Universiteit Leiden
The Netherlands

Predicting CO2 Emissions in Machine Learning Using Hardware, Region, and Model Hyperparameters

Greijn, Tana

Citation

Greijn, T. (2026). *Predicting CO2 Emissions in Machine Learning Using Hardware, Region, and Model Hyperparameters*.

Version:	Not Applicable (or Unknown)
License:	License to inclusion and publication of a Bachelor or Master Thesis, 2023
Downloaded from:	https://hdl.handle.net/1887/4290801

Note: To cite this publication please use the final published version (if applicable).

Predicting CO_2 Emissions in Machine Learning Using Hardware, Region, and Model Hyperparameters

Tana Greijn

Thesis advisor: Prof.dr. M.J de Rooij
Thesis advisor: Quintijn Broshuis

Defended on 4 February, 2026

MASTER THESIS

STATISTICS AND DATA SCIENCE
UNIVERSITEIT LEIDEN

Contents

1 Abstract **3**

2 Introduction **4**

3 Methodology **7**

3.1 Emissions Tracking and Measurement 7

3.2 Models 7

3.2.1 Logistic Regression 7

3.2.2 DistilBERT 8

3.3 Datasets 8

3.3.1 Dataset selection and descriptions 8

3.3.2 Data Preprocessing 9

3.4 Experimental Design 10

3.4.1 Experimental factors 10

3.4.2 Model configuration 12

3.4.3 Experimental procedure 13

3.5 Evaluation Metrics 14

3.6 Statistical Methods 14

3.6.1 Emissions Prediction Analysis (Research Question 1) 14

3.6.2 Emissions-Performance Trade-off Analysis (Research Question 2) 15

4 Results **16**

4.1 Descriptive Overview: Emissions and Performance Metrics 16

4.1.1 Logistic Regression 16

4.1.2 DistilBERT 17

4.2 Research Question 1: Predicting Emissions from Experimental Factors 18

4.2.1 Logistic Regression Classifier 18

4.2.2 DistilBERT Classifier 21

4.2.3 Comparison Across Model Architectures 23

4.3 Research Question 2: Accuracy-Emissions Trade-off 24

5 Discussion **25**

6 Conclusion **29**

1 Abstract

The rapid growth of artificial intelligence (AI) generates significant greenhouse gas emissions. Prior research has primarily reported associated emissions retrospectively, with recent work introducing predictive methods to anticipate the environmental impact of machine learning (ML) models before training. This thesis extends these approaches by applying them to real-world classification datasets and comparing emissions across fundamentally different model architectures, while integrating model performance into the analysis. Emissions generated during the training and evaluation of logistic regression and distilBERT (a transformer-based model) were measured across varying hardware configurations, geographic regions, and hyperparameter settings using three text classification datasets. Linear mixed-effects models were employed to predict CO_2eq emissions from these factors, and Pearson's correlation coefficient was used to evaluate the relationship between emissions and model performance. Findings reveal that geographic region exerts the strongest influence on emissions, across both model architectures. Hardware effects are less pronounced and vary between model architectures, while hyperparameter impacts are least influential and inconsistent across models. No correlation was observed between emissions and model performance, suggesting that reducing model emissions does not necessarily compromise the performance. These findings demonstrate that ML emissions can be reliably predicted based on various experimental factors, providing the insights necessary for practitioners to make informed decisions during model selection and design.

2 Introduction

Artificial intelligence (AI) has experienced rapid growth in recent years, driven by significant advances in computing power and data availability (Van Wynsberghe, 2021). However, this proliferation raises growing concerns about the sustainability of AI systems across their entire life-cycle, from hardware manufacturing, to model training, and eventual disposal (Joshua et al., 2025). The training of state-of-the-art machine learning (ML) models alone is highly resource-intensive and consumes vast amounts of energy, leading to the production of substantial greenhouse gas emissions (Joshua et al., 2025). Even though AI has demonstrated to be a powerful facilitator of sustainable practices, such as optimizing energy management or enhancing green construction (Fan et al., 2023), these growing environmental risks present a double-edged sword. A proposed step toward addressing these challenges is the systematic accounting of energy consumption and emissions in ML research, which can improve transparency, raise awareness and help drive mitigation efforts (Henderson et al., 2020; Joshua et al., 2025).

Addressing this motivation, various research initiatives have emerged in recent years that quantify the environmental impact of ML algorithms, often using hardware energy consumption measurements or open-source emission trackers. Through approaches like these, several influential factors have been identified that affect the sustainability of model training. For instance, the choice of algorithm influences sustainability substantially: Islam et al. (2023) demonstrated that different classifier models consume vastly different amounts of energy, highlighting that conscious model selection can reduce the carbon footprint by 10 times or more. In terms of model infrastructure, research has found that hardware configuration also plays a crucial role. Lacoste et al. (2019) concluded that CPUs can can be up to 10 times less efficient than GPUs due to differences in parallel processing capabilities. Building on this, Li et al. (2016) examined various hardware configurations for training convolutional neural networks, demonstrating that strategic hardware selection can substantially reduce energy requirements while also identifying batch size and its interaction with hardware type as key factors in determining consumption. As such, hyperparemeter optimization can further reduce emissions, with other research noting that energy savings can be maximized through optimal settings of batch size and learning rate (Geißler et al., 2024). Moreover, geographic location significantly affects emissions profiles, as electricity consumption translates to different carbon footprints depending on local energy sources. Lacoste et al. (2019) reported that compute region can affect algorithm emissions by a factor of 40, with dramatic differences between locations using renewable energy versus those relying on fossil fuels. This variability suggests that emissions can, at least in part, be managed through informed choices regarding algorithm, hardware, hyperparameters, and infrastructure.

While these studies contribute to enriching AI sustainability reporting, some limitations constrain their applicability to broader scenarios. For instance, Li et al. (2016) focused exclusively on convolutional neural networks (CNN's) for computer vision, while Islam et al. (2023) evaluated only

classical algorithms (e.g. decision trees, support vector machines), limiting the generalizability of their findings to the more complex deep learning models that dominate today's sustainability discussion. Second, these studies typically examine factors in isolation rather than their combined effects. While Li et al. (2016) demonstrated interactions between batch size and hardware type, and Geißler et al. (2024) optimized batch size and learning rate jointly, comprehensive assessment of how multiple factors (hardware, region, hyperparameters, architecture) determine emissions remains limited. Finally, research generally provides retrospective measurements of emissions after training, rather than making a pre-emptive evaluation of environmental impact. While valuable for documentation and benchmarking, this approach requires that models be fully trained before their environmental impact can be assessed. Hrib et al. (2024) note that taking a predictive approach is important to facilitate informed choices during the model selection phase, which typically involves trial and error across numerous models and configurations, requiring substantial computational resources. They propose that predictive models trained on prior experimental data could enable estimation of emissions across different configurations without requiring exhaustive empirical evaluation, supporting more sustainable choices during model selection.

One approach to addressing these limitations involves shifting from post hoc analyses to a predictive service. Hrib et al. (2024) introduced a methodology for predicting CO_2 emissions associated with ML processes before training, thereby enabling more sustainable choices during model design and selection. In their study, the environmental impact of executing different ML models was documented under various experimental conditions. These data were then used to train a deep ML model that predicted emission metrics. While this approach offered a systematic method to forecast AI sustainability, it was limited in scope: the analysis focused on small classical ML models, relied on synthetic datasets, and did not incorporate model accuracy into the evaluation. These constraints limit the applicability to real-world circumstances and large-scale ML applications often used today, as well as scenarios where predictive accuracy must be weighed against environmental impact.

This final limitation points to a broader challenge in current AI sustainability research, namely, understanding the trade-off between environmental impact and model performance. While predictive accuracy has traditionally been the primary metric of success, there is growing recognition that sustainability must be considered alongside performance (Ariyanti et al., 2025). The nature of this relationship, whether higher accuracy consistently requires greater emissions, and to what extent, remains underexplored in the literature (Ben chaaben et al., 2025).

This thesis addresses these gaps by extending the predictive approach introduced by Hrib et al. (2024) to real-world text classification tasks. Specifically, it compares emissions profiles and predictive accuracy across two fundamentally different architectures: logistic regression (a clas-

sical ML algorithm) and distilBERT (a transformer-based model). By examining these models across three text classification datasets under varying hardware configurations, geographic regions, and hyperparameter settings, this study provides insights into both the predictability of AI emissions and the practical trade-offs between environmental sustainability and model performance in applied contexts.

Accordingly, the following research questions are addressed:

- **Research Question 1:** To what extent can CO_2 emissions be predicted from factors such as hardware type, compute region, and hyperparameter settings?
- **Research Question 2:** What is the relationship between model sustainability and model accuracy?

To cover these questions, this thesis proceeds as follows. The methodology section details the emission tracking procedures, as well as the three text classification tasks employed. Moreover, the experimental design is presented, including the systematic variation of hardware configurations, compute regions, and hyperparameters across both logistic regression and distilBERT models. The results section first presents the predictive modeling analysis, examining which factors most strongly determine CO_2 emissions, and quantifying their relative contributions. Subsequently, the performance-emissions trade-off is explored. Finally, the discussion interprets these findings, addressing practical implications for sustainable AI development, acknowledging methodological limitations, and proposing directions for future research in this field.

3 Methodology

To better understand the impact of hardware type, geographic region, and hyperparameter settings on the sustainability of machine learning models, a framework was used in which these factors were systematically adjusted and the associated CO_2 emissions were tracked. Data were collected for two models of varying complexity, logistic regression and distilBERT, while performing text classification tasks across three datasets. Simultaneously, performance measurements were recorded to assess the trade-off between environmental sustainability and model performance. Apart from enabling a comprehensive understanding of emissions trends, these models were chosen because they can be applied to the same tasks, making their performance directly comparable. The following sections cover a more detailed description of the tracking software used, the experimental procedure, and the statistical analyses that were applied.

3.1 Emissions Tracking and Measurement

Carbon emissions released during model execution were tracked using CodeCarbon (Schmidt et al., 2021), a Python package that estimates the environmental impact associated with running computer programs. This impact is expressed as kilograms of CO_2-equivalents (CO_2eq), a standardized measure that describes the global warming potential of greenhouse gases in terms of the amount of CO_2 that would produce an equivalent effect (CodeCarbon contributors, 2023).

In the context of computing, CO_2 emissions result from the electricity consumed by the underlying hardware. These emissions are measured in kilograms of CO_2-equivalent per kilowatt-hour. CodeCarbon estimates total emissions as the product of two main factors:

- **Energy consumed:** The total electricity used by the computational infrastructure, measured in kilowatt-hours (kWh). CodeCarbon tracks CPU and GPU power draw during execution to estimate energy consumption (CodeCarbon contributors, 2023).
- **Carbon intensity of the electricity consumed for computation:** The amount of CO_2 emitted per unit of electricity (g CO_2/kWh), computed from a weighted average of the energy sources in the local energy grid (e.g., coal, gas, renewables) (CodeCarbon contributors, 2023). Carbon intensity values vary significantly by region depending on grid composition.

3.2 Models

3.2.1 Logistic Regression

Logistic regression was implemented as a single-layer neural network architecture using PyTorch (PyTorch Contributors, 2023). This neural network structure is functionally equivalent to logistic

regression: a linear transformation of input features followed by sigmoid activation to produce binary classification probabilities (Jurafsky & Martin, 2025). Relative to distilBERT's deep neural network architecture, logistic regression demonstrates minimal computational complexity, reflecting the functioning of lightweight machine learning methods in evaluating emissions. PyTorch was selected for consistency with the distilBERT implementation, allowing the same hyperparameters (batch size and epochs) to be tracked across both models. Additionally, PyTorch supports both CPU and GPU processing, enabling assessment of both hardware configurations.

3.2.2 DistilBERT

DistilBERT is a transformer-based neural network model created through knowledge distillation, a process in which a smaller 'student' model learns to reproduce the behavior of a larger 'teacher' model (Sanh et al., 2019). Specifically, distilBERT was distilled from BERT, retaining approximately 97% of BERT's language understanding capabilities while reducing model size by 40% and inference time by 60% (Sanh et al., 2019). The pre-trained distilBERT model was obtained from the Hugging Face Transformers library and fine-tuned on each dataset's training split (Hugging Face, 2024; Sanh et al., 2019). The goal of pre-training is for the model to learn basic language from a large corpus of text. In the fine-tuning stage, the model's parameters are further adapted to solve a specific task, using a smaller body of supervised training data (Prince, 2023). This approach enables the assessment of the environmental impact of modern transformer architectures while avoiding the prohibitive cost of training such models from scratch.

DistilBERT's architecture consists of 6 transformer layers with 12 attention heads and approximately 66 million parameters (Sanh et al., 2019), representing substantially greater complexity than logistic regression's single layer network. As such, these models facilitate the exploration of potential scaling effects across model complexity regarding emissions trends.

3.3 Datasets

3.3.1 Dataset selection and descriptions

Experiments were conducted on three widely used text classification datasets: SMS Spam Collection, IMDB Movie Reviews, and Yelp Polarity. These datasets were selected because of their varying sizes, reducing the risk of findings being specific to a particular data scale, thereby offering robustness. Additionally, their widespread use in machine learning research simplifies the interpretation of results. Finally, each dataset facilitates the classification of text into binary categories, enabling a fair comparison of the model performance between logistic regression and distilBERT. The datasets are publicly available and specified as follows:

- **SMS Spam Collection:** A small dataset consisting of 5,574 labeled SMS messages classified as spam or ham (not spam), commonly used in mobile phone spam research (Almeida & Hidalgo, 2011). Spam messages constitute 13.4% of samples while ham messages represent 86.6%. The dataset was divided into a 80-20 train-test split, yielding 4459 training samples and 1115 test samples.

- **IMDB Movie Review:** A medium-sized dataset containing 50,000 movie reviews labeled by sentiment (positive/negative) with balanced class distribution. It has been pre-split into a training set (25,000 labeled reviews) and a test set (25,000 examples) (Maas et al., 2011).

- **Yelp Polarity:** A large, pre-split, dataset of Yelp reviews containing 560,000 training samples and 38,000 test samples, labeled by positive or negative polarity (sentiment) (Zhang et al., 2015). A positive polarity corresponds to reviews with 3-4 stars, while negative polarity indicates 1-2 stars.

The IMDB and Yelp datasets contain equal amounts of positive and negative reviews, however, the SMS dataset exhibits a class imbalance (13.4% spam). This inconsistency could lead to inflated performance measurements from text classification, by models being biased towards the majority class (Albattah & Khan, 2025). This limitation is further explored in the Evaluation Metrics section.

Additionally, there are differing split ratios across the datasets: 80-20 for SMS, 50-50 for IMDB, and 93.6-6.4 for Yelp. The IMDB and Yelp datasets have already defined splits, which are maintained to ensure consistency with established benchmarks. For the SMS dataset, a practical standard split was chosen. In the analysis, emissions are predicted from model-specific parameters and the emissions-performance trade-off is explored within each dataset rather than across datasets. Preserving the original splits for IMDB and Yelp datasets therefore reflects standard research practice without compromising the research objectives.

3.3.2 Data Preprocessing

For both models, labels were encoded as binary values before training. Furthermore, only essential preprocessing steps were performed to assess baseline model-performance (without extensive optimization), ensuring a fair comparison between models. Preprocessing is necessary in text classification tasks because it allows the data to be transformed from unstructured text into structured numerical representations that can be processed by ML algorithms.

Logistic Regression: For all datasets, the input text was converted to lowercase. Dataset-specific procedures included the removal of URLs in the SMS dataset, and HTML tags in the IMDB and Yelp datasets. These features were removed because they do not contribute meaningful information and would introduce noise into the text representations. Subsequently, the input

texts were transformed into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, and then turned into PyTorch tensors for model compatibility.

DistilBERT: To maintain comparability with logistic regression, the same dataset-specific cleaning was applied: URLs were removed from the SMS dataset, and HTML tags were removed IMDB and Yelp datasets. Text inputs were then tokenized using the distilBERT tokenizer (distilbert-base-uncased), converting them into numerical representations. Additional lowercasing was not performed because distilBERT's tokenizer applies lowercasing prior to processing. A maximum sequence length of 128 was selected to reduce the duration and computational load of the tokenization process. Padding and truncation were applied as necessary.

3.4 Experimental Design

3.4.1 Experimental factors

The experimental factors examined in this study were selected based on the approach described by Hrib et al. (2024). Table 1 lists all factors and evaluated values.

Table 1

Experimental Factors and Evaluated Values

Experimental Factor	Evaluated Values
Epochs	2, 4, 6
Batch size	16, 32, 64
Hardware type	CPU, GPU
Compute region	Canada (Quebec), Germany, India (Mumbai)

Note. The table summarizes the experimental factors and corresponding values evaluated across all model training runs in this study.

Hyperparameters: Two hyperparameters were varied for both models:

- **Batch size**: Defines the number of training samples used to compute each gradient update during optimization. Batch optimization uses the entire dataset to compute exact gradients of the loss function, stochastic optimization uses single observations, and mini-batch optimization, employed in this study, uses subsets of the data determined by the batch size parameter (Prince, 2023). Smaller batches result in noisier gradient estimates but enable more frequent parameter updates, thereby consuming less memory. On the other hand, larger batches provide more stable gradient approximations at the cost of increased memory requirements, but are generally associated with faster training through better hardware

utilization (Ariyanti et al., 2025).

Previous research on neural networks has shown that there is a complex relationship between batch size and energy consumption (Li et al., 2016). Increasing the batch size reduces energy consumption through better hardware utilization and reduced training duration. However, once the hardware utilization saturates, increasing the batch size does not lead to further reductions in energy consumption. Moreover, beyond this saturation point, excessively large batch sizes may exceed available memory capacity, preventing training altogether. The values (16, 32, 64) were selected to capture a range that reflects low to high hardware utilization while remaining computationally feasible.

- **Number of epochs**: Controls how many complete passes the optimization algorithm makes through the entire training dataset. Each additional epoch increases the amount of computational work, with research highlighting that energy consumption increases linearly with the number of epochs (Ariyanti et al., 2025). The values (2, 4, 6) were chosen to provide a sufficient range to capture this relationship while remaining computationally feasible.

Hardware: Models were executed on both CPU- and GPU-based setups to allow emissions across computational infrastructures to be assessed. CPUs are optimized for sequential processing and general computing tasks, handling instructions one after another across a limited number of cores. In contrast, GPUs are designed for parallel computation with thousands of cores, making them well-suited to process the vast number of calculations necessary to train complex models (Gyawali, 2023). This architectural difference means that tasks involving large-scale matrix multiplications, like transformers, can be distributed across many GPU cores at the same time, whereas CPUs process these operations sequentially (Gyawali, 2023; Vaswani et al., 2017). The following hardware specifications were considered:

- **CPU**: Logistic regression experiments were conducted on a machine with an Apple M1 Pro processor. DistilBERT experiments used a machine with an Intel Xeon processor. While this hardware difference limits direct CPU comparison between models, all comparisons made within models (across regions, batch sizes, and epochs) remain valid, as they use consistent hardware.

- **GPU**: All GPU-based experiments were executed using Google Colab's online integrated development environment with an NVIDIA Tesla T4 GPU.

Geographic Region: To capture regional variation in electricity generation, experiments were conducted across three geographic regions representing low, medium, and high carbon intensity electricity grids (Ember, 2025; Our World in Data, 2025):

- Canada (CAN): Low carbon intensity due to primarily hydroelectric power generation. Experiments used Quebec region specifications (northamerica-northeast1).
- Germany (DEU): Moderate carbon intensity reflecting a balanced mix of renewable and fossil fuel sources. Country-level estimates were used (europe-west10).
- India (IND): High carbon intensity due to coal-dominated electricity generation. Experiments used Mumbai region specifications (asia-south1).

For Canada and India, CodeCarbon includes data for multiple regions. Quebec and Mumbai were selected to reflect low and high carbon intensities, respectively. CodeCarbon provides only country-level estimates for Germany, which is why no specific region was specified.

To compute emissions, CodeCarbon uses the regional carbon intensity values published by Google Cloud Platform (GCP). While all experiments were executed locally (CPU experiments) or on Google Colab (GPU experiments), GCP region codes (e.g. northamerica-northeast1, europe-west10, asia-south1) were applied with their corresponding carbon intensity values to calculate what the emissions would have been if the work had been performed in each region.

3.4.2 Model configuration

Logistic Regression Configuration: Models were trained using Stochastic Gradient Descent (SGD) with a fixed learning rate of 0.01. The loss function was Binary Cross-Entropy (BCE) with logits (BCEWithLogitsLoss). Binary cross-entropy is mathematically equivalent to the negative log-likelihood used in classical logistic regression (Raschka, 2022). BCEWithLogitsLoss combines the sigmoid activation and BCE loss calculation in a single operation, which is numerically more stable than applying them separately (PyTorch Documentation, 2025). For a binary outcome y_n and predicted probability $\hat{p}_n$, the BCE loss for observation n can be described as (PyTorch Developers, 2025):

$$\ell_n = -w_n \left[y_n \cdot \log \hat{p}_n + (1 - y_n) \cdot \log(1 - \hat{p}_n) \right], \tag{1}$$

Where w_n is an optional sample weight (used for class balancing), and $\hat{p}_n$ is obtained by applying the sigmoid function to the linear combination of input features (Jurafsky & Martin, 2025). This corresponds to the negative log-likelihood loss for the Bernoulli distribution, typically used in logistic regression.

To address class imbalance in the SMS Spam dataset, where spam messages represent only 13.4% of the samples (747 spam vs. 4827 ham), a positive class weight of 6.46 was applied, corresponding to the ratio of negative to positive samples (PyTorch Documentation, 2025). This weighting penalizes the model more heavily for incorrectly classifying the minority class. Without

class weighting, the logistic regression model achieved an F1-score of zero, indicating complete failure to detect spam. This would affect the comparability of the performance between logistic regression and distilBERT. As such, weighting ensured that performance metrics more accurately reflected the model's true classification capability rather than bias toward the majority class. No class weighting was applied to the experiments that involved IMDB and Yelp datasets, as both have a balanced class distribution.

A single-epoch warm-up run with batch size 16 was performed prior to logistic regression experiments on the SMS dataset to stabilize hardware performance measurements. This procedure was necessary because initial GPU and CPU executions produced inflated measurements, resulting in considerable outliers that skewed the data. Due to the fast execution of logistic regression on the SMS dataset (compared to distilBERT and larger datasets), the impact of hardware stabilization was disproportionately large, taking up a greater proportion of the total execution time.

DistilBERT Configuration: Fine-tuning was conducted using the AdamW optimizer with a fixed learning rate of 2×10^{-5}, following recommended practices for fine-tuning transformer models (DataCamp, 2024; Devlin et al., 2019). Cross-entropy loss was used as the loss function. For distilBERT, class weighting was not applied to the SMS dataset. Experiments applying the same 6.46 class weight to distilBERT showed negligible differences in performance metrics compared to unweighted training, suggesting distilBERT handles class imbalance effectively and does not require explicit weighting. As with logistic regression, no class weighting was applied to experiments with IMDB or Yelp datasets, due to balanced class distributions.

3.4.3 Experimental procedure

Each experimental run involved training the model on the training split and evaluating it on the held-out test split, with emissions tracked throughout the entire process. CodeCarbon was integrated into both training and testing pipelines using the OfflineEmissionsTracker class. For every run, a separate CodeCarbon tracker instance was initialized with the appropriate country ISO code, regional specifications, and cloud provider (Google Cloud Platform) before training commenced.
The complete experimental design was as follows:

- Logistic regression: 3 batch sizes $\times$ 3 epochs $\times$ 3 regions $\times$ 2 hardware types $\times$ 3 datasets = 162 runs
- DistilBERT: 3 batch sizes $\times$ 3 epochs $\times$ 3 regions $\times$ 2 hardware types $\times$ 2 datasets = 108 runs

A loop structure was implemented to iterate through all possible configurations. After each run, the emissions (kgCO_2eq) and model performance metrics were automatically recorded to .csv

files. The resulting data were aggregated into a single dataset per model for subsequent statistical analysis.

DistilBERT was evaluated only on SMS and IMDB datasets due to the substantial computational requirements of training on the Yelp dataset. Moreover, each experimental configuration (combination of batch size, epochs, region, and hardware) was executed once due to computational constraints, particularly for experiments involving the larger datasets. This limits the ability to assess variability across measurements but was necessary to remain computationally feasible.

3.5 Evaluation Metrics

Both accuracy (proportion of correctly classified samples out of all samples) and F1-score (harmonic mean of precision and recall) were recorded for all experiments (Google Developers, 2025). F1-score was emphasized for the SMS dataset due to its imbalanced class distribution. Accuracy was emphasized for the balanced datasets (IMDB and Yelp). Accuracy and F1-score were selected because they are commonly used as evaluation metrics in the literature for these respective tasks (Lorenzoni et al., 2024). Moreover, both metrics allow for a comprehensive assessment of predictive performance and facilitates analysis of the emissions-performance relationship.

3.6 Statistical Methods

3.6.1 Emissions Prediction Analysis (Research Question 1)

To evaluate the extent to which experimental factors predict emissions, separate linear mixed-effects models were fitted for logistic regression and distilBERT using the statsmodels module in Python (Seabold & Perktold, 2010). Separate models were necessary as a combined model violated normality assumptions due to the vastly different emissions profiles of the two architectures, creating bimodal residual distributions.

Model Structure: The model predicted log-transformed emissions from the four experimental factors (fixed effects), while accounting for baseline differences across the classification datasets through a random intercept. $\log_{10}$-transformation was applied to address the right-skewed distribution of emissions data. Dataset was modeled as a random intercept because the vastly different dataset sizes introduced non-independence. Additionally, interactions between each experimental factor were evaluated alongside individual effects. Statistical significance was assessed at $\alpha = 0.05$.

Model Diagnostics: Residual normality was assessed via histograms, and homoscedasticity was evaluated by plotting residuals against fitted values.

3.6.2 Emissions-Performance Trade-off Analysis (Research Question 2)

To examine the relationship between emissions and model performance, Pearson correlation coefficients were computed between log-transformed emissions and each performance metric (accuracy and F1-score) using scipy.stats.pearsonr (SciPy Documentation, 2025). Pearson correlation was selected after visual inspection of scatterplots confirmed approximately linear relationships between log-transformed emissions and performance metrics. Correlations were calculated separately for each model-dataset combination to account for dataset-specific baseline characteristics, ensuring independence of measurements. Correlation strength was based on the guide proposed by Evans (1996), defining the following effect sizes for the absolute value of r:

- $.00 - .19$ = "very weak"
- $.20 - .39$ = "weak"
- $.40 - .59$ = "moderate"
- $.60 - .79$= "strong"
- $.80 - 1.0$= "very strong"

Statistical significance was assessed at $\alpha = 0.05$.

4 Results

4.1 Descriptive Overview: Emissions and Performance Metrics

A descriptive overview of the emissions, accuracy, and F1-score recorded for the logistic regression classifier is presented in Table 2. The same information is described in Table 3 for the distilBERT classifier.

4.1.1 Logistic Regression

For experiments involving logistic regression, emissions measured in kg CO_2eq span several orders of magnitude, ranging from very small values (minimum = 5.69×10^{-10} kg) to a maximum of 5.22×10^{-2} kg. The median emission (1.23×10^{-5} kg) is much lower than the mean (2.52×10^{-3} kg), indicating a strongly right-skewed distribution. This skewness is further reflected by the fact that 75% of the runs produce emissions below 4.04×10^{-4} kg (75th percentile), while the maximum observed value is approximately 130 times larger than this upper quartile. These results suggest that a small number of high-emission configurations contribute disproportionately to total emissions.

In terms of predictive performance, logistic regression achieves a mean accuracy of 84.9% and a mean F1-score of 80.1%, with noticeable variability across runs (standard deviation is 7.2% and 9.1%, respectively). Performance measured using the F1-score is consistently lower than accuracy, which is expected given the greater sensitivity of the F1-score to class imbalance in text classification tasks.

Table 2

Descriptive Statistics for Logistic Regression Model: Emissions, Accuracy, and F1 Score

Statistic	Emissions (kg)	Log-Emissions	Accuracy	F1-Score
Count	162	162	162	162
Mean	2.515×10^{-3}	−4.889	0.849	0.801
SD	7.723×10^{-3}	1.918	0.072	0.091
Min	5.685×10^{-10}	−9.245	0.599	0.399
25th percentile	5.603×10^{-7}	−6.252	0.800	0.783
Median	1.232×10^{-5}	−4.911	0.854	0.814
75th percentile	4.040×10^{-4}	−3.395	0.879	0.860
Max	5.218×10^{-2}	−1.283	0.969	0.886

Note. $N = 162$ observations. Emissions measured in kilograms of CO_2-equivalent. Percentiles represent quartile values.

4.1.2 DistilBERT

The descriptive statistics presented in Table 3 highlight that the distilBERT emissions are substantially higher than those observed for logistic regression, with a minimum of 6.00×10^{-6} kg and a maximum of 8.91 kg. As with logistic regression, the emissions distribution is strongly right-skewed: while most runs produce relatively low emissions (75th quartile is below 3.63×10^{-2} kg), a small number of configurations result in extremely high values, with the maximum exceeding the median by more than three orders of magnitude. Notably, the median emissions for distilBERT (7.97×10^{-3} kg) are approximately 650 times higher than those of logistic regression, highlighting the substantial computational and environmental cost associated with transformer-based models compared to classical ML algorithms.

Regarding predictive performance, distilBERT achieves greater accuracy and F1-scores than logistic regression, with a shared mean value of 93.2%. In contrast to logistic regression, accuracy and F1-scores for distilBERT are closely aligned. This indicates that distilBERT handles the imbalanced dataset more effectively.

Table 3

Descriptive Statistics for DistilBERT Model: Emissions, Accuracy, and F1 Score

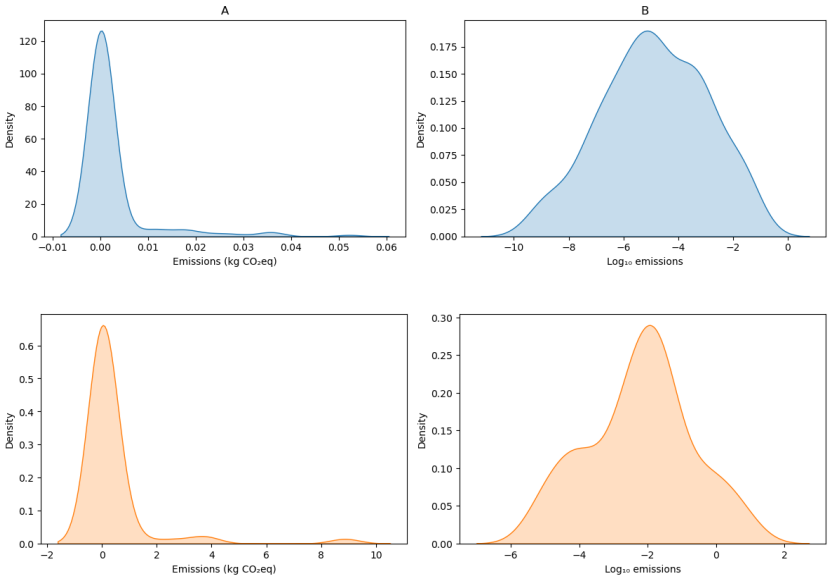
Statistic	**Emissions (kg)**	**Log-Emissions**	**Accuracy**	**F1 Score**
Count	108	108	108	108
Mean	3.880×10^{-1}	-2.226	0.932	0.932
SD	1.372	1.508	0.060	0.060
Min	6.000×10^{-6}	-5.234	0.850	0.849
25th percentile	8.910×10^{-4}	-3.064	0.872	0.872
Median	7.965×10^{-3}	-2.099	0.931	0.931
75th percentile	3.631×10^{-2}	-1.440	0.992	0.992
Max	8.910	0.950	0.996	0.996

Note. $N = 108$ observations. Emissions measured in kilograms of CO_2-equivalent. Percentiles represent quartile values.

Given the strongly right-skewed emissions distributions observed above, all subsequent analyses used the $\log_{10}$-transformed emissions as the outcome variable. Figure 1 highlights the changes in the emission distributions of each model after applying the log-transformation. These figures show that the effects of extreme outliers have been reduced, resulting in more normally distributed data.

Figure 1

Comparison of Emissions Distributions Across Models



Note. Logistic regression distributions are presented in the first row (blue) and distilBERT distributions are presented in the second row (orange). Left column (A) shows the actual emissions distributions. Right column (B) shows the $\log_{10}$-transformed emissions. Emissions were measured in kilograms of CO_2-equivalent.

4.2 Research Question 1: Predicting Emissions from Experimental Factors

Two linear mixed effects models were fitted to predict the log-transformed emissions for logistic regression and distilBERT classifiers. In both models, geographic region, hardware type, batch size and number of epochs were included as fixed effects, while dataset was modeled as a random intercept. Subsequently, model diagnostics were measured and their outputs were interpreted.

4.2.1 Logistic Regression Classifier

Mixed-Effects Model Results: Table 4 reports the fixed-effects estimates from the linear mixed-effects model for logistic regression. None of the interactions tested were statistically significant, which is why they were omitted. As far as the individual predictor effects, increasing the number of training epochs is associated with higher emissions. Relative to the baseline of two epochs, training for four and six epochs increases log-emissions by 0.260 and 0.424, respectively. On the original scale, these correspond to emissions that are 1.82 times and 2.65 times

the baseline level, representing increases of approximately 82% and 165%.

In contrast, larger batch sizes are associated with lower emissions, with batch sizes of 32 and 64 reducing log-emissions by 0.072 and 0.103, relative to the baseline batch size of 16. On the original scale, these correspond to emissions that are 0.85 times and 0.79 times the baseline level, representing reductions of approximately 15% and 21%.

Hardware choice exhibits a stronger effect. Executing the logistic regression classifier on a GPU increases log-emissions by 1.292 compared to CPU execution, corresponding to emissions that are nearly 20 times higher than the baseline CPU level.

Regional differences are the most substantial: relative to Canada (baseline), experiments conducted in Germany and India are associated with log-emission increases of 2.266 and 2.539, respectively. On the original scale, these correspond to emissions that are approximately 184 times and 346 times higher than Canada.

The model included dataset as a random intercept to account for baseline differences across the SMS, IMDB, and Yelp datasets. The estimated random-effect variance ($\sigma^2 = 2.787$) indicated substantial variation in baseline emissions across datasets, suggesting that, when setting all experimental factors to zero, the dataset characteristics contribute to emissions. However, the variance estimate should be interpreted cautiously, as it is based on only three levels (datasets).

Model Diagnostics: Model diagnostics did not indicate major violations of linear mixed-effects model assumptions. The histogram of residuals (Figure 2, left) shows an approximately normal distribution, while the plot of residuals against fitted values (Figure 2, right) does not reveal systematic patterns, suggesting homoscedasticity and appropriate model specification.

Table 4

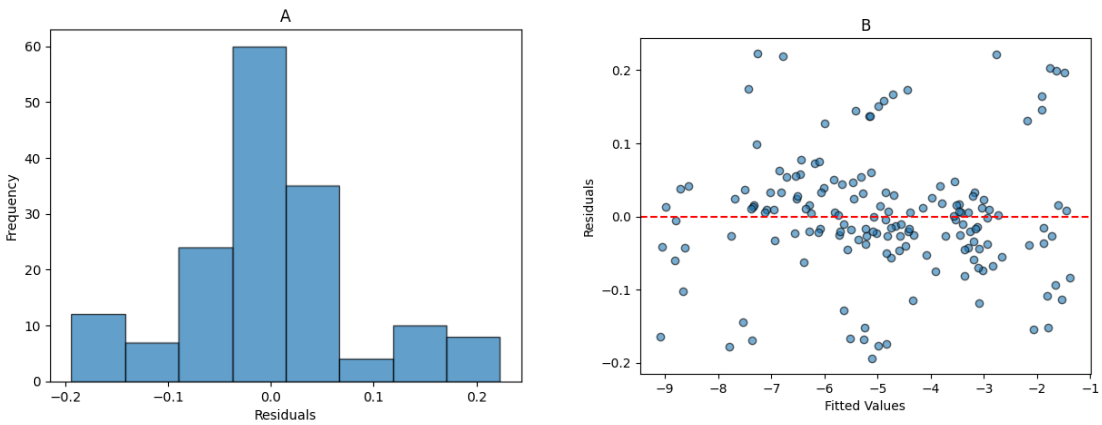
Fixed Effects Estimates From Mixed-Effects Model for Logistic Regression Classifier

Predictor	Coef.	*SE*	*p*	95% CI	
				LL	UL
Intercept	−7.297	0.964	<.001	−9.186	−5.407
Epochs (ref: 2 epochs)					
4 epochs	0.260	0.017	<.001	0.227	0.293
6 epochs	0.424	0.017	<.001	0.391	0.457
Batch Size (ref: 16)					
32	−0.072	0.017	<.001	−0.105	−0.039
64	−0.103	0.017	<.001	−0.137	−0.070
Hardware (ref: CPU)					
GPU	1.292	0.014	<.001	1.265	1.319
Country (ref: Canada)					
Germany	2.266	0.017	<.001	2.233	2.299
India	2.539	0.017	<.001	2.506	2.573

Note. The model included a random intercept for dataset (variance = 2.787). The dependent variable was $\log_{10}$-transformed emissions (kg CO_2eq). All predictors were categorical.

Figure 2

Model Diagnostics Plots for Linear-Mixed Effects Model (Logistic Regression Classifier)



Note. The left panel (A) shows a histogram of model residuals. The right panel (B) shows residuals plotted against fitted values, illustrating model fit and variance distribution.

Interpretation Overall, the results indicate that emissions associated with logistic regression are strongly influenced by both model hyperparameters, hardware choice, and compute region,

with varying effect magnitudes. Increasing the number of epochs raises emissions, while larger batch sizes demonstrate the opposite pattern. Moreover, the effect of epochs on emissions is more pronounced compared to the batch size.

An even greater contribution to emissions arises from the type of hardware that is used, with GPU computations substantially increasing the carbon footprint. The largest effect stems from the compute region, highlighting a notable increase in emissions across levels.

4.2.2 DistilBERT Classifier

Mixed-Effects Model Results: Table 5 reports the fixed-effects estimates for distilBERT. None of the interactions tested were statistically significant, which is why they were omitted. Firstly, increasing the number of training epochs is associated with higher emissions, similar to the effect in logistic regression. Relative to the baseline of two epochs, training for four and six epochs increases log-emissions by 0.246 and 0.508, respectively. On the original scale, these correspond to emissions that are approximately 1.76 times and 3.22 times the baseline level, representing increases of approximately 76% and 222%.

Batch size effects for logistic regression were conclusive, but this was not the same for distilBERT. While the estimated coefficients for batch sizes of 32 and 64 are positive (0.022 and 0.029), neither effect is statistically significant ($p > 0.05$), suggesting that emissions resulting from adjusting the batch size cannot be reliably predicted in this experimental setting.

Hardware choice has a substantial and directionally opposite effect compared to logistic regression. Executing distilBERT on a GPU is associated with significantly lower emissions than CPU execution, with a coefficient of -1.491. On the original scale, this corresponds to emissions that are approximately 0.032 times the CPU baseline, representing a 97% reduction in emissions. This inconsistency highlights the difference in how logistic regression and distilBERT interact with underlying hardware. DistilBERT's complex architecture involves extensive parallel matrix operations, maximizing GPU utilization, while logistic regression underutilizes this hardware due to its simple structure (Li et al., 2016).

Regional effects are consistent with those observed for logistic regression. Experiments conducted in Germany and India produce significantly higher emissions (coefficients of 2.238 and 2.428, respectively) than those conducted in Canada. On the original scale, emissions in Germany are approximately 173 times higher and emissions in India are about 268 times higher than in Canada.

The model included dataset as a random intercept to account for baseline differences across the two datasets used for distilBERT training (SMS and IMDB). The estimated random-effect

variance ($\sigma^2 = 0.503$) indicates moderate variation in emissions across these datasets, suggesting dataset characteristics contribute to emissions released by distilBERT, though to a lesser extent than observed for logistic regression. Importantly, the random-effect variance should be interpreted with caution because it is based on solely two datasets.

Table 5

Fixed Effects Estimates From Mixed-Effects Model for DistilBERT Classifier

Predictor	**Coef.**	*SE*	***p***	**95% CI**	
				LL	**UL**
Intercept	−3.303	0.516	<.001	−4.315	−2.291
Epochs (ref: 2 epochs)					
4 epochs	0.246	0.106	0.020	0.038	0.453
6 epochs	0.508	0.106	<.001	0.300	0.716
Batch Size (ref: 16)					
32	0.022	0.106	0.835	−0.186	0.230
64	0.029	0.106	0.786	−0.179	0.236
Hardware (ref: CPU)					
GPU	−1.491	0.014	<.001	−1.661	−1.322
Country (ref: Canada)					
Germany	2.238	0.106	<.001	2.030	2.445
India	2.428	0.106	<.001	2.220	2.636

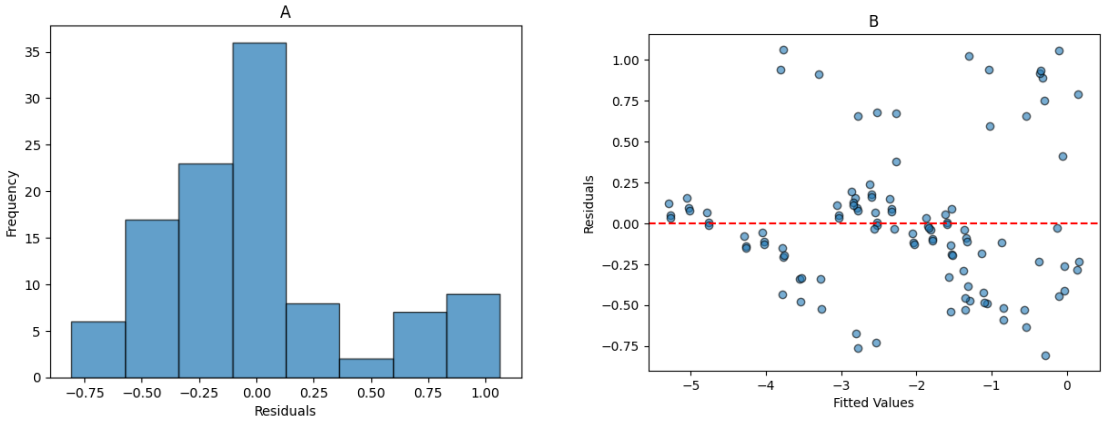
Note. The model included a random intercept for dataset (variance = 0.503). The dependent variable was $\log_{10}$-transformed emissions (kg CO_2e). All predictors were categorical.

Model Diagnostics: Diagnostic plots suggest that the assumptions of the linear mixed-effects model are reasonably satisfied. The histogram of residuals (Figure 3, left panel) indicates an approximately normal distribution, though with slight positive skew. The plot of residuals against fitted values (Figure 3, right panel) does not reveal strong systematic trends or severe heteroskedasticity.

Notably, the residuals for the distilBERT model (Figure 3A) are substantially larger than those for logistic regression (Figure 2A). This likely reflects the considerable difference in computational and architectural complexity between the two architectures. More specifically, distilBERT might exhibit a more complex relationship between the experimental factors and emissions compared to logistic regression, making its emissions harder to predict using the linear mixed effects model. Even though distilBERT was better at solving the classification tasks (higher accuracy and F1-scores), this could explain why it yielded greater residuals.

Figure 3

Model Diagnostics Plots for Linear-Mixed Effects Model (DistilBERT Classifier)



Note. The left panel (A) shows a histogram of model residuals. The right panel (B) shows residuals plotted against fitted values, illustrating model fit and variance distribution.

Interpretation: The results demonstrate that emissions from distilBERT are primarily driven by the number of epochs, hardware choice, and compute region, while batch size does not exhibit a significant effect. The smallest effect is given by the number of epochs, with emissions increasing as more epochs are introduced to model training. Unlike logistic regression, distilBERT benefits from GPU execution, which significantly reduces emissions relative to CPU-based training. The region remains the dominant predictor, consistent with the pattern observed for logistic regression.

4.2.3 Comparison Across Model Architectures

Comparing the mixed-effects results across model architectures reveals several key differences. The number of epochs generates similar proportional effects for both models, with emissions increasing about 76-82% for four epochs and 165-222% for six epochs compared to two epochs. Regional effects are also consistent in size and direction: emissions in Germany and India are approximately 173-184 and 268-346 times higher than Canada for both models.

In contrast, batch size and hardware effects differ markedly between architectures. Larger batch sizes reduce emissions by 15-21% for logistic regression but could not reliably predict emissions for distilBERT. Hardware choice exhibits opposite effects, with GPU execution increasing logistic regression emissions by roughly 20 times but reduces distilBERT emissions by 97%.

Finally, dataset-level variation is considerably larger for logistic regression ($\sigma^2 = 2.787$) than for distilBERT ($\sigma^2 = 0.503$), suggesting that dataset characteristics may play a greater role in determining emissions for logistic regression. However, as the random intercept for logistic

regression is based on three datasets and distilBERT's is based on only two, the reliability of these variance estimates is limited.

4.3 Research Question 2: Accuracy-Emissions Trade-off

To assess the relationship between emissions and model performance, Pearson correlation coefficients were calculated for each dataset and model combination (Table 6). Correlations between emissions and both accuracy and F1-score were consistently very weak, with $|r| < 0.19$. Logistic regression showed slight positive correlations ($r = 0.03$ to 0.12), while distilBERT showed slight negative correlations ($r = -0.05$ to -0.08), though neither pattern was statistically significant ($p > 0.05$). These findings indicate the lack of a measurable tradeoff between log-emissions and model performance within the tested experimental configurations.

Table 6

Pearson Correlation Coefficients Between Log-Emissions and Performance Metrics by Dataset and Model

Dataset	**Model**	**Accuracy** *(r)*	***p***	**F1-Score** *(r)*	***p***
SMS	Logistic Regression	0.100	0.471	0.120	0.386
	DistilBERT	−0.084	0.547	−0.082	0.556
IMDB	Logistic Regression	0.101	0.467	0.033	0.810
	DistilBERT	−0.049	0.724	−0.048	0.733
Yelp	Logistic Regression	0.050	0.723	0.075	0.599
	DistilBERT	—		—	

Note. All correlations were non-significant at $\alpha = 0.05$. Correlations were calculated separately for each dataset to account for differing emissions profiles across datasets.

5 Discussion

This study examined the extent to which hyperparameters, hardware type, and geographic region predict CO_2 emissions from machine learning execution, and explored the trade-off between model performance and emissions. Using logistic regression and distilBERT across three text classification datasets, the results provide insights into both research questions posed in this thesis. Regarding Research Question 1, the results demonstrate that CO_2 emissions can be reliably predicted from infrastructure-level factors, particularly region and hardware type, which yielded the highest impacts. Hyperparameter effects were considerably smaller and less consistent across model architectures. Notably, batch size did not reliably predict emissions for distilBERT, while the number of epochs showed consistent effects across both models.

Geographic region emerged as the dominant predictor of emissions across both model architectures. Experiments conducted in Germany and India produced emissions substantially higher than those in Canada, reflecting differences in regional carbon intensity of electricity grids. These findings align with Anthony et al. (2020), noting a considerable increase in emissions released by model training in a location that relies primarily on fossil fuels compared to one that uses renewable energy sources. This highlights the practical significance of considering geographic region prior to model selection. For complex models like transformers, which can require weeks of training, strategic region selection can avoid considerable quantities of CO_2eq (Lacoste et al., 2019).

Notably, even moderate differences in carbon intensity between regions translated into substantial emissions changes. For instance, comparing emissions between Germany and Canada revealed that geographic location alone could account for emissions differences exceeding 170-fold, further underscoring the substantial impact of region selection. The consistency of regional effects across both logistic regression and distilBERT suggests that geographic region is the dominant factor in emissions outcomes regardless of model complexity, making this experimental factor a particularly reliable and generalizable variable to consider for emissions reduction.

Hardware selection demonstrated model-dependent effects on emissions. For logistic regression, GPU execution increased emissions nearly 20-fold compared to CPU execution, while for distilBERT, GPU execution reduced emissions by 97%. This reversal highlights fundamental differences in how these architectures are capable of utilizing GPU parallel processing. More specifically, GPUs are designed to excel at highly parallel tasks, such as intensive AI workloads, by dividing processing tasks across thousands of specialized cores (Intel Corporation, 2025). This makes them well-suited to handle transformer architectures like distilBERT, which involve extensive parallel matrix operations and multi-head attention mechanisms (Vaswani et al., 2017). These operations can fully exploit GPU parallelization, resulting in substantial efficiency gains and corresponding emissions reductions.

In contrast, logistic regression's single-layer architecture does not take advantage of GPU par-

allelization. Research on GPU utilization patterns indicates that non-deep learning tasks with insufficient computational complexity tend to underutilize GPU resources, resulting in poor efficiency (Gao et al., 2024). Considering logistic regression's minimal computational requirements, it likely failed to exploit the GPU's processing capabilities. Additionally, GPUs maintain higher idle power consumption compared to CPUs (Abe et al., 2012), meaning that baseline emissions remain elevated even when the GPU is underutilized. The combination of underutilization and higher baseline power draw could explain why GPU execution exhibited greater emissions for this simple model architecture. These findings underscore the importance of aligning model architectures with appropriate hardware in practical applications.

The number of epochs showed a consistent and predictable relationship with emissions across both model architectures. Each additional epoch increases the amount of computational workload, thereby extending training duration and directly increasing emissions, supporting findings of previous research (Ariyanti et al., 2025). This suggests that the number of epochs serves as a straightforward measure for managing the environmental footprint during model training, with emissions scaling proportionally to this experimental factor.

Batch size effects, however, differed between architectures. For logistic regression, larger batch sizes reduced emissions by 15-21%, reflecting the computational benefit of processing more samples before updating model parameters. In contrast, batch size did not reliably predict emissions for distilBERT. For practitioners, this suggests that batch size optimization for emissions reduction is architecture-dependent, with predictable effects for simple models but inconsistent effects for more complex architectures.

Overall, these findings extend the predictive approach introduced by Hrib et al. (2024) in several ways. While their analysis focused on classical ML models with synthetic datasets, this research demonstrates that the predictive framework applies to both classical algorithms and modern transformer architectures on real-world text classification tasks of varying sizes. This not only improves confidence in the applicability of this approach across different computational scales, but also allows for emissions trends to be compared across widely different architectures.

The results enable emissions to be estimated for untested factor levels within the experimental ranges. However, the reliability of such predictions varies by experimental factor. Regional and epoch effects showed consistent patterns across both architectures, suggesting that regions with carbon intensities within the tested range would follow a similar trend, and epoch counts within the range of 2 to 6 would follow the same proportional relationship. Moreover, the linear relationship between epoch number and energy consumption reported in prior research (Bouza et al., 2023) further suggests that emissions for higher epoch counts may be extrapolated.

In contrast, hardware and batch size effects were strongly architecture dependent. While it became evident that smaller models benefit from CPU execution and complex models benefit from GPU execution, the threshold of model complexity at which GPU execution becomes

advantageous remains unclear, limiting the confidence in emissions predictions for moderately sized architectures. As for the batch size, emissions effects within the tested range (16 to 64) appear predictable for simple models but remain unclear for complex architectures, suggesting architecture-specific optimization may be required.

Extension beyond the tested ranges should be approached with caution, as the point at which even consistent trends may change cannot be determined from these data. Despite these limitations, extending this approach to additional model architectures and broader hyperparameter ranges would strengthen the predictive framework.

Regarding Research Question 2, correlations between emissions and model performance across both accuracy and F1-score metrics could not be established. This finding indicates that, within the experimental configurations tested, emissions and performance varied independently. This can be explained by two key factors. First, the experimental design adjusted infrastructure factors (region and hardware type), which strongly influence emissions but have minimal direct impact on model accuracy. Geographic region affects emissions through the local carbon intensity without altering performance, and hardware type mainly determines execution efficiency, but does not significantly affect model accuracy (Gyawali, 2023). Consequently, emissions varied substantially across configurations while performance metrics remained relatively stable.

Second, the tested hyperparameter configurations (2-6 epochs, and batch sizes from 16-64) were selected to maintain computational feasibility. However, this constrained parameter space may have limited the detection of potential relationships due to the narrow performance range observed (means exceeding 80% across all metrics). Ariyanti et al. (2025) note that models risk underfitting with epoch numbers that are too low, and can overfit when this hyperparameter is too large, both reducing performance. Similarly, very large batch sizes can limit generalization and reduce performance. These extreme settings, which might simultaneously affect both emissions and performance, were not examined in this study. Investigation across wider hyperparameter ranges would therefore yield more conclusive insights into potential emissions-performance trade-offs.

Several limitations must be acknowledged. First, this study focused exclusively on text classification tasks. Generalizability to other domains such as computer vision or speech recognition therefore remains uncertain. Previous research has shown that energy consumption patterns can differ between computer vision and natural language processing tasks, potentially leading to different emissions profiles (Rodriguez et al., 2024). This highlights the importance of extending the scope of this research to additional tasks.

Secondly, the linear mixed-effects models included dataset as a random intercept based on only three datasets for logistic regression and two for distilBERT, which provides insufficient information to reliably estimate dataset-level variability. While the random intercept structure appropriately accounts for clustering within datasets, the estimated variance components

($\sigma^2 = 2.787$ for logistic regression, $\sigma^2 = 0.503$ for distilBERT) should be interpreted cautiously given this small sample size. Additionally, the inclusion of only two datasets for distilBERT experiments restricts the robustness of findings for this architecture. Future research should validate these findings across a broader range of datasets to establish more reliable estimates.

Furthermore, each experimental configuration was executed only once due to computational constraints. While this was necessary for feasibility, it prevented the assessment of variability across repeated runs. As such, future research should consider repeated measurements in order to strengthen confidence in the findings.

Although CodeCarbon is a widely used library for emissions tracking, its estimation accuracy has known limitations. CodeCarbon allows carbon emissions to be dynamically estimated by measuring the power draw of the hardware over time (Fischer, 2025). However, besides the processor (CPU/GPU) power consumption, other components including peripheral devices, power supply, and cooling, are typically not tracked. Prior research has shown that cooling can contribute to over 10% of the total AI power consumption (Fischer, 2025). Notably, Fischer (2025)'s research on a range of deep learning models also yielded that CodeCarbon's estimation consistently underestimated the actual power consumption by 20-30%. These findings suggest that the absolute emissions values reported in this study likely represent lower bounds, though the relative comparisons between configurations remain informative. Future research employing CodeCarbon could thus introduce hardware systems that have built-in capabilities for measuring the consumption of other components. Alternatively, measurements should be taken externally in order to capture the influence of all environmental components. Fischer (2025) further proposes the introduction of constant factors when tracking emissions, which can account for these estimation errors.

Finally, emissions tracking was limited to model training and evaluation phases. A comprehensive lifecycle assessment would include phases such as data collection, preprocessing, model deployment, and inference (Joshua et al. (2025)). Training and testing phases represent only a component of the total environmental footprint, and for widely deployed models, inference emissions may ultimately exceed these emissions (Ben chaaben et al., 2025). Beyond model execution, research has shown that the embodied carbon footprint of AI models, which includes emissions that arise from hardware manufacturing, contributes significantly to the overall carbon footprint of AI (Faiz et al., 2023). Moreover, training and testing phases each have distinct emissions profiles, yet this study analyzed them together. Future research would thus benefit from examining these phases independently in order to identify specific approaches for reducing emissions when designing ML models. Ultimately, extending the scope to include the complete ML pipeline would offer a more comprehensive understanding of the environmental impact of machine learning architectures and enable more informed strategies for reducing emissions across the AI lifecycle.

6 Conclusion

This study demonstrates that CO_2 emissions from machine learning training are predictable from region, hardware, and hyperparameter choices, with geographic region exerting the strongest influence on emissions. The absence of a correlation between emissions and model performance suggests that sustainability and performance are not necessarily competing objectives. These findings indicate that practitioners can substantially reduce environmental impact without sacrificing performance by strategically selecting low-carbon regions, appropriate hardware and hyperparameters during model design. By extending predictive emissions modeling from classical algorithms to modern transformer architectures and from synthetic to real-world text classification tasks, this work addresses key limitations in prior research and demonstrates the practical applicability of estimating emissions before model training. These findings contribute to the growing body of work on sustainable AI development and provide actionable insights for reducing the carbon footprint of machine learning models.

References

Abe, Y., Sasaki, H., Peres, M., Inoue, K., Murakami, K., & Kato, S. (2012). Power and performance analysis of {gpu-accelerated} systems. *2012 Workshop on Power-Aware Computing and Systems (HotPower 12)*.

Albattah, W., & Khan, R. U. (2025). Impact of imbalanced features on large datasets. *Frontiers in Big Data*, *8*, 1455442.

Almeida, T., & Hidalgo, J. (2011). SMS Spam Collection [DOI: https://doi.org/10.24432/C5CC84].

Anthony, L. F. W., Kanding, B., & Selvan, R. (2020). Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*.

Ariyanti, S., Suryanegara, M., Arifin, A. S., Nurwidya, A. I., & Hayati, N. (2025). Trade-off between energy consumption and three configuration parameters in artificial intelligence (ai) training: Lessons for environmental policy. *Sustainability*, *17*(12), 5359.

Ben chaaben, E., Koch, J., & Mackay, W. E. (2025). " should i choose a smaller model?': Understanding ml model selection and its impact on sustainability. *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–13.

Bouza, L., Bugeau, A., & Lannelongue, L. (2023). How to estimate carbon footprint when training deep learning models? a guide and review. *Environmental Research Communications*, *5*(11), 115014.

CodeCarbon contributors. (2023). Codecarbon documentation [Accessed: 2025-10-10]. https://mlco2.github.io/codecarbon/

DataCamp. (2024). Adamw optimizer in pytorch tutorial [Accessed January 3, 2026]. https://www.datacamp.com/tutorial/adamw-optimizer-in-pytorch

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 4171–4186.

Ember. (2025). Electricity data explorer [Dataset accessed January 3, 2026; country electricity generation and fuel share by metric]. https://ember-energy.org/data/electricity-data-explorer/

Evans, R. H. (1996). An analysis of criterion variable reliability in conjoint analysis. *Perceptual and motor skills*, *82*(3), 988–990.

Faiz, A., Kaneda, S., Wang, R., Osi, R., Sharma, P., Chen, F., & Jiang, L. (2023). Llmcarbon: Modeling the end-to-end carbon footprint of large language models. *arXiv preprint arXiv:2309.14393*.

Fan, Z., Yan, Z., & Wen, S. (2023). Deep learning and artificial intelligence in sustainability: A review of sdgs, renewable energy, and environmental health. *Sustainability*, *15*(18), 13493.

Fischer, R. (2025). Ground-truthing AI energy consumption: Validating codecarbon against external measurements. *arXiv preprint arXiv:2509.22092*.

Gao, Y., He, Y., Li, X., Zhao, B., Lin, H., Liang, Y., Zhong, J., Zhang, H., Wang, J., Zeng, Y., et al. (2024). An empirical study on low gpu utilization of deep learning jobs. *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 1–13.

Geißler, D., Zhou, B., Liu, M., Suh, S., & Lukowicz, P. (2024). The power of training: How different neural network setups influence the energy demand. *International Conference on Architecture of Computing Systems*, 33–47.

Google Developers. (2025). Classification: Accuracy, recall, precision, and related metrics [Accessed January 3, 2026]. https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall

Gyawali, D. (2023). Comparative analysis of cpu and gpu profiling for deep learning models. *arXiv preprint arXiv:2309.02521*.

Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., & Pineau, J. (2020). Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, *21*(248), 1–43.

Hrib, I., Topal, O., Šturm, J., & Škrjanc, M. (2024). Measuring and modeling CO_2 emissions in machine learning processes [Held 7–11 October 2024]. *Proceedings of the 27th International Multiconference Information Society (IS 2024)*, 67–72. https://is.ijs.si/wp-content/uploads/2024/11/IS2024_Volume-C-1.pdf

Hugging Face. (2024). *Transformers*. https://huggingface.co/docs/transformers

Intel Corporation. (2025). Cpu vs. gpu: What's the difference? [Accessed January 4, 2026].

Islam, M. S., Zisad, S. N., Kor, A.-L., & Hasan, M. H. (2023). Sustainability of machine learning models: An energy consumption centric evaluation. *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1–6.

Joshua, C., Marvellous, A., Matthew, B., Pezzè, M., Abrahão, S., Penzenstadler, B., Mandal, A., Nadim, M., & Schultz, U. (2025). Sustainable ai: Measuring and reducing carbon footprint in model training and deployment. *EAI Endorsed Transactions on Tourism, Technology and Intelligence*.

Jurafsky, D., & Martin, J. H. (2025). *Speech and language processing* [Draft of August 24, 2025]. https://web.stanford.edu/~jurafsky/slp3/

Lacoste, A., Luccioni, A., Schmidt, V., & Dandres, T. (2019). Quantifying the carbon emissions of machine learning. https://arxiv.org/abs/1910.09700

Li, D., Chen, X., Becchi, M., & Zong, Z. (2016). Evaluating the energy efficiency of deep convolutional neural networks on CPUs and GPUs. *2016 IEEE international conferences on big data and cloud computing (BDCloud), social computing and networking (SocialCom), sustainable computing and communications (SustainCom)(BDCloud-SocialCom-SustainCom)*, 477–484.

Lorenzoni, G., Portugal, I., Alencar, P., & Cowan, D. (2024). Exploring variability in fine-tuned models for text classification with distilbert. *arXiv preprint arXiv:2501.00241*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. http://www.aclweb.org/anthology/P11-1015

Our World in Data. (2025). Carbon intensity of electricity generation [Dataset accessed January 3, 2026; processed by Our World in Data from Ember and Energy Institute data]. https://ourworldindata.org/grapher/carbon-intensity-electricity

Prince, S. J. D. (2023). *Understanding deep learning*. MIT Press.

PyTorch Contributors. (2023). *Pytorch* (Version 2.9.1). https://pytorch.org

PyTorch Developers. (2025). *Torch.nn.bceloss — pytorch 2.9 documentation* [Accessed on February 17, 2026; Official API documentation for the BCELoss loss function, measuring binary cross-entropy between target and input probabilities.]. https://docs.pytorch.org/docs/stable/generated/torch.nn.BCELoss.html

PyTorch Documentation. (2025). Torch.nn.bcewithlogitsloss — pytorch 2.x documentation [Accessed January 3, 2026].

Raschka, S. (2022, April). *Losses learned — optimizing negative log-likelihood and cross-entropy in pytorch (part 1)* [Accessed on February 17, 2026]. https://sebastianraschka.com/blog/2022/losses-learned-part1.html

Rodriguez, C., Degioanni, L., Kameni, L., Vidal, R., & Neglia, G. (2024). Evaluating the energy consumption of machine learning: Systematic literature review and experiments. *arXiv preprint arXiv:2408.15128*.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Schmidt, V., Ligozat, A.-L., Schaefer, T., & Lacoste, A. (2021, March). *Codecarbon: Estimate and track carbon emissions from machine learning computing* (Version 1.1.0). Zenodo. https://doi.org/10.5281/zenodo.4658424

SciPy Documentation. (2025). Scipy.stats.pearsonr — scipy 1.16.2 manual [Accessed January 4, 2026].

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. *9th Python in Science Conference*.

Van Wynsberghe, A. (2021). Sustainable AI: AI for sustainability and the sustainability of ai. *AI and Ethics*, *1*(3), 213–218.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

Appendix

All the code for this research can be found in the following GitHub repository: repository link.

Appendix on Generative Artificial Intelligence use

ChatGPT was used to assist with specific parts of the research and report. ChatGPT was applied in the following ways:

- The tool was used to brainstorm research ideas, using prompts such as: "What are currently underexplored applications in AI sustainability research?". Outputs were used as inspiration for possible research directions.
- ChatGPT was also used for structuring and organization. In particular, it was useful in suggesting appropriate structure formats for the Methodology and Discussion sections. Prompts such as the following were used: "What kind of structure is the most appropriate to convey my methods clearly?". Insights from these prompts were used as ideas for a basic structure.
- ChatGPT was also used for language and clarity through prompts like: "Where can I improve/correct the clarity/grammar/spelling in this section?". Outputs were used to revise and refine specific sections.
- Finally, ChatGPT assisted in the coding section of this research by verifying correct usage of specific libraries (CodeCarbon, Transformers) and clarifying specific coding errors. Prompts such as the following were used for this: "Can you clarify what this Python error means?". Outputs were used to revise broken code or improve code.

All the core work, including the data collection and analysis, and interpretation of results was conducted independently or in consultation with supervisors. ChatGPT served as a supplementary tool for improving clarity, structure and code. All outputs from ChatGPT were critically evaluated and, when possible, compared to other available literature before considering them. As such, the final thesis represents my own work and interpretations.