
DATA DESCRIPTION DOCUMENT:
CORPORATE GHG EMISSION ESTIMATIONS

OVERVIEW

This project provides estimates of Greenhouse Gas (GHG) emissions for a vast dataset of around **47 000 listed companies**. The emission data are categorized into scopes 1, 2, 3 and 123 (sum of all scopes) for each company. Data is available from **2005 to 2023**. With around **650 000 samples**. A scientific paper detailing the methodology will soon be published, and the code is accessible in the following [GitHub repository](#).

COGEM DATASET

The employed methodology uses the best available data science techniques to derive meaningful estimates. Several dataset sources are used to this end, namely Refinitv, CDP, WorldBank, and Ember. These provide information on the financial, geographical, and extra financial context of companies. A rigorous preprocessing approach was applied to clean the merged dataset.

Three top performing Machine Learning boosting models (XGBoost, CatBoost, and LightGBM) were trained on the cleaned dataset. Evaluation was performed using a strict test dataset that accounted for biases related to the countries and subsectors within the dataset. Notably, CatBoost emerged as the top-performing model across all our four scopes (1, 2, 3 and 123). These are rounded up to the Ton of CO2 equivalent (TCO2e) before being shared.

DATASET VARIABLES

Variable	Description	Unit	Missing values	Type
company_id_year	Combination of an id and a specific year.	-	-	object
company_id	Company ID.	-	-	object
company_name	The name of the company.	-	-	object
ticker	The stock ticker symbol of the company.	-	0.03%	object
lei	Legal Entity Identifier or financial institution identifier.	-	0.56%	object
isin	The International Securities Identification Number is a 12-character code that uniquely identifies companies.	-	0.00001 %	object
fiscal_year	Year (2010 to 2023).	-	-	int64
country_of_headquarters	The country where a company's main office or headquarters is located.	-	-	object
scope1_estimated	Company's scope 1 emissions.	TCO2e	-	float64

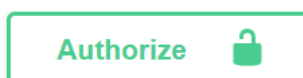
scope1_estimated_lower_bound	Company's scope 1 emissions lower bound estimations.	TCO2e	-	float64
scope1_estimated_upper_bound	Company's scope 1 emissions upper bound estimations.	TCO2e	-	float64
scope2_estimated	Company's scope 2 emissions.	TCO2e	-	float64
scope2_estimated_lower_bound	Company's scope 2 emissions lower bound estimations.	TCO2e	-	float64
scope2_estimated_upper_bound	Company's scope 2 emissions upper bound estimations.	TCO2e	-	float64
scope3_estimated	Company's scope 3 emissions.	TCO2e	-	float64
scope3_estimated_lower_bound	Company's scope 3 emissions lower bound estimations.	TCO2e	-	float64
scope3_estimated_upper_bound	Company's scope 3 emissions upper bound estimations.	TCO2e	-	float64
scope123_estimated	Company's scope 123 emissions.	TCO2e	-	float64
scope123_estimated_lower_bound	Company's scope 123 emissions lower bound estimations.	TCO2e	-	float64
scope123_estimated_upper_bound	Company's scope 123 emissions upper bound estimations.	TCO2e	-	float64
has_reported_scope1	Boolean variable indicating whether the company has reported data relevant to scope 1 emissions. This value is utilized during training processes	-	-	bool
has_reported_scope2	Boolean variable indicating whether the company has reported data relevant to scope 2 emissions. This value is utilized during training processes	-	-	bool
has_reported_scope3	Boolean variable indicating whether the company has reported data relevant to scope 3 emissions. This value is utilized during training processes	-	-	bool
has_reported_scope123	Boolean variable indicating whether the company has reported data relevant to scope 123 emissions. This value is utilized during training processes	-	-	bool

USING THE API

1.1. Web interface

To make requests from the interface available on our Pladifes website, follow the following steps:

- Click on the "Authorize" button in green.



- Enter your unique credentials that have been assigned to you.

Available authorizations

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. API requires the following scopes. Select which ones you want to grant to Swagger UI.

OAuth2PasswordBearer (OAuth2, password)

Token URL: token
Flow: password

username:

password:

Client credentials location:

client_id:

client_secret:

- After you have logged in, your session will remain active for 30 minutes. Once this time has elapsed, you will need to log in again to continue using the API. Here is an example request that allows you to download the entire database.

1.2. Additional Technical details

The COGEM API is a FastAPI application deployed on Cloud Run after containerization, and it allows to retrieve data stored in Google Cloud. To access this data from the API, a JSON Web Tokens (JWT) based authentication system is utilized to ensure data security. Please directly reach out to [Mohamed Fahmaoui](#) for more specific information about the data management.

Apart from the web interface, other way of querying the API can be used for more advanced usage.

1.2.1. From Postman

To make a request to a FastAPI API that uses JWT (JSON Web Token) authentication, you'll need to include the JWT token in the header of your HTTP request. Here's how you can do it using the Postman tool:

- Open Postman: Launch Postman on your computer.
- Create a New Request: Click on "New" to create a new request.
- Obtain a JWT Token:
 - o Choose "POST" as the request type.
 - o In the "Enter request URL" tab, enter the Authentication URL of the FastAPI Authentication endpoint.
 - o In the request body, you should typically send your credentials. This could be done in the body as form data, JSON, or other supported formats.
- Send the Request: Click the "Send" button to send the request to the FastAPI Authentication endpoint. If the credentials are correct, you will receive a JWT token in the response.
- Configure the Requested URL: In the "Enter request URL" tab, enter the URL of the FastAPI endpoint you want to call.
- Set Authorization Header: Choose OAuth 2.0 In Authorization and enter the copied token.
- Send the Request: Click the "Send" button to send the request to the FastAPI endpoint with the JWT token included in the header.
- Get the Response: Postman will display the API's response in the response section at the bottom

of the interface. You can view the JSON response, or any other format returned by the API.

1.2.2. From Terminal

To access the API from CMD terminal, you can use the curl command. Here are the general steps:

- Obtain OAuth 2.0 Access Token: First, you need to obtain an access token from the OAuth 2.0 authorization server. This typically involves making a POST request to the server with your credentials using the following command:

```
curl -X POST -d  
"grant_type=password&username=YOUR_USERNAME&password=YOUR_PASSWORD&client_id=  
YOUR_CLIENT_ID&client_secret=YOUR_CLIENT_SECRET" https://api.com/token
```

- Use the Access Token: Once you have the access token, you can include it in your API requests to authenticate yourself.

```
curl -X GET -H "Authorization: Bearer $ACCESS_TOKEN " https://api.com/endpoint
```