My project implements what my hackathon worked on during my trip to the UK. There have been some slight changes since the hackathon, as I didn't quite like the original output. The changes that I implemented were the prediction of future years up to about 10 years into the future of the collected data. The original only tested from collected years; the only issue is that it depends on the past, so like any percentages can go above 100%, thus forcing the use of a cut-off. The main core aspects of the code that were developed during the hackathon still remain, and we still proceed to use linear regression for the model. The data used to train the model is whatever was available, which, at the time of implementation, is from 2019 to 2022. The data content is the air quality that was tested from around the worl at each country. The data that was used is a CSV file from WHO, which is the World Health Organization. To know more about the model, we need to know about linear regression. So, Linear regression is an attempt to find a linear relationship between some sort of input variable and an output variable. There can be more than one linear regression, as it could be a multi-linear regression as well, but we used just a simple linear regression. The input variable that was chosen for the code is the year of measurement, and the output variable can be 2 options when running the code. It can be either the coverage of PM2.5 in percent or the concentration of PM2.5 in micrograms/m^3 that the user can choose to see the actual number of microorganisms that are breathable, or just the percent due to the required implementation of the cutoff. The model learns two parameters: an intercept, which in the formula is b0, and a slope, b1, so predictions follow $y = b0 + b1 \cdot year$. I compute b0 and b1 using the closed-form “sum” formulas based on Σx, Σy, Σx^2, and Σxy, then test accuracy on a small holdout set and report R^2 and MAE. This matters because it shows the model is not just producing numbers, but can be evaluated for how well it generalizes to years it did not train on.

This connects to Ada Lovelace's ideas because Lovelace argued that a computing machine could manipulate symbols according to rules, and that the same machine could perform many different tasks depending on the instructions. My program follows that pattern: it is a general-purpose pipeline that takes symbolic inputs, a target variable choice, a continent label, and a prediction year, then applies a fixed set of rules such as cleaning, grouping, fitting, and predicting, and produces outputs, which are a forecast value, plots, and a table. The regression math is a clear example of "symbol manipulation": the program does not "understand" pollution, but it reliably transforms data symbols through a defined procedure into a model and predictions.

The dataset has a global dimension because WHO air-quality monitoring and reporting differ dramatically across countries and regions. I aggregate the data into continents to reduce any errors due to some cities having inefficient readings or readings that are just nonexistent. We also generalize it to the continents to compare broad trends in measurement coverage or pollution levels across the world. This reveals inequality in at least two ways. Firstly, the time span of the data collected might not be equal due to differences in the availability of monitoring facilities, stable institutions, and funding. Secondly, despite the similarity in levels of pollution, not all countries might have equal opportunities to monitor and publish data, and this might create interdependence in the visibility of the problem. There is also interdependence in the sense that air pollution knows no boundaries, and economies are interlinked.

The experience I had going into the project was honestly quite minimal. I am a computer science major who has no experience in machine learning whatsoever. I did a few classes in Python as a freshman, but I have yet to use much of Python. The main languages used in classes so far have been C and Java, so I was able to get a start. I attempted to learn about my group during the hackathon, and we were able to mesh out a working algorithm, but I wanted more from what we

created. So I used AI to help a little with my lacking areas to expand upon the algorithm. The background of being a Computer Science major definitely allowed me to understand and analyze the data on a deeper scale. Then, as my personality, I love nature and the health of that nature and of human beings, so creating an algorithm to predict the levels of contaminated air that can get into your lungsis an interesting topic. So the conclusion is that working in a team really does help to hasten the process of development and implementation. I can draw that the algorithm is not quite perfect, as my inexperience in the topic does show, and the lack of readings also contaminates the data a smidge. The algorithm from the math does get as close as possible and, in conclusion, does predict the future air quality of the specific continent chosen.