Quest Journals
Journal of Software Engineering and Simulation
Volume 9 ~ Issue 9 (2023) pp: 08-16
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Sustainability in the world of Generative AI

Neha Jain
Marketing Analytics,
Accenture Technology,
Gurgaon, India

Aayushi Tayal
Marketing Analytics,
Accenture Technology,Gurgaon, India

Isha Sachdev
Marketing Analytics,
Accenture Technology,Gurgaon, India

Abstract
As the world is moving at fast pace towards using Generative AI in their day-to-day life, the need to measure the carbon footprints by these models becomes a priority. We need to ensure as we are doing technological advancement at same time not destroying the nature. This paper talks in-depth about the need for Sustainable Generative AI Models as we move ahead in using them in any stream, what are the ways we can achieve that and build a sustainable world.

Keywords—*Generative AI,Conversational AI,Sustainability*

R*eceived 11 Sep., 2023;* ***R****evised 22 Sep., 2023;* ***A****ccepted 24 Sep., 2023 © The author(s) 2023.*
P*ublished with open access at* ***www.questjournals.org***

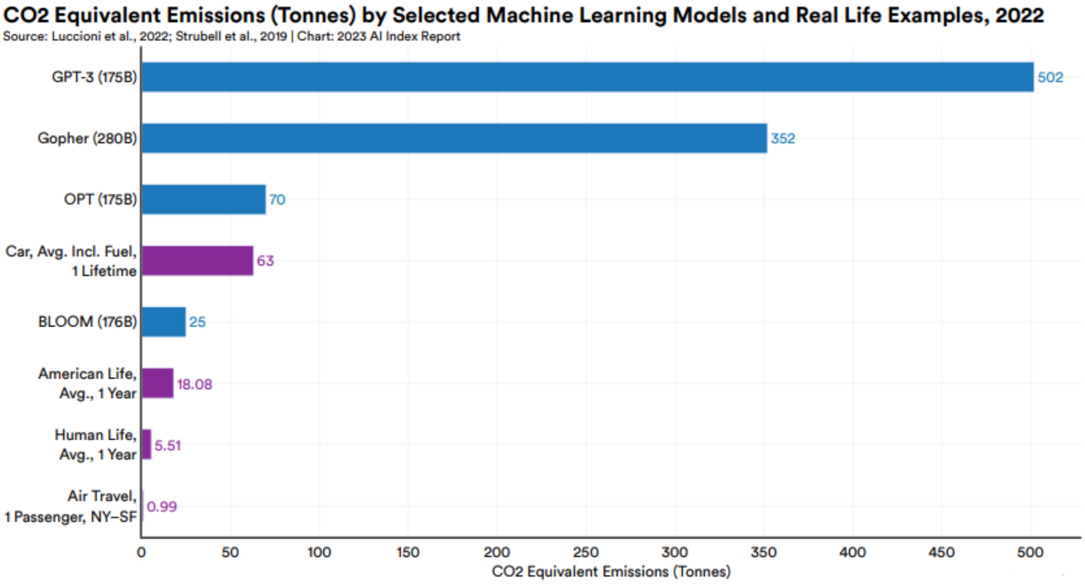
I. INTRODUCTION

Generative AI has made remarkable progress in recent years, revolutionizing the industry and transforming our lives in ways never before imagined. However, this progress comes at a price, as the increased use of deep learning algorithms leads to a significant contribution to the global carbon footprint. In this blog post, we will examine the ways in which generative AI is contributing to the global carbon footprint and the urgent need to address this issue.

Controlling CO_2 emissions will help in improving the quality of life all over the world. Therefore, every country needs to pay attention to its CO2 emissions within its region. With the rise of deep learning, the demand for computing power has increased significantly. Researchers and engineers train large-scale AI models over days, weeks, and even months. And to develop a single application, he typically trains not just one or two models, but many models to perform ablation studies that help improve KPIs. Unfortunately, this significantly increases the carbon footprint of AI technology.

The Contribution of Generative AI to the Global Carbon Footprint

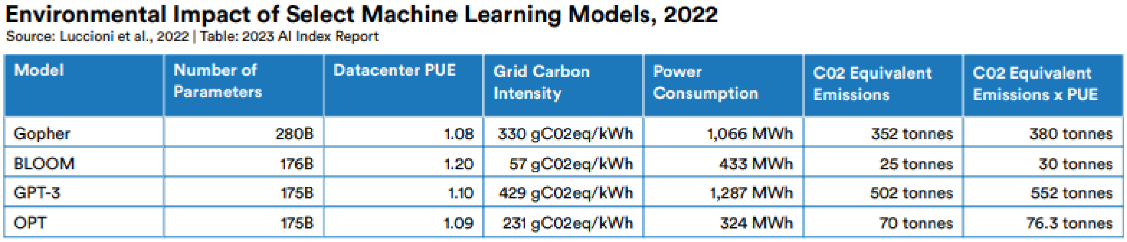
According to a 2020 report from the Massachusetts Institute of Technology, training large-scale deep learning models can emit the equivalent of 626,000 pounds of carbon dioxide. This is equivalent to the lifetime emissions of 5 cars. In a 2022 report by Stanford University, CO2 equivalent emissions by some selected machine learning models were up to 10 times the lifetime emissions of an average car.


CO2 Equivalent Emissions (Tonnes) by Selected Machine Learning Models and Real Life Examples, 2022

Source: Luccioni et al., 2022; Strubell et al., 2019 | Chart: 2023 AI Index Report

Source: Artificial Intelligence Index Report 2023, Stanford University

In this report, a metric called Power Usage Effectiveness (PUE) is used to assess the energy efficiency of data centers. PUE is the ratio of the total amount of energy consumed by a computer data center facility (this includes the energy consumption by support systems like air conditioning) to the energy delivered to computing equipment. This is how four LLMs considered in the report scored in terms of data center PUE and other factors.


Environmental Impact of Select Machine Learning Models, 2022

Source: Luccioni et al., 2022 | Table: 2023 AI Index Report

Model	Number of Parameters	Datacenter PUE	Grid Carbon Intensity	Power Consumption	C02 Equivalent Emissions	C02 Equivalent Emissions x PUE
Gopher	280B	1.08	330 gC02eq/kWh	1,066 MWh	352 tonnes	380 tonnes
BLOOM	176B	1.20	57 gC02eq/kWh	433 MWh	25 tonnes	30 tonnes
GPT-3	175B	1.10	429 gC02eq/kWh	1,287 MWh	502 tonnes	552 tonnes
OPT	175B	1.09	231 gC02eq/kWh	324 MWh	70 tonnes	76.3 tonnes

Source: Artificial Intelligence Index Report 2023, Stanford University

According to a report from the International Energy Agency, data centers, which are essential for running deep learning algorithms, account for about 1% of the world's electricity consumption. Data center energy consumption in 2020 is estimated to be 200 terawatt hours (TWh) in the United States alone, representing about 2% of the nation's total electricity consumption. Despite efforts to improve energy efficiency and use renewable energy sources, data center energy consumption is expected to continue to increase. By 2030, global data center energy consumption could reach 1,200 TWh, equivalent to the annual electricity consumption of the entire UK.

Despite continuous improvements in chips dedicated to neural network processing, the demand for GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units) for aggressive computing is still very high. Other risks associated with Generative AI include:

Waste Generation: As technology advances rapidly, the hardware used to train deep learning algorithms quickly becomes obsolete, resulting in large amounts of e-waste. According to a report by the United Nations, the world generated 53.6 million metric tons of e-waste in 2019, a figure that is projected to rise to 74.7 million metric tons by 2030.

Resource Depletion: The production of electronic devices used in training deep learning algorithms requires the extraction and consumption of natural resources such as minerals, metals, and rare earth elements. As the demand for these resources continues to rise, their depletion can have significant environmental impacts, including deforestation, pollution, and habitat destruction.

Water Consumption: Data centers, which are critical to running deep learning algorithms, require large amounts of water for cooling purposes. According to a report by the Natural Resources Defense Council, data centers in the United States consume an estimated 626 billion liters of water annually, equivalent to the annual water consumption of 7.2 million Americans. The developments in the generative AI space are expected to drive with them the consumption of water and pose a hazard to the environment.

Problem Statement

Though it's hard to measure the exact energy cost of AI models, their carbon footprint is growing alarmingly. They use more energy than other types of computing.

In fact, training a single AI model can consume more electricity than one hundred American homes use in one year. And the sector is growing so rapidly and their models are not cheap to train.

Their emissions vary a lot, depending on the type of source that powers the technology. For instance, a data center that is run by a coal or gas-fired power plant will have more CO2 emissions than one that gets power from renewable energy sources.

Greater Power Means Greater Energy

The specific energy consumption of AI models is unknown, but generally includes the space required to manufacture computing equipment, create AI models, and use them. However, there are some estimates of how big the carbon footprint of this technology is.

According to MIT Technology Review, training a single AI model can emit more than 626,000 pounds worth of CO2. That's about five times the lifetime CO2 emissions of an average passenger car. And the more powerful the AI model, the more energy it requires. In a 2019 study, researchers found that creating BERT, a generative AI with 110 million parameters, used the same amount of energy as one person would use in a transcontinental flight. found.

The number of parameters in an AI model is related to its size. The larger the model, the larger the footprint. According to computer scientist Kate Saenko, building the GPT-3, a much larger model with 175 billion parameters, emitted more than 550 tons of CO2e while consuming 1,287 megawatt hours of electricity. It says, that's the same number of emissions as one person taking 550 flights between New York and San Francisco. And this doesn't even include other sources of emissions, just preparations for AI deployment.

AI Query like ChatGPT Emits More CO2

Generative AI models were once only available to researchers, but that is changing with the release of ChatGPT by OpenAI. It also comes with the carbon footprint of the sector.

There's a lack of information on the CO2 emissions of a single generative AI query. But industry estimates show it is **4x to 5x** bigger than that of a search engine query. One Google search emits about **0.2g of CO2**.

ChatGPT had seen more than **1.5 billion** visits in March 2023 alone.

Saenko asserts the exponential growth of this AI tool as tech giants integrate them into their search engines saying:

"As chatbots and image generators become more popular, and as Google and Microsoft incorporate AI language models into their search engines, the number of queries they receive each day could grow exponentially."

The Chinese search company Baidu has also announced plans to do the same.

ChatGPT and other AI assistants have many other uses than search. They can also write, solve math problems, and create marketing campaigns.

The carbon emissions of building ChatGPT is not known publicly, but it's more likely higher than GPT-3's footprint, per Saenko. And since ChatGPT has to be updated, it can process data only until 2021, its emissions will increase even more.

Yet, generative AI's carbon footprint can be reduced.

RECOMMENDATIONS FOR SUSTAINABLE GEN AI

A study by Google found that using a more efficient AI model architecture, processor and a greener data center can reduce the tech's carbon footprint by 100x to 1,000x.

Greener data center means using power from renewable energy sources like solar or wind farms.

AI developers can also schedule computation at times when renewable sources are more available. This can cut AI's carbon footprint by as much as **30% to 40%**, compared to using a fossil fuel powered-grid.

But the more pressing concern to address is to make data on generative AI model's carbon footprint more publicly available. This is crucial to know the sector's real impact on the environment and base emission reduction efforts from there.

There are several approaches that emerge from the sustainable AI mindset that can be used to address the challenges. This includes advancing smaller models, choosing alternative deployment strategies, and choosing optimal runtimes, locations, and hardware to make systems carbon-aware and carbon-efficient.

Elevating smaller models

There are several research initiatives that are exploring how to train models faster and more efficiently that rely on pruning, compression, distillation, and quantization among other techniques with the goal of shrinking down the size of the models and utilizing fewer compute cycles which have direct implications on the financial and environmental costs of building and deploying AI systems.

With the proliferation of edge computing and IoT devices, which have limited resources for memory and computation, the field of TinyML has also seen a lot of uptake. For example, with devices that have RAM sizes in KBs, model size can be minimized along with prediction costs using approaches like Bonsai that proposes a shallow, sparse tree-based algorithm. Another approach, called ProtoNN, is inspired by kNN but uses minimal computation and memory to make real-time predictions on resource-constrained devices. Novel domain-specific languages like SeeDot, which expresses ML-inference algorithms and then compiles that into fixed points, makes these systems amenable to run on edge-computing devices.

Alternate deployment strategies

One of the essential additives of the environmental footprint of AI systems is the embodied carbon withinside the hardware this is required to run those systems. There are 3 techniques right here that may be implemented to mitigate those impacts:

(1) Use specialized hardware like ASICs and TPUs to boost up the run instances of those jobs (provided the embodied carbon is amortized over a long period of usage)

(2) Obtain higher utilization rates on existing hardware preventing idle power consumption and sub-optimal computational processing distribution

(3) Optimizing the use of existing hardware like general-purpose CPUs instead of specialized hardware which means that we reduce the demand for manufacturing new hardware to a certain extent. Each approach comes with tradeoffs but choosing the appropriate hardware platform to run jobs can result in significant carbon savings.

Federated learning ,a technique where training happens in a decentralized manner across multiple devices without the training data leaving those devices can be used as an alternative approach which can help to provide privacy protections by keeping data on-device; additionally, it can enable the compute-intensive part of the AI lifecycle to be pushed (potentially) to regions where carbon intensity is low mitigating the carbon costs of that AI system. Such a strategy helps us achieve two goals with the same change.

Carbon-efficiency and carbon-awareness

Carbon efficiency refers to optimizing at both the software and hardware level to maximize the desired performance of a system per computing unit and the energy expended to achieve that value as much as possible. It's a way of thinking. Small-scale models and alternative deployment strategies can help achieve carbon efficiency and carbon awareness.

Being carbon conscious means adapting the operating parameters of the AI system to the state of the energy grid so that it can dynamically choose the most favorable time and place to minimize the carbon footprint of the system. means to adapt. The study found that differences in the carbon intensity of the energy used to power the infrastructure behind AI systems can make a big difference between the most polluted and the least polluted regions of the United States. , showing a difference of as much as 30 times in some cases. Air pollution in Canada. This will affect where you train your AI system and where you deploy it. This is made possible through tools provided by cloud providers, such as the Azure Sustainability Calculator, and internal load balancing and balancing that cloud providers perform to meet their own sustainability goals.

Make carbon impacts a core consideration alongside functional and business requirements

Building sustainable AI systems is not just about doing right by people and our planet. Consumers are becoming more informed about the environmental impacts of the products and services they use. This has a direct impact on the business where consumers vote with their actions of choosing products and services from organizations that have greener solutions.

Running a greener version of the system will significantly reduce financial costs in most scenarios and help provide a strong business case for companies adopting a sustainable approach to AI. However, even if the financial trade-offs of this new approach are not clear, we are positioned in a competitive market as a company willing to change the way we work and build and deliver solutions that meet long-term needs and can benefit your ESG efforts. As we observed with the launch of the GDPR and subsequent pressure for organizations to conform to stricter norms with data processing, organizations that took these challenges head on stood to benefit in an increasingly competitive marketplace.

While there is no certificate or standardized way yet to report on the environmental impact of software systems, much less so AI systems, work from organizations like the Green Software Foundation towards presents a determined path forward in creating an interoperable and actionable approach that can inform consumers to make meaningful choices as they seek green solutions. Regulators and policymakers might also seek to leverage policy recommendations based on standardization work in the interest of nudging the AI ecosystem towards more sustainable practices.

II. HOW TO ESTIMATE THE CARBON FOOTPRINT OF AN AI MODEL

Before we dive into the specifics of the tools that can estimate the carbon footprint of your AI models, it is helpful to familiarize ourselves with a formula for computing carbon footprint. It is strikingly simple:

Carbon footprint = E * C

E :Number of electricity units consumed during some computational procedure. This can be quantified as kilowatt-hours (kWh).

C : Amount of CO_2 emitted from producing one of said unit of electricity. This can be quantified as kg of CO_2 emitted per kilowatt-hour of electricity and is sometimes referred to as the *carbon intensity* of electricity.

The carbon intensity varies between geographic regions, because the energy sources vary between regions. Some regions have a lot of renewable energy, some have less. Given this equation, we can now see that any tool that estimates the carbon footprint of some computational procedure must measure or estimate *E* and *C.*

Several tools exist for estimating the carbon footprint of machine learning models. It has been my experience that these tools fall into one of two categories:

1. Tools that estimate carbon footprint from *estimates* of *E (*energy consumption)
2. Tools that estimate carbon footprint from *measurements* of *E (*energy consumption)

In this post we'll take a closer look at two such tools:

1. ML CO2 Impact, which relies on *estimates* of *E* and thus falls into category 1 above
2. CodeCarbon, which relies on *measurements* of *E* and thus falls into category 2above

Note that other software packages, e.g. *carbontracker* and *experiment-impact-tracker* provide similar functionality to CodeCarbon, but I've chosen to focus on CodeCarbon as this package seems to be continuously updated and expanded, whereas the most recent commits to *carbontracker* and *experiment-impact-tracker* were made long ago.

It must be noted that the tools presented in this post only estimate the carbon footprint of the electricity used for some computational procedure. They do not, for instance, consider the emissions associated with manufacturing the hardware on which the code was run.

2.1. Estimating machine learning model carbon footprint with ML CO2 Impact

The free web-based tool ML CO2 Impact estimates the carbon footprint of a machine learning model model by estimating the electricity consumption of the training procedure. To obtain a carbon footprint estimate with this tool, all you have to do is input the following parameters:

1. Hardware type (e.g. A100 PCIe 40/80 GB)
2. Number of hours the hardware was used
3. Which cloud provider was used
4. In which cloud region the compute took place (e.g. "europe-north1")

The tool then outputs how many kilograms of CO_2 e your machine learning model emitted. It is calculated as:

Power consumption * Time * Carbon Produced Based on the Local Power Grid, e.g.:

250W x 100h = **25 kWh** x 0.56 kg eq. CO2/kWh = **14 kg CO_2 e**

The tool also shows what the emission would have been in a cloud region with a lower carbon intensity.

The benefit of a tool like ML CO2 Impact is that it can be used post-hoc to estimate the carbon footprint of your own or other people's models, and you don't have to edit your scripts to compute the estimates.

The downside of a tool like ML CO2 Impact is that it relies on estimates of energy consumption, which naturally means that its carbon footprint estimates can be off. In fact, such tools can be off by a ratio of 2.42 as illustrated by Figure 1 below.

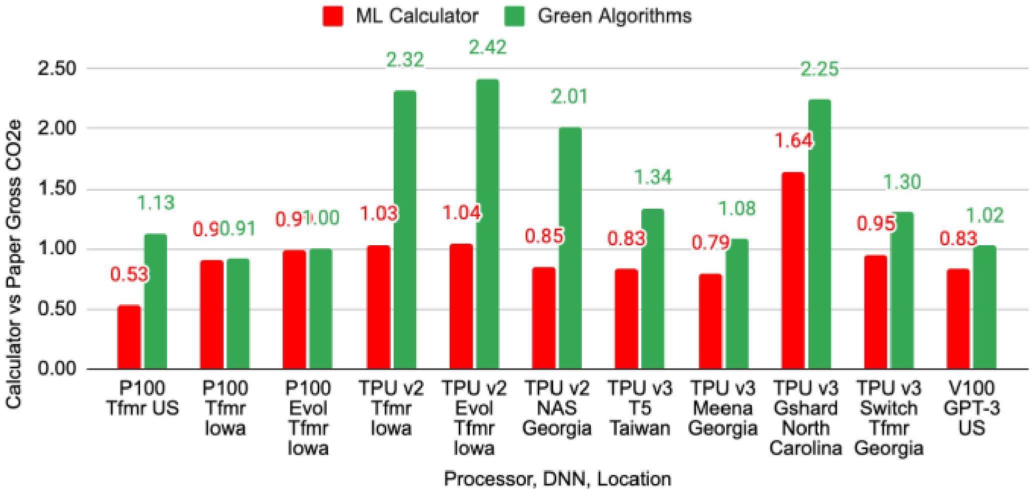


Figure 6. Ratio of ML Emissions and Green Algorithm calculators vs gross CO$_2$e in Tables 1 and 4.

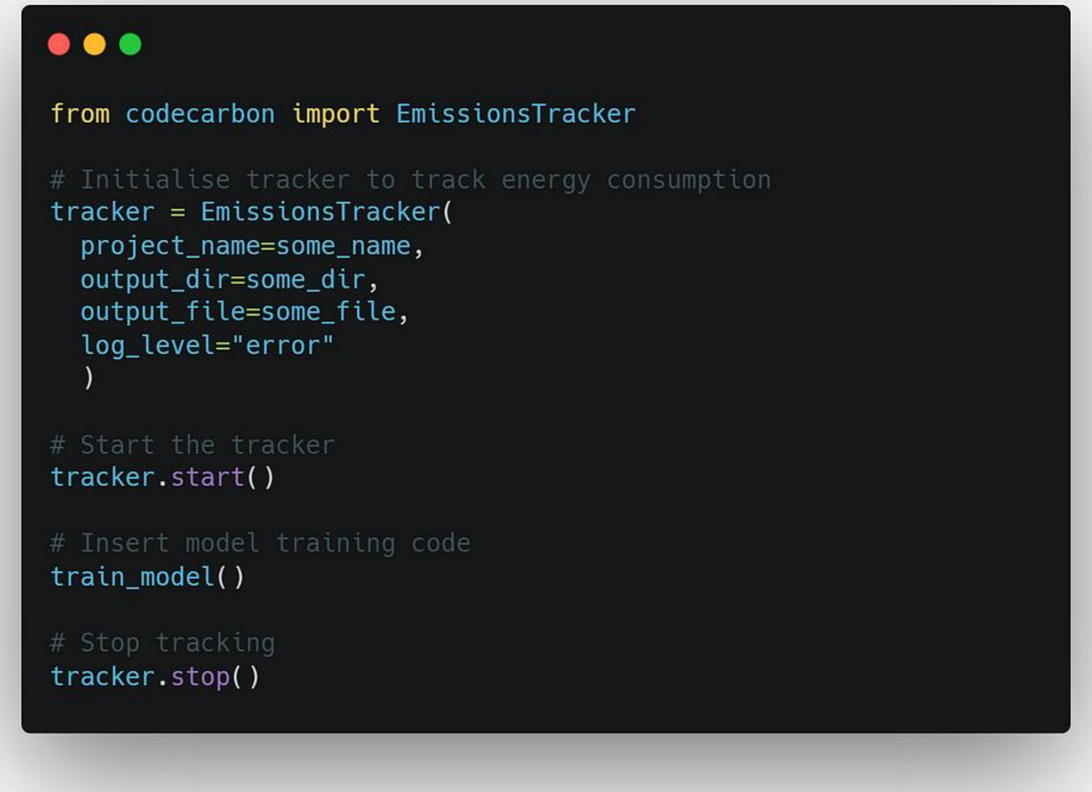
Fig. 1 [17]

2.2. Estimating machine learning model carbon footprint with CodeCarbon

CodeCarbon is a software package that is available for Python amongst other languages and can be installed by running pip install codecarbon from your command prompt. CodeCarbon computes the carbon footprint of your code like this:

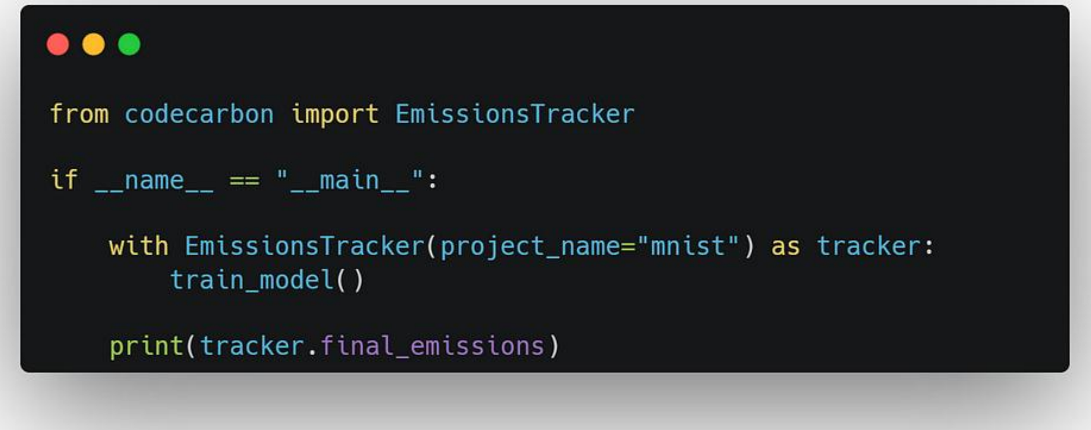
CodeCarbon directly measures the GPU, CPU, and RAM power consumption of your code running at regular intervals, such as after about 15 seconds. Note that CodeCarbon works on both local and cloud computers. This package monitors the execution time of your code and uses this information to calculate the overall power consumption of your code. CodeCarbon then retrieves information about the CO2 intensity of electricity at the hardware's geographic location. When training in the cloud, CodeCarbon automatically retrieves information about the location of your cloud instance. Then, multiply the CO2 intensity of electricity by the amount of power consumed by the code to get an estimate of the total CO2 emissions of the power consumed by the code.

The tool can be used in your code in several ways. One way is to initialise a tracker object. When the tracker object is stopped, CodeCarbon’s default behavior is to save the results to a .csv file which will contain information about how much electricity in kWh your code consumed and how much CO$_2$ e in kg this electricity emitted. Instead of writing to the file system, the information can be sent to a logger [30]. Suppose you have a function, train_model() , which executes model training, then you can use CodeCarbon like this:



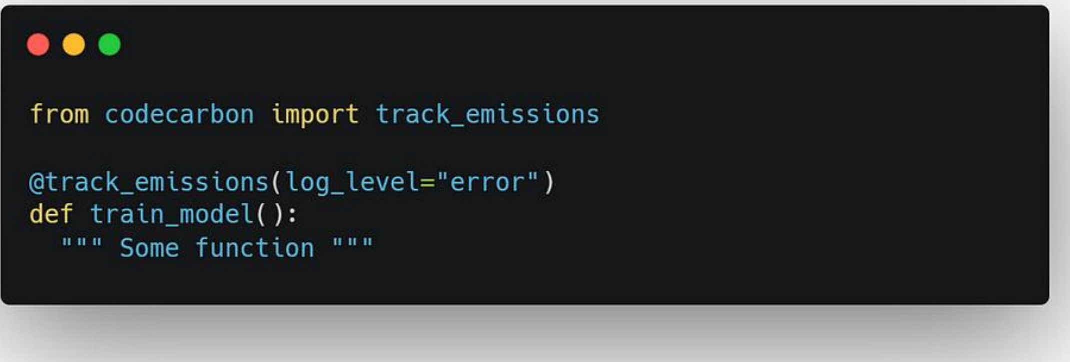
```
from codecarbon import EmissionsTracker

# Initialise tracker to track energy consumption
tracker = EmissionsTracker(
  project_name=some_name,
  output_dir=some_dir,
  output_file=some_file,
  log_level="error"
  )

# Start the tracker
tracker.start()

# Insert model training code
train_model()

# Stop tracking
tracker.stop()
```

Another way is to use CodeCarbon as a context manager like this:



```
from codecarbon import EmissionsTracker

if __name__ == "__main__":

    with EmissionsTracker(project_name="mnist") as tracker:
        train_model()

    print(tracker.final_emissions)
```

Finally, CodeCarbon can be used as a function decorator:



```
from codecarbon import track_emissions

@track_emissions(log_level="error")
def train_model():
  """ Some function """
```

Note that if the constructor argument log_level is set to its default, CodeCarbon will print out several lines of text every time it pings your hardware for its energy consumption. This will quickly drown out other information that you may be interested in viewing in your terminal during model training. If you instead set log_level="error" CodeCarbon will only print to the terminal if it encounters an error.

It is also possible to visualize the energy consumption and emissions, and the tool can also recommend cloud regions with lower carbon intensity [19].

2.2.1. Additional information about CodeCarbon methodology

Carbon Intensity (C) is the weighted average of emissions produced by the energy sources (coal, wind, etc.) used in the energy grid on which the calculation is performed. This means that the carbon footprint reported by CodeCarbon is an estimate, not an actual carbon footprint. The CO2 concentration of electricity varies throughout the day. A more accurate approach to calculating carbon footprint would therefore be to use the real-time carbon intensity of electricity. The energy sources used in the local energy grid are called the energy mix. This package assumes that the carbon intensity of the energy source is the same regardless of where in the world the energy is produced. For example, the carbon intensity of coal is believed to be comparable in Japan and Germany. All renewable energy is assigned a carbon intensity of 0.

Power Consumed (E) is measured as kWh and is obtained by tracking power supply to the hardware at frequent time intervals. The default is every 15 seconds but can be configured with the measure_power_secs constructor argument.

ADVANTAGES OF SUSTAINABLE GEN AI

GEN AI CAN HELP IN CALCULATING SCOPE3 EMISSIONS

For small businesses, it may be a best practice to manually research product descriptions and vendors to categorize purchases. But what about companies with thousands of vendors and hundreds of thousands of purchases? Unless you can adopt an approach that combines the power of machine learning with speed and accuracy, you'll have to repeat every year.

One of the basic steps in combining per-purchase emissions data with per-vendor emissions data is to use NLP (Natural Language Processing) models to classify this type of data into different types of businesses. is.

As longtime innovators of this technology, we have developed our own internal model to accomplish this task. However, some labeled data is still required, which also requires human effort and time. With the release of the OpenAI API, you can now perform this task seamlessly. Before the release of GPT-4, GPT-3 was one of the largest models available, with a capacity of 175 billion parameters. These Large Language Models (LLMs) are trained on vast data corpora from various sources such as books, articles, websites, and other text-based resources.

This education provides his LLM with a broad knowledge base across a variety of subjects including science, technology, politics, history and culture. Therefore, LLM has the ability to classify purchased goods and services using industry-standard classification system codes such as NACE, SITC, NAICS, and HS. If you want state-of-the-art performance when classifying purchased goods and services, all you need, apart from your data of course, is the latest OpenAI text-davinci-003 update, a few parameter tweaks, and fast engineering. is.

GPT-3, especially the variant based on the text-davinci-003 update, is one of the most advanced and powerful models for natural language processing classification tasks. So it's perfect if you're looking for an accurate and fast way to categorize your Scope 3 reports. NTT DATA uses OpenAI on Azure, which provides strong data protection guarantees.

III. CONCLUSION

Sustainable Gen AI can revolutionize the world of AI. It allows you to complete tasks quickly and is applicable to almost any industry use case. At the same time, as we all strive to make the world a more sustainable place for future generations, Gen AI can offer great value.

REFERENCES

[1]. How Big is the CO2 Footprint of AI Models? ChatGPT's Emissions (carboncredits.com)
[2]. How to Streamline Scope 3 Emissions Reporting using OpenAI | NTT DATA
[3]. https://community.nasscom.in/communities/digital-transformation/sustainable-generative-ai
[4]. https://thegradient.pub/sustainable-ai/
[5]. How Generative AI can build an organization’s ESG roadmap (ey.com)