**GHG Emission Prediction Project**

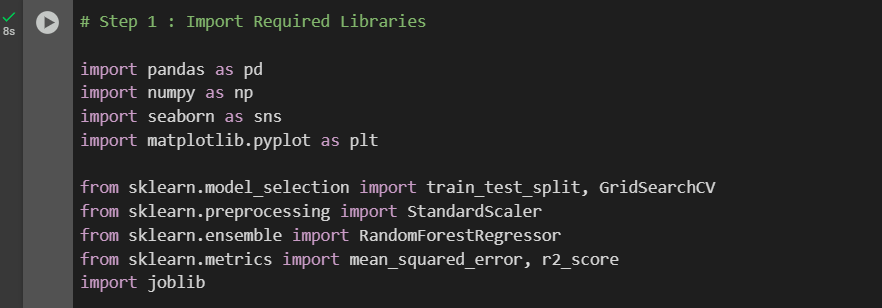
# **Day 2 -** Understanding Code for Week 01

# **Date -** 18 June 2025 (Week 1)

# **Internship Name** - Edunet-Shell Skills4Future AICTE Internship

# **Intern Name:** Abhinay Singh

**Step 1: Importing Required Libraries – Full Explanation**

****

**Explanation:**

**import pandas as pd**

📦 pandas

* **Purpose**: Used for **loading**, **managing**, and **analyzing** structured data.
* **Why used**: Our Excel/CSV dataset (GHG emissions) will be handled as a **DataFrame**, which makes it easier to manipulate rows, columns, filter data, etc.
* **Functions we’ll likely use**: read\_excel(), head(), dropna(), merge(), etc.

**import numpy as np**

**🧮 numpy**

* **Purpose:** Used for numerical operations, arrays, and mathematical functions.
* **Why used:** Many ML and statistical operations need numpy arrays, such as during normalization or matrix operations.
* **Common usage**: np.array(), np.mean(), np.isnan(), etc.

**import seaborn as sns**

**import matplotlib.pyplot as plt**

📊 seaborn and matplotlib.pyplot

* **Purpose**: For **visualization** of your dataset and model results.

**Why used**:

* seaborn: Makes complex plots (heatmaps, correlation plots) look clean and professional.
* matplotlib.pyplot: Base plotting library (like drawing canvas).
* **Common usage**:
  + Plot missing values
  + Correlation matrix
  + Barplots of feature importance

**from sklearn.model\_selection import train\_test\_split, GridSearchCV**

🔀 train\_test\_split, GridSearchCV

**Purpose**:

* train\_test\_split: Split your dataset into **training and testing** parts (usually 80/20 or 70/30).
* GridSearchCV: Used to **automatically search** for the **best hyperparameters** for models (e.g., best number of trees in Random Forest).
* **Why used**: Essential for training your model properly and improving accuracy.

**from sklearn.preprocessing import StandardScaler**

**📏 StandardScaler**

* **Purpose:** Normalize/scale numerical columns (e.g., DQ scores).
* **Why used:** Makes all numeric features have the same scale (mean = 0, std = 1) so that models perform better.
* Especially helpful for algorithms sensitive to scale (e.g., linear models, gradient descent).

**from sklearn.ensemble import RandomForestRegressor**

**🌳 RandomForestRegressor**

* **Purpose:** Your main regression model that learns from data to predict GHG Emissions.
* **Why used:** Works great with mixed data (categorical + numeric), handles outliers well, and gives high accuracy with less tuning.

**from sklearn.metrics import mean\_squared\_error, r2\_score**

**📈 mean\_squared\_error, r2\_score**

**Why used:**

* **mean\_squared\_error (RMSE):** Shows how far your predictions are from real values.
* **r2\_score:** Shows how well your model explains the variation in the data. Closer to 1 = better.

**import joblib**

**💾 joblib**

* **Purpose:** Used to save your trained ML model for later use without retraining.
* **Why used:** At the end of the project, you’ll save your model as a .pkl file and load it anytime to make predictions again.

**📘 Summary for Notes / Report**

* *"In Step 1, I imported key Python libraries required for data loading (pandas), numerical operations (numpy), data visualization (matplotlib, seaborn), preprocessing (StandardScaler), model building (RandomForestRegressor), evaluation (mean\_squared\_error, r2\_score), hyperparameter tuning (GridSearchCV), and model saving (joblib). These form the foundation of any machine learning workflow."*