Estimating Carbon Emissions of HuggingFace AI Models

Thijs Nulle, Harmen Kroon & Petter Reijalt

March 2024

1 Introduction

The exponential growth of machine learning (ML) models has brought unprecedented potential and sophistication, revolutionising various industries. However, the surge in model complexity comes with a significant environmental cost, as evidenced by the escalating carbon emissions associated with training these models.

Machine learning, a branch of Artificial Intelligence, involves training computers to perform tasks based on data, without needing constant human intervention [8]. These models learn from a dataset and are then tested on a separate set. This process often involves multiple trials and adjustments to various settings, called hyperparameters. These adjustments lead to increased power consumption and, consequently, a greater environmental impact. The environmental cost is influenced by three factors: the cost of training the model on a single data point, the size of the training dataset, and the number of experiments performed by adjusting hyperparameters, with the total environmental cost increasing linearly with each of these factors [7].

The latest machine learning models require ever-growing amounts of parameters and power during training. For example, GPT-2, with its 1.5 billion parameters, reportedly costs around $50,000 to train. Even more extravagant, Google's leading large-language model is estimated to have used 540 billion parameters and cost over $8 million to train.

This surge in power consumption translates to a rise in carbon emissions. The BLOOM model is estimated to have produced as much CO_2 as a single passenger taking 25 round-trip flights between San Francisco and New York. Similarly, GPT-3 is estimated to have generated as much as 502 tonnes of CO_2 equivalent emissions [6].

This constant increase in power consumption and model size is an example of so-called red AI, in which the main focus lies on accuracy, no matter the cost. It is, however, not entirely clear what the benefits are of optimising this one-dimensional metric. Namely, in order for linear increases in accuracy to take place exponentially larger models are needed. The focus on this single metric is to the detriment of environmental and economic costs. This results in an environment in which state-of-the-art becomes the domain of big corporations who are the only ones to have the resources to train such large models [7].

To foster a more sustainable environment for AI research, research should strive for efficiency over accuracy. One of the metrics that embodies efficiency is the total carbon emission emitted during the training phase of an AI model [7]. However, in the current state of open-source AI models, there has yet to be a general consensus on reporting carbon emission data. For example, roughly 99% of all models on HuggingFace do not report emissions, and from the 1% that do, more than 90% of models automatically reported the statistics as they used HuggingFace AutoTrain [3].

As a result, our investigation aimed to assess whether carbon emissions could be reliably estimated using readily available information such as dataset size or model size. This would facilitate researchers in predicting and disseminating carbon emission data. By advocating for a shift towards prioritising efficiency, we aimed to foster a more equitable comparison between AI models while simultaneously mitigating financial and environmental costs. We developed a browser extension designed to be utilised before training an AI model, allowing researchers to make informed decisions regarding the environmental impact of their work.

Our key findings underscore significant disparities in reported emissions between self-reported and Autotrain models on the HuggingFace platform, highlighting the imperative for standardised reporting measures within the AI community. Furthermore, we identify a relationship between model performance metrics and carbon emissions, shedding light on the trade-offs between model efficiency and accuracy in AI development. As a main contribution, we introduce a novel approach to estimating carbon emissions for AI models through a Firefox extension, providing developers with valuable insights to make informed decisions regarding the environmental sustainability of their work.

2 Background

2.1 HuggingFace

HuggingFace is a platform on which AI models are distributed. On HuggingFace a variety of machine learning models can be downloaded. From state-of-the-art large language models from Meta to user-created machine learning models.

In the description of a model on HuggingFace, it is possible to report on the environmental impact. This can be used to report CO_2 equivalent emissions and other factors impacting the latter such as geographical location, GPU, training size etc.

HuggingFace has a feature, called AutoTrain, which lets the user train their models on the HuggingFace servers. This always comes with an automatic report on the CO_2 equivalent emissions. Whereas as for other ML models providing such information is optional.

2.2 Carbon Dioxide Equivalent

Instead of talking about power consumption, most machine learning models report their carbon dioxide equivalent (CO_2-eq). Whilst power consumption can have a different impact on the planet, depending on numerous factors, e.g. in which country the electricity was generated, the carbon dioxide equivalent tries to give an estimation of the exact impact on global warming.

The concept of CO_2-eq involves a weighted average of all greenhouse gases emitted during a process relative to their impact on global warming, as determined by their 100-year global warming potential (100-GWP). CO_2-eq is calculated by summing the products of each gas's GWP and the total mass emitted, as depicted in Equation 1.

$$\mathrm{CO_2eq} = \sum g \in \mathrm{GHG}(\mathrm{GWP}_g \cdot m_g) \tag{1}$$

where GHG represents the set of all greenhouse gases, GWP_g is the GWP for a gas g, and m_g is the total mass of gas g emitted [4]. Table 1 provides a list of notable greenhouse gases and their corresponding 100-GWP values.

Greenhouse Gas	**100-GWP**
Carbon dioxide (CO_2)	1
Methane (CH_4)	21
Nitrous oxide (N_2O)	310
Sulphur hexafluoride (SF_6)	23900

Table 1: 100-year global warming potential [4]

2.3 HuggingFace Carbon Estimation

The HuggingFace Autotrain feature incorporates Code Carbon [1] to estimate CO_2-eq emissions. Code Carbon assesses carbon emissions in real-time rather than post-training, considering two factors: hardware energy consumption (kWh) and the carbon intensity of electricity in the region (CO_2-eq/kWh).

The carbon intensity of electricity can differ from region to region. Depending on how the electricity was generated. In table 2 the carbon intensity is shown for numerous energy sources. Carbon Code uses this when the Carbon Intensity of a cloud provider or country is not known upfront [2].

Energy Source	Carbon Intensity (g/kWh)
Coal	995
Petroleum	816
Natural Gas	743
Solar	48
Geothermal	38
Hydroelectricity	26
Nuclear	29
Wind	26

Table 2: Carbon intensity per energy source [2]

3 Methodology

In this section, we describe the approach taken to investigate carbon emissions of HuggingFace AI models and the subsequent development of a model for estimating emissions in cases where the information is not present. Covering the data collection a subsequent exploratory analysis, followed by the model development we aim to provide insights into the process of assessing and mitigating the environmental impact of AI technologies.

3.1 Dataset Selection

As the basis of our predictive model, we utilise a dataset created in a repository mining study analysing the measurements of 1417 HuggingFace AI models by Castaño et al.[3] The dataset consists of the size of the dataset in bytes, CO_2 equivalent emissions in grams emitted and performance scores (e.g. accuracy, F1 score, ROUGE-1 and ROUGE-L).

3.2 Exploratory Data Analysis

To gain a comprehensive understanding of the dataset and its implications for estimating carbon emissions of HuggingFace AI models, we conducted an exploratory data analysis (EDA) encompassing several key aspects.

Data Overview

With a total of 170k entries of AI models, only 1417 models reported carbon emissions. From those models, 1301 models reported carbon emissions and at least one other metric of their model; 111 models self-reported and 1190 were trained with HuggingFace Autotrain. Other metrics include dataset size, CO_2 equivalent emissions, geographical location of the training, the domain (e.g. NLP or Computer Vision) and the size of the output model in bytes.

Utilising this information, we could plot a scatter plot with on the X-axis the model size in bytes and the Y-axis containing the CO_2 equivalent emissions, as shown in Figure 1. With models being grouped together in the center, we analysed the outliers and found that they were all models trained without Autotrain.

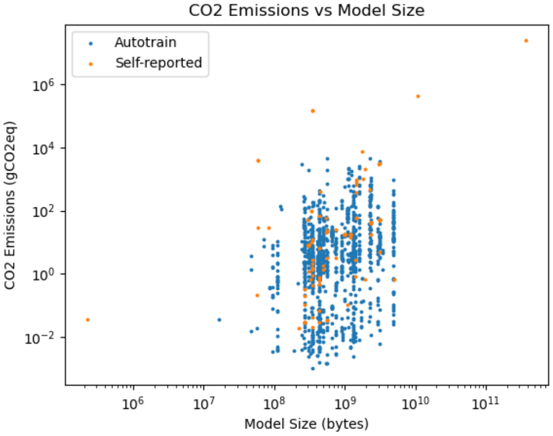


Figure 1: Distribution of CO_2 Emissions vs. Dataset Size

To compare between Autotrained and self-reported AI models, we compared the models CO_2 equivalent emissions per dataset size in bytes to see the difference in their efficiencies. Notably, we found that the first 150 most efficient models were all using Autotrain, and the 6 least efficient models were self-reported. To determine if a statistically significant difference exists between the set of models trained with Autotrain and self-reported models, we utilise different methods to assess this.

Firstly, we need to perform a test to see if the datasets are normally distributed, for which we use the Shapiro-Wilk test, with the values $P_{auto} = 7.306 \times 10^{-16}$ and $P_{non-auto} = 8.422 \times 10^{-6}$. Values below 0.05 are not normally distributed, thus, we cannot utilise a T-test to determine a statistically significant difference. We opt to use a Mann-Whitney U test, which is a non-parametric test suitable for comparing the distributions of two independent samples when normality is violated. With a P-value of $P = 1.311 \times 10^{-6}$, it indicates statistically significant evidence to reject the null hypothesis, suggesting a difference between the two groups' distributions.

Subsequently, we explored the number of models reporting additional information on their training metrics; the result is shown in table 3. The dataset is sparsely populated with nearly all data entries reporting domain and model size, and roughly half reporting accuracy and F1 score performance metrics. Table 3 further illustrates that most HuggingFace models do not report all metrics and that only a handful of models report all six metrics alongside carbon emissions.

Metric	Times Present
Carbon Emissions	1417
Domain	1362
Model Size	1306
Accuracy	845
F1 Score	775
ROUGE-1	231
ROUGE-L	228
Geographical Location	75
Dataset Size	65

Table 3: Number of times metric is present in dataset

3.3 Feature Selection

Firstly, we determine based on the Mann-Whitney U test that since a statistically significant difference exists between the carbon emissions of models trained with Autotrain and self-reported models, we opt to only estimate the carbon emissions for models trained with Autotrain. Due to the higher availability of Autotrain models and less absolute variance, we deem it a higher chance of succeeding compared to the full set of models.

Based on the data overview and insights gathered from the exploratory analysis, we have identified several metrics that are potentially relevant for predicting carbon emissions of HuggingFace AI models. These metrics include dataset size, CO_2 equivalent emissions, domain (e.g., NLP or Computer Vision), size of the output model in bytes, accuracy, F1 score, ROUGE-1, and ROUGE-L.

To prepare the dataset for analysis, we include all available metrics and adjust how we handle categorical variables. Specifically, we use a technique called one-hot encoding for *domain*. This method transforms categorical variables into binary columns, representing each category separately [5]. By doing this, we can include these variables in our model without introducing any biases from how we assign numerical values to them.

We opt not to use the geographical location of training as a feature due to its limited presence within the dataset, and of these locations a substantial amount is unique. The utilisation of one-hot encoding might result in a feature that offers little informative value for the intended outputs.

3.4 Model Development

Initially, we performed data preprocessing to address missing values, scale numerical features, and encode categorical variables utilising one-hot encoding. Following data preprocessing, the dataset was partitioned into training and testing sets. We partitioned the dataset into an 80/20 split for training and validation respectively.

Subsequently, we trained a linear regression model using the training dataset. The model sought to establish a linear relationship between a single independent variable (predictor) and the dependent variable (carbon emissions). Post-training, the model underwent rigorous evaluation using the testing dataset. Finally, coefficients obtained from the linear regression model were interpreted to elucidate the relationship between predictor variables and carbon emissions.

4 Results

4.1 Model Evaluation

In this section, we employ a set of criteria to evaluate the quality and effectiveness of the model's outputs, ensuring a thorough assessment of its performance. Due to the task at hand being a regression task, we utilise root-mean-squared error to assess the accuracy of our model's predictions.

Root Mean-squared error (RMSE) measures the square root of the average squared differences between the predicted and actual values. RMSE penalises larger errors more than mean-absolute errors and is useful for understanding the spread of errors. It provides a more interpretable measure of error compared to MSE. With RMSE ≈ 100, we can interpret that, on average, our model's predictions deviate from the actual values by approximately 100 units. This implies that the model's performance might be better or worse depending on the scale and context of the predicted variable.

In Table 4, we present the weights assigned to each feature in the linear regression model. These weights indicate the magnitude and direction of the influence that each feature has on the model's predictions. Positive weights signify a positive correlation with the target variable, meaning that an increase in the feature's value is associated with an increase in the predicted outcome. Conversely, negative weights indicate a negative correlation, implying that higher values of the feature correspond to lower predicted outcomes.

Feature	**Weight**
ROUGE-1	2326.1913
NLP (Domain)	73.0034
Computer Vision (Domain)	43.4408
Dataset Size	2.6005×10^{-7}
Model Size	-1.3961×10^{-8}
F1 Score	-10.7843
Accuracy	-109.3057
Not Specified (Domain)	-116.4442
ROUGE-L	-2242.8073

Table 4: Weights for the individual features.

5 Firefox Extension

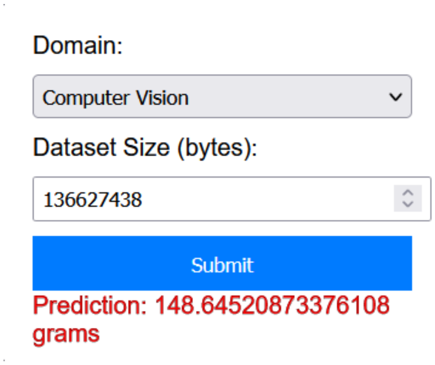
As mentioned in Section 1, the transparency of carbon emissions of AI models is lacking and needs improving. To facilitate conscious decisions by developers we created a Firefox extension that gives insights into the environmental cost of training models with HuggingFace Autotrain, based on the domain and the dataset size.

Functionality

With the singular goal of estimating carbon emissions for the training phase of an AI model, the functionality is relatively simple; fill in the pre-determined details about the AI model and estimate the carbon emissions. Currently, the user can only fill in the domain (NLP, Computer Vision or Other) and the dataset size in bytes.

Usage

When the user clicks the submit button, the information is transmitted through an HTTP request to a server, which feeds the subsequent data into the linear regression model. Based on the weights in Table 4, the model calculates the estimated CO_2 equivalent emissions in grams and returns this to the extension.



Domain: Computer Vision

Dataset Size (bytes): 136627438

Prediction: 148.64520873376108 grams

Figure 2: Firefox extension showing predicted emissions for training a Computer Vision model with a dataset size of roughly 130 MB

Upon receiving the response, the extension renders the prediction, as seen in Figure 2. The text displays different colours — green, yellow, orange or red — based on which quartile the estimated carbon emissions are in. Respectively, the first quartile is green, the second quartile is yellow, the third quality is orange, and the fourth quartile is red.

Conclusion

In conclusion, the development of the Firefox extension represents a significant step towards addressing the lack of transparency regarding carbon emissions in AI model training. By providing developers with a simple yet effective tool to estimate the environmental impact of their training processes, we aim to foster more informed decision-making.

The extension simplifies carbon emissions estimation for AI models by integrating with a linear regression model, offering insights based on domain and dataset size. Its intuitive interface empowers developers to prioritise environmental sustainability, visually representing emissions predictions.

6 Discussion

In this section, we explore the gap between emissions of Autotrain and self-reported models, attributing it to varied training methods or potentially lenient emission calculations. We also assess the impact of different features on carbon emissions, emphasising the prominence of performance metrics and the necessity for a consensus on reporting carbon emission data.

Difference Between Autotrain and Self-reported Emissions

During the exploratory data analysis, we found that a substantial difference exists between the carbon emissions of models trained with Autotrain and self-reported carbon emissions. To explain this phenomenon, we propose two different answers; different types of training or Autotrain could report too lenient carbon emission data.

Firstly, the difference could lie within the type of training, as there is a substantial difference in computational power needed for (pre-)training compared to fine-tuning of an already existing model. Within the dataset there was a column containing training-type information, however, only a handful of models actually reported on this information, thus we could not conclude anything from this.

Secondly, the carbon emission data that Autotrain publishes could be too lenient compared to the original figure. Based on the complexity of the Autotrain functionalities, the carbon emissions are most likely calculated by some estimation function. However, we cannot determine what this is comprised of and a more individual, in-depth analysis on a per-model basis is needed.

Indicators of Carbon Emissions

Based on the weights of the linear regression model in Table 4, it seems the most important features to determine carbon emissions are performance metrics. This seems logical as increasing the performance of a model normally means increasing training time, and thus increasing energy consumption. Notably, dataset size barely influences the carbon emissions and increasing the output model size actually decreases carbon emissions, which warrants further investigation as these metrics are one of the only metrics able to be gathered before starting training.

Opposite Effects of ROUGE-1 and ROUGE-L

ROUGE-1 and ROUGE-L are both performance metrics, where ROUGE-1 measures the overlap of unigrams between machine-generated and reference summaries, while ROUGE-L considers the longest common subsequence. Both these performance metrics are domain-specific and only warrant usage within the NLP domain. Even though the weights of both these parameters in absolute value are high, they can only assess carbon emissions for a subset of NLP tasks, primarily summarisation, as they are tailored specifically for evaluating the quality of generated summaries against reference summaries.

Approximately 20% of all models featured ROUGE-1 and ROUGE-L performance metrics, rendering them irrelevant for the remaining 80% of estimations. Consequently, we posit that these metrics may have been overfitted on the subset of models possessing them, serving to reduce the overall error of the model without enhancing its generalisability.

Quality of Reported Emissions Data

The reporting of carbon emissions is already low, but the extra information that is reported for each model that does report carbon emissions is even worse, with some statistics, such as dataset size, only a handful reporting it. All parties - model developers, the HuggingFace platform and the Autotrain developers - should be pushed to more rigorously report on model attributes. A tool such as Autotrain is the ideal avenue to streamline reporting since the entire configuration is done through it.

Furthermore, HuggingFace itself does not provide consistent APIs across languages which forces researchers to use the Python API for fetching model and dataset attributes and the JavaScript API for fetching model cards. There is not a single source of data that can be retrieved for each model because attributes are either written in plain text, added as model labels or inferred from file sizes.

Thirdly, model developers that self-report emissions data should do so preferably through the aforementioned model labels instead of just plain text. These labels can then be configured using consistent units of measurement such that, for example, there is no ambiguity on whether the emissions are reported in grams or kilograms.

Choice of Metric for Reporting Emissions

In the race to a zero carbon world, the focus on emission values is understandable and above all explainable to policymakers and management. However, we argue that relying solely on CO_2 equivalent emissions to compare the efficiency of models may oversimplify the environmental impact analysis. This is due to the regional and temporal fluctuations of emission cost per energy spent, whereas direct energy consumption in kWh provides an absolute value that is separate from the energy source used.

While CO_2 equivalent emissions offer a comprehensive measure of environmental impact, incorporating direct energy consumption metrics can provide additional insights into the efficiency of machine learning models. By considering both CO_2 equivalent emissions and direct energy consumption, stakeholders can better understand the environmental footprint of AI models across different contexts and make informed decisions regarding sustainability.

Realistic reporting necessitates the wider adoption and user-friendly implementation of tools such as Code Carbon [2] by developers to assess environmental impact. In an ideal scenario, both CO_2-eq and energy usage in kWh would be reported, as each serves a distinct purpose in assessing the sustainability of AI models.

6.1 Future Work

Research in the area of analysing carbon emissions produced by machine learning models can be majorly improved from two perspectives; dataset quality and fine-tuned modelling.

Improving Dataset Quality

To be able to estimate carbon emissions for models that do not report carbon emissions, we need a higher quality dataset of AI models that have carbon emissions and information about variables the carbon emissions might depend on. A significant missing data entry is the number of parameters of a model. The parameter amount is a big indicator of model size and to a certain extent the produced emissions. It is barely found - [report count here] - in the dataset even though it is a major design decision. Model parameters could improve the precision of carbon emission prediction models such as the one presented in this report.

Developing Models Independent of Performance Metrics

Further research should aim to create a prediction model that does not depend on performance metrics, and solely on information that is available before training the model. This requires better reporting and more reliant extraction of the model attributes from HuggingFace. A more well-trained prediction model enables more careful consideration of model attributes and training parameters. Such models can also be presented as more explainable as their coefficients represent known design factors of AI models.

7 Conclusion

The work on estimating carbon emissions of HuggingFace AI models sheds light on the evermore pressing issues of environmental sustainability in the field of artificial intelligence. We analysed a dataset of carbon emissions of AI models on HuggingFace and uncovered insights into the factors that influence carbon emissions during the training phase.

The investigation revealed a statistically significant difference between self-reported and Autotrain models, highlighting a discrepancy in reported carbon emissions within the dataset. This disparity shows the necessity for standardised reporting measures and enhanced transparency within the AI community to asses the actual environmental impact the field has.

The stark difference in carbon emissions between self-reported and Autotrain models emphasises the urgent need for a standardised reporting measure for AI models. Establishing consistent reporting standards that include statistics regarding the models, including training type, geographical location and hardware used, will facilitate more reliable comparisons of environmental impact, ultimately driving the field towards more sustainable AI practices.

The correlation observed between performance metrics and estimated carbon emissions suggests a delicate balance between model efficiency and accuracy in AI development. As developers partake in an arms race to optimise model performance, careful consideration is warranted about the environmental consequences of increased computation demands. Balancing efficiency and accuracy will be crucial in mitigating the environmental impact of future AI technologies.

Future research endeavours should prioritise improving dataset quality, developing models independent of performance metrics, and advocating for consistent reporting of emissions across the AI community. By addressing these key areas, researchers can advance our understanding of the environmental impact of AI models and work towards more sustainable and ethical practices in AI development.

8 Code Acknowledgments

All the code used in this project can be found in our GitHub repository at https://github.com/thijsnulle/sse-project2/ with a concise README document that elaborates further on steps for reproducing our results and running the extension.

References

[1] Displaying carbon emissions for your model. *HuggingFace.com.*

[2] Methodology - codecarbon 2.3.4 documentation. *HuggingFace.com.*

[3] Joel Castaño, Silverio Martínez-Fernández, Xavier Franch, and Justus Bogner. Exploring the carbon footprint of hugging face's ml models: A repository mining study. In *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–12, 2023.

[4] Luís Cruz and Philippe de Bekker. All you need to know about energy metrics in software engineering, May 2023.

[5] David Harris and Sarah Harris. *Digital design and computer architecture.* Morgan Kaufmann, 2010.

[6] Nestor Maslej, Loredana Fattorini, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Helen Ngo, Juan Carlos Niebles, Vanessa Parli, Yoav Shoham, Russell Wald, Jack Clark, and Raymond Perrault. Artificial intelligence index report 2023, 2023.

[7] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *CoRR*, abs/1907.10597, 2019.

[8] Mohammad Wazid, Ashok Kumar Das, Vinay Chamola, and Youngho Park. Uniting cyber security and machine learning: Advantages, challenges and future research. *ICT Express*, 8(3):313–321, 2022.