Machine Learning with Life Cycle Assessment Data

Hannah Wang

1. Introduction

With climate change becoming an urgent topic, sustainable operations and policy decisions have attracted wide attention. Sustainability means a balance between the environment, economy, and society. Currently, the most popular tool to assist with sustainability evaluation is Life Cycle Assessment (LCA). LCA can model the life cycle of any process and product from raw material acquisition to disposal (1). Each process or product consists of multiple functional unit processes, and each unit process consumes input resources and output emissions. Traditionally, analysts would collect raw inputs and outputs data, and convert the raw data into equivalent factor numbers to calculate impact indices (e.g., climate change impact) through solving systems of linear equations (2).

However, for complex systems, it can be time-consuming to solve the equations. Consider the following example - evaluating the climate change impact on the life cycle of a car manufacturer, the supply chain is composed of numerous component manufacturing and assembling processes. Each process was fed on resource inputs and output emissions. Emissions that go to nature including greenhouse gases (GHG) and chemicals will create climate change impact. To calculate the impact, disparate GHG and chemicals need to be transformed into their equivalent CO_2 emission using the corresponding factors. One can imagine that there would be numerous examinations, and even with the help of LCA software, it can be exhausting. Furthermore, sharing the model or building models through collaborations can be challenging because the models often heavily depend on specific local software environments.

If machine learning algorithms can be used to predict impacts given sets of input parameters (resource, material, energy, etc.), we can save time by avoiding the intermediate calculations. This can be beneficial for real-time decision-making. Moreover, well-built machine learning models are easy to share, and through learning from previously built machine learning models, prediction accuracy may increase as well. Despite the potential of machine learning algorithms, there were only a few studies that discussed the application of Machine Learning in LCA field. The reason may lie in that LCA data is usually proprietary and are therefore not public to the Machine Learning world. To address this gap, in this paper, I will apply Machine Leaning Algorithms on the car life cycle dataset generated by `carculator` using the pipeline from (2) and discuss the findings.

2. Data

The dataset was generated by `calculator`, an open-source fully parameterized Python model to perform environmental and economic life cycle assessments of vehicles (3). Typically, the LCA of a passenger vehicle includes the raw material extraction, the manufacture of the vehicle, its distribution, use and maintenance, and its disposal. As a result, the dataset compiled inventories of material and energy required along the life cycle of the vehicle (features), and several impact categories (responses).

In particular, the dataset contains 15,820 entries. There are 80 input features (77 continuous variables, 3 categorical variables), and 21 different output environmental and economic indices (Appendix Table S1). To name a few, continuous input variables include those related to compressed natural gas (CNG), battery life parameters, energy consumption, glider, vehicle mass, etc.; categorical variables are country, vehicle size, and powertrain type. Responses are all continuous variables, including human toxicity, metal depletion, climate change, fossil depletion, etc. For experimental and demonstration purposes, only these four impact categories mentioned were used for training/testing algorithms, and visualization. The data generation process can be found in the repo.

The underlying structure of the data was that: each impact category was determined by only a subset of input parameters, and there might be some similarities between the output impact categories. Therefore, the goal is two-fold: (1) uncover the optimal subset of input parameters for the outputs; (2) efficiently build predictive models with these subsets of input features.

3. Problem Formulation

This impact prediction problem can be formulated as a simple output prediction task or multi-output prediction task. Empirically, some studies may care about only one kind of impact, such as climate change; others may want to predict several impacts at once. Here, I will account for both scenarios.

In general, for both scenarios, the prediction problem can be formulated as an empirical risk minimization model.

$$\min_{\mathrm{W}} = \sum_{\mathrm{i}=1}^{\mathrm{t}} \mathrm{L}(\mathrm{W_i}|X_i, Y_i) + Reg(\lambda, W)$$

Where L is the least square loss function for our linear regression problem. $Reg(\lambda, W)$ is a regularization term, which differs by algorithms (Table *1*). t represents the number of tasks. For mono-output, t = 1; for multi-output in this paper, t = 4. $X = \{X^j = n^j \times p | j \in 1, \dots, t\}$ is the feature matrix, consists of p = 80 features, and $Y = \{Y^j = n^j \times 1 | j \in 1, \dots, t\}$ is the response matrix, and it consists of t = 1 impact category for mono-output regression and t = 4 for multi-output regression task. Each task j contains n^j subjects. All tasks were trained on the same data; thus, $n^j = 15820\ \forall\ j$, and $X^j = 15820 \times 80\ \forall\ j$. W = p × t is the coefficient matrix, where W^j is the jth column of W refers to the coefficient vector of task j.

The meaning of multi-output regression is that different environmental impacts may have relatedness with each other, and mono-output regression cannot capture the relatedness as it predicts each impact separately. For multi-output prediction, we can put a prior on W to capture the task relatedness. That is, the empirical risk minimization model can be derived through a probabilistic view (4), and the $Reg(\lambda, W)$ will jointly modulate the prior structures of W^js. Assume that the data {X, Y} is drawn independently from Gaussian distribution (parametrized by σ). A prior on W is defined as follows. The i-th row of W, denoted as $W_i \in R^{1 \times t}$

corresponds to the i-th feature in all tasks. Assume that W_i is generated according to the exponential prior (parametrized by δ^i).

$$p(W_i|\delta^i) \propto exp(-||w_i||\delta_i), i = 1,2,\dots,n; n = 15820$$

Then, the prior for W can be expressed as:

$$p(W|\delta) = \prod_{i=1}^{n} p(W_i|\delta_i)$$

Thus, the posterior distribution for W is proportional to the product of the likelihood of data and the prior of W:

$$p(W|X,Y,\sigma,\delta) \propto p(Y|W,X,\sigma)p(W|\delta)$$

Taking the negative logarithm of the equation, we can obtain the empirical risk minimization function with $Reg(\lambda, W) = quadratic\ loss + L_{2,1}norm.$

4. Methods

Due to the fact that not all features determine the impacts, we would want to select a subset of meaningful features for each impact category, that is, we hope to shrink some parameters among the 80 features to zero, but at the same time perform good prediction. Therefore, we select algorithms with regularization terms that involve the L_1 norm to enforce the sparsity on the features. For the mono-output regression problem, we examined the performance of LASSO and Elastic Net using R^2. For the multi-task output regression problem, we compared Multi-task Elastic Net (MTEN), Efficient Multi-task $L_{2,1}$ norm (EMT), and Neural Network (Keras Library) using MSE and computation time. EMT was a variance for MTEN; both leveraged $L_{2,1}$ norm as their regularizer, but MTEN was solved through coordinate gradient descent, and EMT was solved through an accelerated proximal gradient descent.

LASSO, Elastic Net, MTEN were performed using the Python Scikit Learn Library (SKLearn). Neural Network model was built using Python Keras Library. EMT was implemented by referring to Liu et al.'s paper (4), their MatLab code, and an early Python repo. Linear regression models from SKLearn implemented normalization for data implicitly; thus, no data preprocessing was done for Lasso, Elastic Net, and MTEN. For Keras model, data was standardized and one-hot encoded before running the algorithms as suggested by the Keras documentation. For EMT, data was normalized before running the algorithm. All implementations can be found in this project repo.

Table 1 Algorithms, Loss functions, and Regularization terms

	Mono-output regression (Single Task learning)		Multi-output regression (Multi-Task Learning)		
Algorithm	LASSO	Elastic Net	MTEN	EMT	Neural Network (Keras)
$Loss(W\|X,Y)$	$\left\|\|w-Xy\|\right\|_2^2, w \in n\times 1, y \in n\times 1$		$\left\|\|W-XY\|\right\|_F^2, W \in p\times t, Y \in n\times t$		
$Reg(W,\lambda)$	$\lambda_1\left\|\|w\|\right\|_1$	$\lambda_1\left\|\|w\|\right\|_1+\lambda_2\left\|\|w\|\right\|_2^2$	$\lambda_1\left\|\|w\|\right\|_{2,1}+\lambda_2\left\|\|w\|\right\|_F^2$		-

5. Results and Discussion

5.1 Mono-output regression

For mono-output regression, I separately leveraged the feature data X to predict four impact categories: Y_1 = climate change, Y_2 = fossil depletion, Y_3 = metal depletion, Y_4 = human toxicity. These 4 categories were chosen as they each represents different aspects of sustainability - Y_1 relates to all the aspects; Y_2, Y_3, represent the economy and environment; Y_4 represents society. R^2 of Each $Lasso(X,Y_i)$ model, and the number of selected features were presented in *Table 2*. Overall, the R^2 was the highest for the climate change predictive model, meaning that the subset of features chosen for predicting climate change can account for more variance in the data. Additionally, it was observed that several subsets of input features were repeatedly chosen for predicting each impact (Figure 1). In particular, the middle four target nodes where all arrows pointed to were the four impact categories. Source nodes were the input features, and those that were closer to the center were subsets of features shared among more impact categories; those that were at the periphery were more specific features characterizing a specific target impact (in this case the periphery nodes characterized climate change).

Table 2 R^2 and number of selected features

	Climate change	Fossil Depletion	Human toxicity	Metal depletion
R^2	0.646	0.545	0.472	0.430
Number of features	25	19	21	15

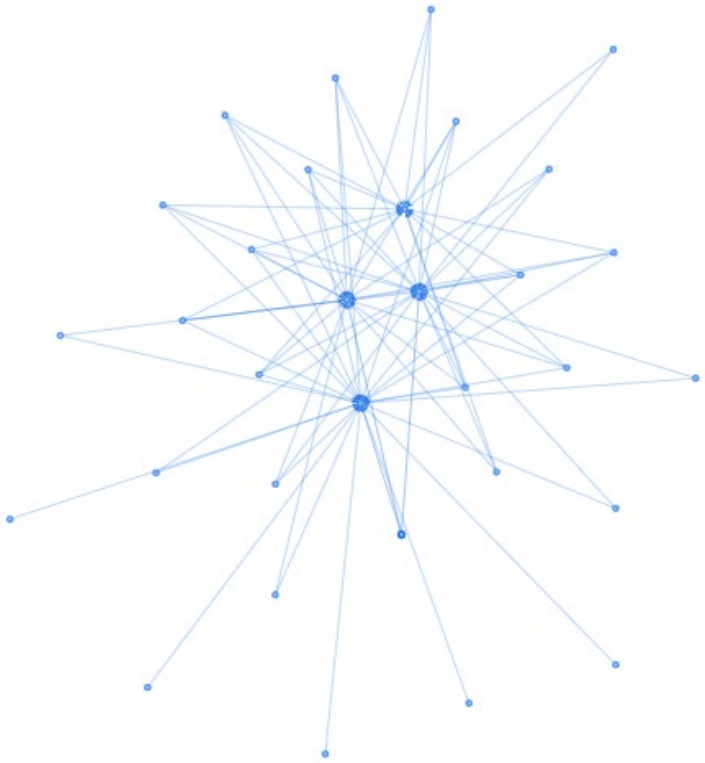

Figure 1 Selected feature-response relationship from Mono-output Lasso regression

The reason that R^2s was not ideal for the other three impact predictive models can be that some critical features were not included in the model. This might result from the limitations of Lasso in the following two scenarios (5). These two scenarios match the properties of our data - some input features (resources, materials, energy) generated similar emissions that can lead to the same response (environmental impact).

Scenario 1: there is a group of variables among which the pairwise correlations are very high; then the lasso tends to select only one variable from the group and does not care which one is selected.

Scenario 2: when n > p, if there exist high correlations among predictors, ridge regression performs better than lasso.

Thus, to address these limitations, Elastic Net was performed. Ideally, we would hope that a group of correlated features should all be selected if one of them is selected. To examine the effectiveness of Elastic net, we tested it on the climate change impact prediction task.

Indeed, the R^2 from Elastic net climate change model improved to 0.684. In addition, Table 3 showed the selected features from Lasso and Elastic net. Number of features increased to 37, and it was observed that group of correlated features were selected. For example, previously only “fuel cell power area density” was selected, now “fuel cell cost per kW”, “fuel cell lifetime hours” were selected too. In conclusion, Elastic net algorithm is more appropriate for building

an interpretable single predictive linear model, and it preserved the sparse nature for feature selection while maintaing the prediction power.

Table 3 Feature selection by Lasso and Elastic Net

	Lasso	Elastic Net
Selected Features	CNG tank mass intercept, auxilliary power base demand, average passenger mass, battery cell production energy, battery lifetime kilometers, battery onboard charging infrastructure cost, combustion exhaust treatment cost, combustion fixed mass, combustion powertrain cost per kW, cooling thermal demand, electric fixed mass, electric powertrain cost per kW, energy battery cost per kWh, energy battery mass, fuel cell power area density, fuel mass, fuel tank cost per kg, glider base mass, glider cost intercept, heat pump cost, inverter mass, kilometers per year, lifetime kilometers, power to mass ratio, year	CNG tank mass intercept, LHV fuel MJ per kg, auxilliary power base demand, average passenger mass, average passengers, battery cell power density, battery cell production energy, battery lifetime kilometers, battery onboard charging infrastructure cost, cargo mass, combustion exhaust treatment cost, combustion fixed mass, combustion powertrain cost per kW, cooling thermal demand, electric fixed mass, electric powertrain cost per kW, energy battery cost per kWh, energy battery mass, fuel cell cost per kW, fuel cell lifetime hour', fuel cell power area density, fuel mas, fuel tank cost per kg, glider base mass, glider cost intercept, glider cost slope, heat pump cost, heating thermal demand, inverter mass, kilometers per year, lifetime kilometers, power battery cost per kW, power to mass ratio, powertrain fixed mass, country, size, year

5.2 Multi-output regression

As shown in Figure 1, subsets of features were shared among different impact categories. Considering this sharing property and the performance of elastic net, Multi-task elastic net (MTEN), the joint feature learning algorithm for related tasks, was performed. That is, the algorithm can jointly select a subset of features for all impact predictions at once. R^2 and MSE were used to evaluate the performance of MTEN. The overall R^2 was 0.655, 39 features were selected, and the MSE was 0.002. Compared to single-task models, for climate change, the multi-task model did not improve in accounting for variance in the data; for the rest of the impact categories, the mult-task elastic net model performed better. This may be because MTEN selected features that were common among all features; nevertheless, if some of the selected features were not as critical to an individual response, the model performance can be inferior to the single task predictive model. Overall, the R^2 and MSE were acceptable, and the multi-task model can be an efficient method for impact prediction. Nevertheless, which tasks can be predicted together for obtaining optimal performances should be further investigated (6).

In multi-task elastic net, the regularizer term is $L_{2,1}$ norm, which is a non-smooth convex function, and it can converge slowly. The implementation from SKLearn leveraged coordinate descent to solve the optimization problem. With a view to further improve the computation efficiency, an accelerated proximal gradient descent algorithm (called EMT) from Liu et al. (4) was implemented. Specifically, they proposed to reformulate the non-smooth convex optimization problem into a constrained smooth convex optimization problem. Then, they proposed to employ Nesterov's accelerated gradient method, an optimal solver for smooth convex optimization problems to solve this reformulated constrained convex optimization. In particular, they suggested conducting Euclidean projection onto the set of constraints in each step of Nesterov's method.

To ensure fair comparisons for the computation time of MTEN and EMT, we set the following parameters. Regularization parameters for the $L_{2,1}$ norm were set to 0.05, and the max iteration was set to 1000. For each algorithm, we trained using 5-fold cross-validation. Several tests on the LCA dataset showed that EMT was on average around 4 times slower than MTEN. The reason might be that coordinate descent can leverage the sparse property of our input feature matrix. Therefore, for our data, to accelerate the computation efficiency for multi-task learning, perhaps one future direction can be looking for sparse proximal gradient descent algorithms.

Aside from linear models, given the underlying complex network relationships between inputs and outputs, a neural network model with one hidden layer was built using Keras to capture these implicit relationships. The MSE of the Keras model was 0.005, which was about the same as the 0.002 from Multi-task elastic net model. In this case, I would prefer the higher interpretability Multi-task elastic net regression model.

5.3 Comparison with other datasets

As far as I am concerned, there is only one paper on this dataset (2). The author generated 700,000 entries with the calculator package, and they did the single-task learning for Climate Change impact prediction. They constructed one Linear model and one Artificial Neural Network model. However, no detailed implementations were documented.

On an abstract level, this is similar to the gene selection problems, where a subset of genes is related to determining specific diseases or expressions. However, it differs in the data dimension. In gene expression datasets, number of samples can be largely greater than the feature dimension (i.e., $n \ll d$), while in LCA datasets, depending on the industries and the technologies being assessed, either $n > d$ or $n < d$ (e.g., new technology investigation can suffer from a small dataset) might happen. As a starting point, I imagine multi-task algorithms developed for the computational biology field can be transferred for computational sustainability problems.

6. Conclusion and Future Work

In this paper, the performance of 5 supervised machine learning algorithms on the LCA dataset was examined. The results showed that Multi-task Elastic Net has the potential to be an efficient solver for building an interpretable model that predicts several impacts at once. In addition, there are still spaces left for accuracy and computational efficiency improvements.

While only four impact categories were selected for exploration, in the future, it can be meaningful to build models with more impact categories. If more impact categories are of interest at the same time, leveraging algorithms that jointly selected features may not be appropriate. That is, among numerous impact categories, there can be outlier impact categories, i.e., categories that were determined by features much different than other impact categories. If this is the case, algorithms that can detect outlier tasks and select different subsets of input features for each impact category may be more desirable. Moreover, to better assist decision-making, explainability and interpretability of the models should also be kept in mind.

7. Appendix

Table S1. Features and Responses of the LCA dataset

Features (Inputs)	CNG pump-to-tank leakage,CNG tank mass intercept,CNG tank mass slope,CO2 per kg fuel,H2 tank mass per energy,LHV fuel MJ per kg,aerodynamic drag coefficient,auxilliary power base demand,average passenger mass,average passengers,battery DoD,battery cell energy density,"battery cell energy density, LFP","battery cell energy density, NCA","battery cell energy density, NMC","battery cell mass share, LFP","battery cell mass share, NCA","battery cell mass share, NMC",battery cell power density,battery cell production energy,battery cell production energy electricity share,battery charge efficiency,battery discharge efficiency,battery lifetime kilometers,battery onboard charging infrastructure cost,cargo mass,charger mass,combustion exhaust treatment cost,combustion fixed mass,combustion mass per power,combustion power share,combustion

	powertrain cost per kW,converter mass,cooling energy consumption,cooling thermal demand,drivetrain efficiency,electric fixed mass,electric mass per power,electric powertrain cost per kW,emission factor,energy battery cost per kWh,energy battery mass,energy cost per kWh,engine efficiency,frontal area,fuel cell ancillary BoP mass per power,fuel cell cost per kW,fuel cell essential BoP mass per power,fuel cell lifetime hours,fuel cell own consumption,fuel cell power area density,fuel cell power share,fuel cell stack efficiency,fuel mass,fuel tank cost per kg,fuel tank mass per energy,glider base mass,glider cost intercept,glider cost slope,glider lightweighting cost per kg,heat pump cost,heating energy consumption,heating thermal demand,interest rate,inverter mass,kilometers per year,lifetime kilometers,lightweighting,maintenance cost per glider cost,markup factor,power battery cost per kW,power distribution unit mass,power to mass ratio,powertrain fixed mass,powertrain mass per power,rolling resistance coefficient,country,size,powertrain,year
Responses (Outputs)	freshwater ecotoxicity,human toxicity,marine ecotoxicity,terrestrial ecotoxicity,metal depletion,agricultural land occupation,climate change,fossil depletion,freshwater eutrophication,ionising radiation,marine eutrophication,natural land transformation,ozone depletion,particulate matter formation,photochemical oxidant formation,terrestrial acidification,urban land occupation,water depletion,noise emissions,renewable primary energy,non-renewable primary energy

8. References

1. Finnveden G, Hauschild MZ, Ekvall T, Guinée J, Heijungs R, et al. 2009. Recent developments in Life Cycle Assessment. *J. Environ. Manage.* 91(1):1–21
2. Starlinger V, De C, Lope R, Ghoshdastidar D. Machine Learning Benchmark to Assess the Environmental Impact of Cars
3. Cox B, Mutel CL, Bauer C, Mendoza Beltran A, Van Vuuren DP. 2018. Uncertain Environmental Footprint of Current and Future Battery Electric Vehicles. *Environ. Sci. Technol.* 52(8):4989–95
4. Liu J, Ji S, Ye J. 2009. Multi-task feature learning via efficient ℓ2, 1-norm minimization. *Proc. 25th Conf. Uncertain. Artif. Intell. UAI 2009*, pp. 339–48
5. Zou H, Hastie T. 2005. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67(2):301–20
6. Standley T, Zamir A, Chen D, Guibas L, Malik J, Savarese S. 2020. Which tasks should be learned together in multi-task learning? *37th Int. Conf. Mach. Learn. ICML 2020*. PartF168147-12:9057–69