecasting tools
- By enabling data-driven decision-making, the app promotes transparency, compliance, and greener supply chain practices.
- Github link- https://github.com/Nishit-singh/Enter-Week-1.git

Screenshot of Output:



Screenshot of Output:

Snippet of ipynb file(jupyter notebook)


ghg_emission_analysis(3).ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
Connect
Variables
Terminal

Greenhouse Gas Emission Prediction Project

```
[ ]  #IMPORTING LIBRARIES
     import pandas as pd
     import numpy as np
     import seaborn as sns
     import matplotlib.pyplot as plt

     from sklearn.model_selection import train_test_split, GridSearchCV
     from sklearn.preprocessing import StandardScaler
     from sklearn.ensemble import RandomForestRegressor
     from sklearn.metrics import mean_squared_error, r2_score
     import joblib
```

LOADING DATASET

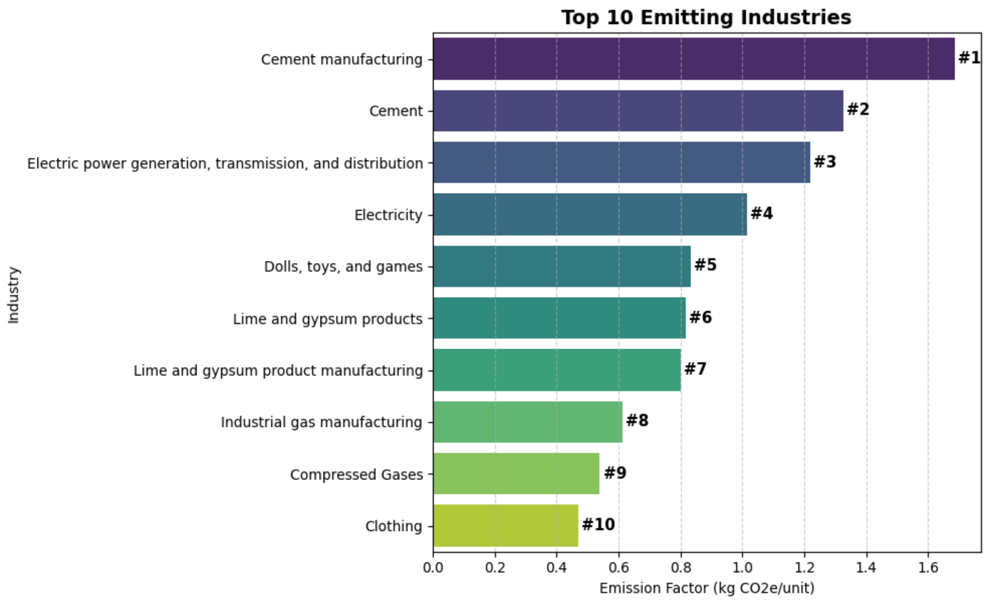
```
[ ]  excel_file = 'https://github.com/Nishit-singh/Enter-Week-1/raw/refs/heads/main/SupplyChainEmissionFactorsforUSIndustriesCommodities.xlsx'
```

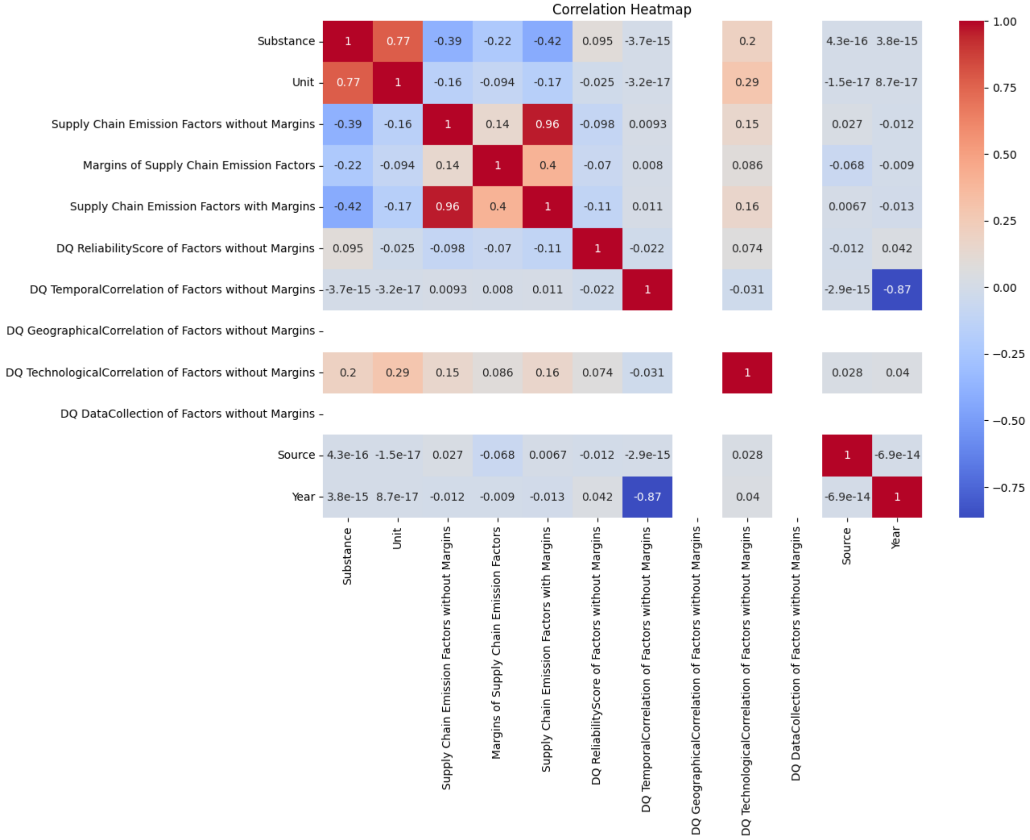
```
[ ]  years = range(2010, 2017)

     def load_data(sheet_prefix):
         return pd.concat([
             pd.read_excel(excel_file, sheet_name=f"{year}_{sheet_prefix}").assign(Year=year)
             for year in years
         ])

     df_commodity = load_data("Detail_Commodity")
     df_industry = load_data("Detail_Industry")
```

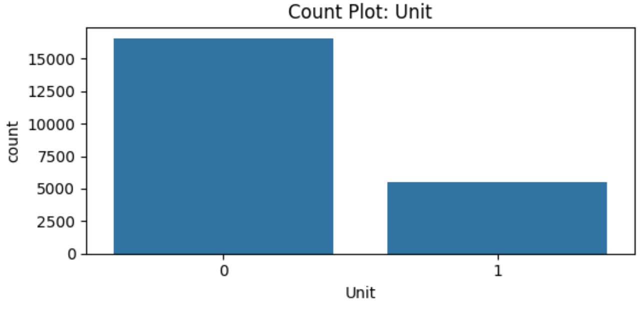
```
[ ]  df_1 = pd.read_excel(excel_file, sheet_name=f'{years[0]}_Detail_Commodity')
     df_1.head()
```

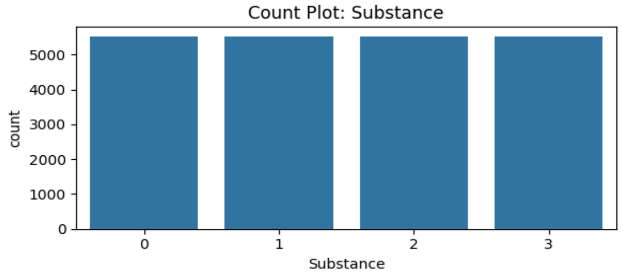
Screenshot of Output:

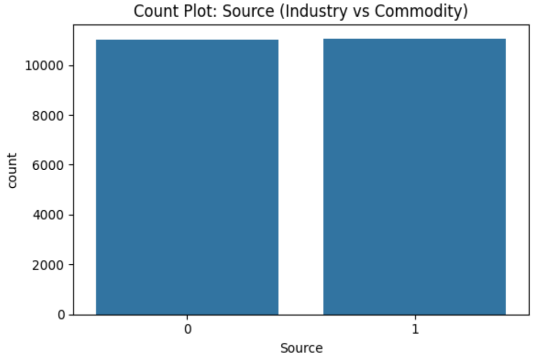
Visualization of data











Conclusion:

Working on this project has been a highly rewarding learning experience. Coming in with little to no background in environmental data or emissions modeling, I gained hands-on exposure to key machine learning concepts, data preprocessing, model training, evaluation, and deployment. Building the predictive model helped me understand how data quality metrics and domain-specific features can influence real-world sustainability challenges.

By addressing the challenge of incomplete GHG emission data, this project promotes the development of green skills combining environmental awareness with technology to enable impactful change. It stands out as a strong integration of data science, sustainability, and usability. It has been both technically enriching and personally meaningful, contributing to the broader goal of supporting a more sustainable future.

Conclusion:

Future Scope & Improvements:

- Integrate more advanced models like Random Forest or XGBoost to improve accuracy.
- Implement hyperparameter tuning for model optimization.
- Add confidence intervals to predictions for better risk assessment.
- Expand the app to support batch uploads and downloadable reports.
- Enhance explainability using SHAP values for feature contribution insights.

These improvements can further increase the tool’s accuracy, usability, and impact for real-world applications.

Github repo link- https://github.com/Nishit-singh/Enter-Week-1.git