CSC110 Project Final Submission: Meat Monitor

Ian Lavine, Prithee Roy, Jeremy Wong, Ayush Sahi

Monday, December 14, 2020

Problem Description and Research Question

With many scientists predicting that we have approximately 10 years left to prevent irreversible damage to our planet as a result of climate change and global warming, it has become clear that as a society we must act proactively to combat the issue before it becomes too late. In recent times, CO2 emissions have continued to rise past the predictions of many scientists around the world. We have observed that a surprisingly significant percentage of greenhouse gas emissions come from agriculture, specifically the raising of livestock, which is approximated to account for 10% of all greenhouse gas emissions in the United States. According to a study published by Oxford, cutting meat products from an individual's diet can reduce their carbon footprint by up to two-thirds. Research indicates that with a carbon footprint of 3.3 tons of greenhouse gas emissions, the average meat-eater produces far more than that of a vegetarian diet (1.5 tons). This suggests one of the easiest ways to combat climate change is simply through improving and altering one's diet. However, the impacts of going vegetarian undoubtedly vary widely per person, depending on several factors. These factors can range from how much meat they are currently eating, how much they can cut out by going vegetarian, the type of meat they were consuming, the way those animals are raised, what substitutes the person in question could replace the meat within their diet, etc. These factors will be affected by any given person's location, wealth, age, culture, and more. For example, a wealthier person may be able to buy more ethically raised meat, which would require more land and resources, creating a larger carbon footprint. If that person were to cut meat from their diet, it would have a bigger impact than if the average person became vegetarian. For these reasons, it may make more sense for one person to be vegetarian, whereas others may have more cost-effective measures they can take to curb their carbon footprint. Thus, bringing us to our query. **We want to know the extent to which a person's meat consumption and location affect the impact on their carbon footprint if they were to become vegetarian. More specifically, how much greenhouse gas emissions, in tons, any given person can reduce by becoming vegetarian**.

Dataset Description

One data set we have used is from the United Nations Food and Agricultural Organization in the form of a csv file. This data set contains comprehensive information on the average person's yearly meat consumption in most countries across the world. The value of the average meat consumption is broken down into the types of meat: beef and buffalo, poultry meat, pig meat, and mutton & goat meat. This information will be compared to more data, specifically data sets that contains statistics regarding total greenhouse gas emissions from livestock agricultural practices, and average meal serving sizes of different types of meat. This data set after being processed and manipulated through our algorithm, will illustrate the relationship between any user's weekly servings of different types of meat, and the exact emissions it causes, and how that compares to the average meat consumption induced CO2 emissions of people in their country.

Here is a sample of this data set:

Entity	Code	Year	Bovine meat food supply quantity (kg/capita/yr) (FAO, 2020)	Poultry meat food supply quantity (kg/capita/yr) (FAO, 2020)	Mutton and goat meat food supply quantity (kg/capita/yr) (FAO, 2020)	...
...	...	...	...	...	...	...
Canada	CAN	2013	30.25	36.68	22.81	...
Canada	CAN	2014	28.44	37.05	22.86	...
Canada	CAN	2015	27.43	37.88	24.88	...
Canada	CAN	2016	21.00	38.57	23.26	...
Canada	CAN	2017	18.13	39.02	24.42	...
Cape Verde	CPV	1961	0.79	0.18	1.53	...
Cape Verde	CPV	1962	0.68	0.18	1.58	...
Cape Verde	CPV	1963	0.79	0.17	1.71	...
...	...	...	...	...	...	...

Another data set we have used is from a study by Poore, J., Nemecek, T. in the form of a csv file. Although the data set is in the form of a csv file, we have decided to manually implement the data into our python file using a dictionary. This is because the csv file contains a lot of unnecessary information, of which we only need 4 statistics. This data set contains information on the CO2 emissions for every unit mass of certain food items, including meats. The data points we have chosen to use are beef, pig meat, lamb mutton meat and poultry meat. This data set, combined with the first data set, will allow us to compute the total amount of CO2 emissions for any given amount of meat in our four chosen categories, which will be used in the final product of meat monitor.

Here is a sample of this data set:

Food	Greenhouse gas emissions per 100g of protein (kgCO2eq per 100g protein)
...	...
Pig Meat	7.61
Potatoes	2.71
Poultry Meat	5.70
Prawns (farmed)	18.19
Rice	6.27
Root Vegetables	4.30
...	...

We have also used four other data sets from 'healthline' and 'verywellfit' which tell us the average serving size for each meat type. This data is in the form of text in an online article. Therefore, we created a dictionary in line 28 called 'serving_size_per_animal' which has the meat types as keys with their associated serving sizes as values. This data allows us to calculate the average quantity of meat the user consumes when he or she inputs how many times per week they eat a certain meat.

Computational Overview

Major Computations

As stated before, the main dataset we will be referring to contains various data points on per-capita meat consumption by type kilograms per year. This dataset is currently in the form of a csv file. To use this data in python, we used the pandas module to import the data, and then read the file and index different data points on a need to need basis. For example, we have only indexed to the most recent data points per country, as the meat consumption in earlier years are irrelevant.

To begin, a stored variable is generated called 'proper_country_data', which reads the csv file of the first data set, which contains all the data points of meat consumption in every country in the most recent year that the data is available. This stored variable is then used to mutate another stored variable called 'countries', which is initially an empty dictionary, but is eventually mutated into a dictionary with the the values from 'proper_country_data'. This dictionary contains the data values of the average meat consumption of the **user's inputted country** in the most recent year. This stored variable allows us to make calculations comparing the user's meat consumption with the average person in their country.

Next, in our code, we have created 3 classes: 'Country', 'Animal', and 'User'. This is so we can create associated objects that build a profile of the user, which we can then access in our data calculations.

- In the 'Country' class, we have the class attributes of the name of the country of where the user is located and its average consumption in each respective meat type.
- In the 'Animal' class, we have the class attributes for the name of the animal, the user's inputted weekly consumption for that certain animal, and the user's location. When the user enters his or her meat consumption information, an object is initialized with these three class attributes. The seven other instance variables below the first three are introduced to store values of our calculations for that certain animal for the user.
- In the 'User' class, we have the class attributes of the name of the user, their location and the type of meat products that they consume. Similar to the 'Animal' class, the other instance variables are to store values of our calculations, adding onto the profile of the user. The 'User' class contains a variable 'animal_list' which is a dictionary of the four animal types in 'str' format, to four animal classes. These animal classes get instantiated when the 'User' inputs information about their meat consumption, through the 'create_animal_classes' function.

To determine the total emissions emitted to produce the meat the user consumes, we calculate the total emission for each animal by multiplying the users' weekly consumption of each respective meat with the data of the kilograms of CO2 emissions per kilogram of that type of meat, respectively, in the form of a dictionary starting on line 21 of the python file. This is done in the function 'find_stats' under the 'Animal' class. This is then summed in the user class by adding the total emissions for each meat type to achieve the total emissions for the user's meat consumption. This is done in another function called 'find_stats' under the 'User' class. We have chosen to have the user input their weekly consumption instead of a year or day basis because we came to the conclusion that most users don't know how much meat they consume in a year and meat consumption in one day can vary a lot. The weekly consumption of the user is also inputted as the number of servings of meat, instead of a unit measure of mass. These serving sizes are in the form of a dictionary in line 28. We are using servings as a measure instead of unit mass because we understand that it may be hard for the user to determine exactly how many grams or kilograms of meat they consume in a week.

To compare the user's meat consumption with the average person in their respective country that they have inputted, we have taken their input data of their weekly meat consumption for each type of meat and stored it into the weekly_consumption instance variable of the respective 'Animal' object for the user. Then, for each type of meat, we found the difference between the meat consumption for the user and their country for each individual meat type. This process is found in the function 'find_stats' under the 'Animal' class, where we find 'self.consumption_difference'.

After finding the difference between the meat consumption of user and the average person in their respective countries, we can find the difference in the CO2 emissions that would be emitted as well. We have done this by subtracting the total CO2 emissions of the average person in the user's country and the user's total emissions. This process is found in the function 'find_stats' under the 'User' class. We then divided this by the average meat consumption in the user's country to find the percentage difference.

The user can also input their goal meat consumption, which we would return how much CO2 emissions their new meat consumption would emit to produce the new meat quantity, and it returns the amount of CO2 emissions they would save if that new value of meat quantity was produced compared to the original.

After that, to visualize the data that we have just been presented with in these steps, we used matplotlib to graph the users current CO2 emissions due to their meat consumption, their potential CO2 emissions if they were to change their diet, and the CO2 emissions of the average person in their country as a reference.

User Interface (UI)

We have illustrated the results of each computation using the TkInter python package. The user will have to input their name, location and approximate servings per week for each of the four meat categories of beef, pork, lamb and poultry, and once inputted, a generated aggregation of information and statistics in regards to their CO2 impact, is displayed through Tkinter. Here is a step-by-step run through of what the user should expect in our program and how it displays the results in a visual and interactive way:

1. When the user runs main.py, a window opens displaying the starter screen. This should be a window with a purple background and logo with an indication to the user to input their meat consumption. Four labelled sliding bars are given to the user to input their meat consumption.

2. When the user has finished adjusting the values of their meat consumption using the sliders, they can press the button 'submit', which will take the user to another screen in the same window, which displays the amount of CO2 emitted to produce the amount of meat they consume in a week.

3. When the data on the user's CO2 impact is displayed, the number is either in green, yellow, or red, depending on the user's country and inputted meat consumption. If the number is green, the user's CO2 emissions induced by meat consumption are below the average in his or her country by at least 25 percent. If the number is yellow, the user's CO2 emissions induced by meat consumption are within 25 percent of the average in his or her country. If the number is red, the user's CO2 emissions induced by meat consumption are above the average in his or her country by at least 25 percent.

4. After the user reads and takes in the information, the user has three options, to set a meat consumption goal, view their statistics on a graph, and view info and facts about their meat consumption.

5. If the user chooses to set a meat consumption goal, they could re-enter their meat consumption and the program would give the user a new value of their CO2 emissions and the difference between their initial value. This new consumption goal displays as green if it produces more than 25 percent less CO2 emissions than they inputted originally, yellow if more than 12.5 percent, and red otherwise.

6. If the user chooses to view their graphical analysis, they will be taken to a separate window, displaying a bar graph comparing their CO2 emissions per week of each type of meat, with the average in their country, and their new set consumption goals. This graph is generated by matplotlib.

7. If the user chooses to view info and facts about their meat consumption, in this screen, the user is also shown a blurb of text showing them whether they are below, at, or above the average meat consumption in their country and other facts about the relationship between meat consumption and climate change.

8. After the user is finished using these three options, the user can hit the 'Next' button to receive some facts about people's diets and their influence with climate change.

9. After reading this text, the user can hit the 'Restart' button and restart this process.

New Libraries

In terms of technical requirements, the pandas module being imported is specifically being used to import, read, and store the csv file containing the data. Pandas is a data analysis and manipulation module built for python that is widely used amongst programmers interested in data science and analytics. We decided to use pandas because it can import and read the csv file in minimal code (one line; a singular function call) making it efficient and easy to implement. `country_data = pd.read_csv(...filepath of dataset...)` is the line in which we used the pandas module. This can be found on line 64.

The TkInter module will also be used. The reason for choosing TkInter is that, firstly, it is built into the Python standard library. Furthermore, TkInter is quite efficient to use when compared to other potential frameworks. Although using TkInter may not result in the most modern-looking GUIs, our top priority is to code a GUI that is functional and allows the user to see their relevant statistics without obfuscation. That is why TkInter is the perfect choice for this project. While using TkInter, we not only implemented a whole user interface which opens up in a different screen, but we also focused on the small details and thus implemented things such as sliders (to input the number of times you at a certain meat in a week, and we used slides so that only numbers would be able to get inputted and allow our computations to work seamlessly), buttons, and even drop-down menus to make our program more interactive. Furthermore, we utilized TkInter's drawing function to draw arrows, which makes our data representation more intuitive.

We have also used the matplotlib library to create a graphical model of the users CO2 emissions based on their diet and compares it to the CO2 emissions of an average person in the country they reside in. The numpy library (comes with the pandas library) is briefly used as well to aid in creating the graph. We used this library to give the user a more visual representation of their results.

Obtaining Data sets and Running our Program

In order to run our project file successfully, a few python libraries must be installed, accompanied by the data set which our program extracts data from. Listed below are steps to download/install everything you need to run our

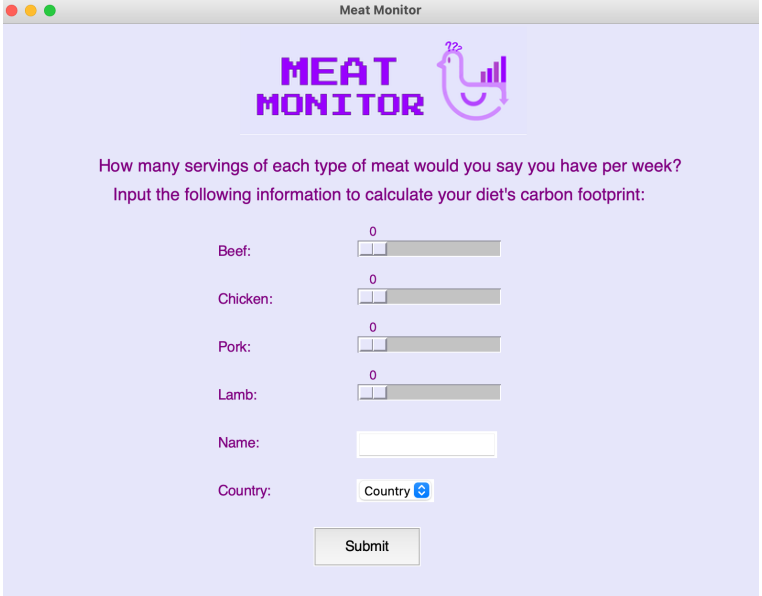
program. In the event that these instructions do not work for your device, we've included resources with a wide variety of troubleshooting methods pertaining to the library in question.

- **pandas library**
  - To install the pandas library, open the command prompt (in a Windows OS) or the terminal (in a Mac OS) and type `pip install pandas` into the command line interface.
  - If all goes well, you should get a `successfully installed numpy pandas` message
  - In the event that this is not working for you, this resource has a plethora of different methods you can try: `https://pandas.pydata.org/pandas-docs/stable/getting_started/install.html`
  - It should be noted that the numpy module also gets installed when pandas is installed.
- **matplotlib library**
  - Similar to installing the pandas library, to install matplotlib, open the command prompt (in a Windows OS) or the terminal (in a Mac OS) and type `pip install matplotlib` into the command line interface.
  - If all goes well, you should get a `successfully installed matplotlib` message
  - In the event that this is not working for you, this resource has a plethora of different methods you can try: `https://matplotlib.org/3.3.3/users/installing.html`
- **TkInter library**
  - The TkInter library is a built in python library, so you shouldn't need to install anything extra.
  - However, if our program is not running for you, that may mean that the version of TkInter you currently have on your computer may be outdated.
  - Please visit `https://tkdocs.com/tutorial/install.html` and follow the instructions for your operating system.
- **Datasets**
  - The link to download the dataset we used is:
    `https://ourworldindata.org/grapher/per-capita-meat-consumption-by-type-kilograms-per-year`
  - Once you have downloaded the dataset, you can save it to the same folder containing our `main.py` file.
  - You will then need to copy the file path of where the dataset is saved and paste it into `country_data = pd.read_csv(...filepath of dataset...)`. This is located on line 64.
- **Images**
  - In order for our program to run to it's true potential, you must also use the photos we used for our GUI.
  - Please download splash2.png and splash3.png from markus and save them to the same file in which `main.py` is in. Then please follow the steps below to ensure the correct images open up in the right windows.
  - On line 314, you will see a line that says `png2 = Image.open(...filepath of splash3...)`. Please paste the filepath of splash3.png into the designated place.
  - On line 441, you will see a line that says `png3 = Image.open(...filepath of splash3...)`. Please paste the filepath of splash3.png into the designated place.
  - On line 553, you will see a line that says `png4 = Image.open(...filepath of splash3...)` Please paste the filepath of splash3.png into the designated place.
  - On line 811 you will see a line that says `png1 = Image.open(...filepath of splash2...)`. Please paste the filepath of splash2.png into the designated place.
- **Restart**
  - If a user wants to use the restart button, the user must be using MacOS.

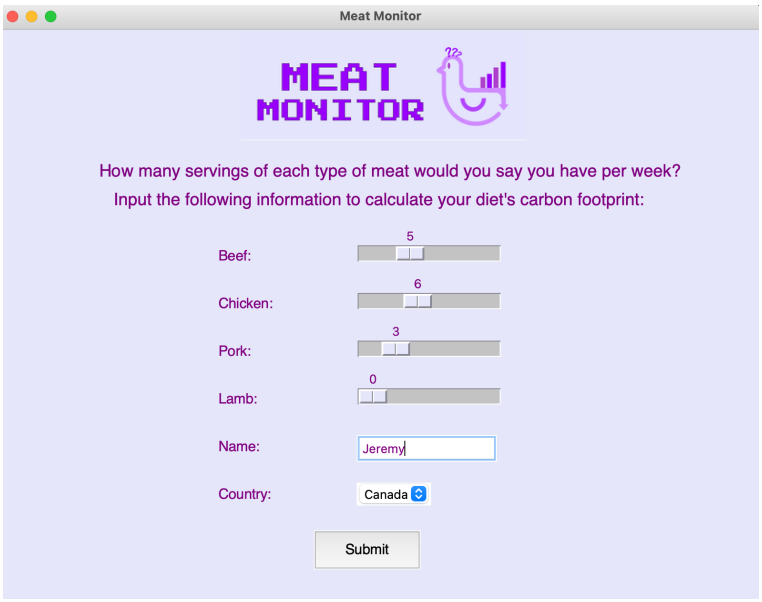
Program expectations

When running main.py, a new window should appear. You will see a loading screen like the following image:

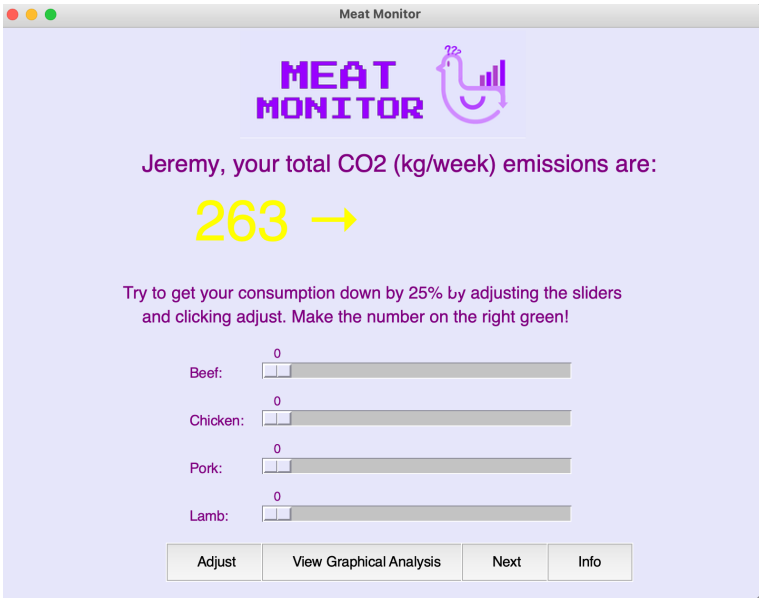
After a few seconds, you should be directed to the splash screen. This is the screen where the user inputs their meat consumption information, their name, and country of residence. This screen should look like the following image:



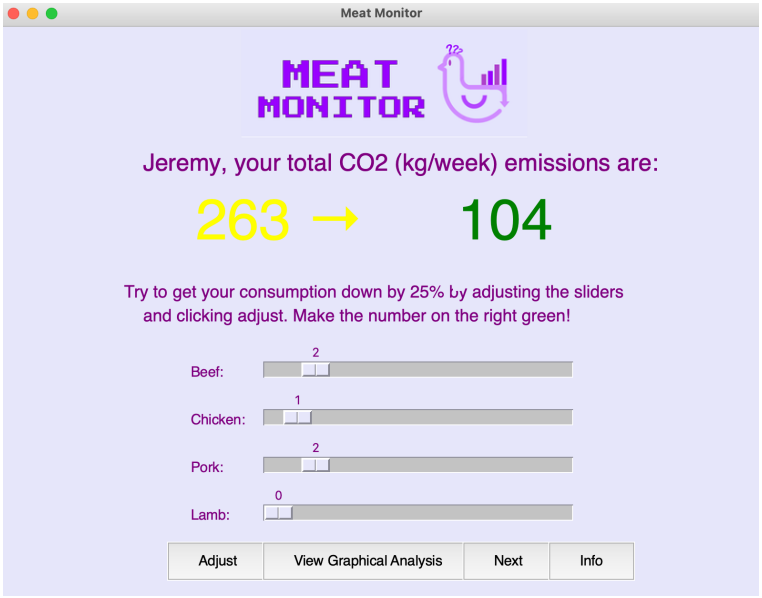
The way a user inputs their meat consumption in each meat type is interactive, where the user can click and drag the sliders to their desired meat quantity.



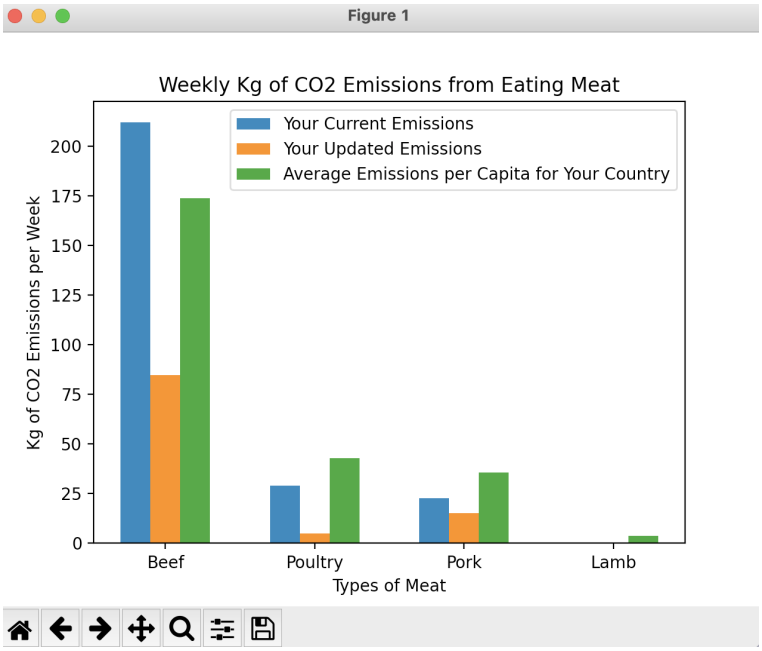
After the user clicks submit, they will be taken to a screen displaying the results from their input values. The displayed number will have different colours depending on your input, indicating whether you're above, below, or at the average consumption of your country.



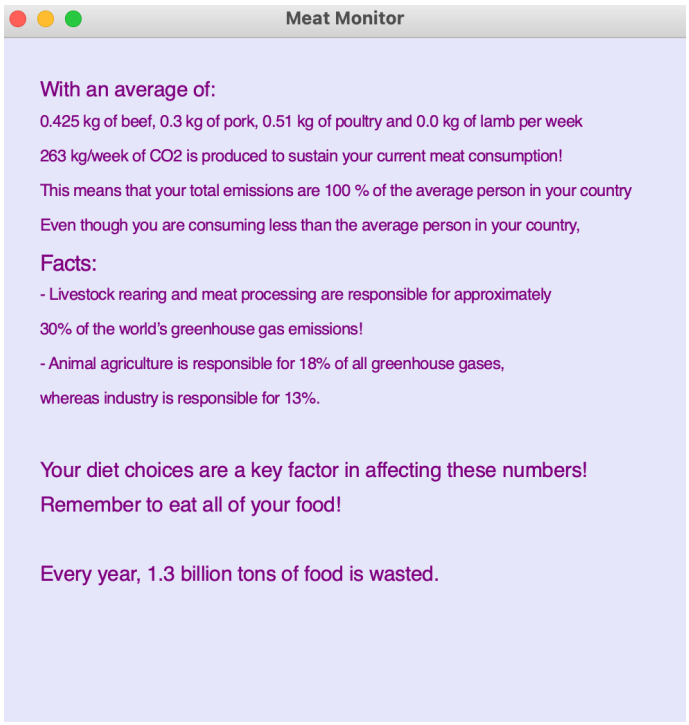
After the user views this page, they can re-adjust their consumption by using the sliders and clicking the button 'Adjust', which will give the user another value, which is the new CO2 emissions.



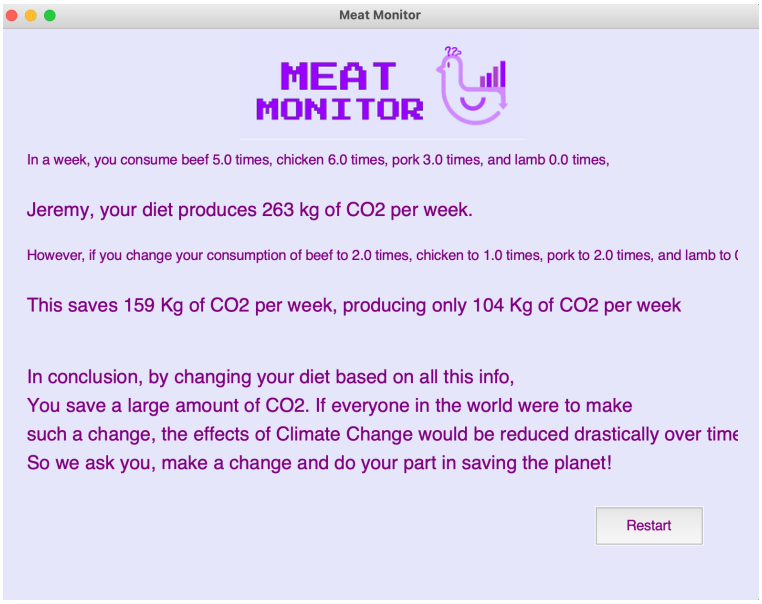
After that, clicking the 'View Graphical Analysis' button opens a new window to show a graph of the user's analysis.



The user can also press the 'Info' button after they receive their results, which takes the user to another window containing more specific details on their analysis.



The user can then click the 'Next' button to be directed to the final page which gives a summary of the user's input and analysis. After that, the user can click the 'Restart' button to head back to the splash screen.



Changes from our Project Proposal

In our final submission, we have included less variables in building our user profile when calculating their average meat consumption for their demographic. For example, we chose not to include age and wealth as a factor of our user profile. This is because we could not find reliable data sets to use that would aid us to produce meaningful results.

Discussion

The results of our computations has helped immensely in answering our research question, as we can discover how different users' diets' impact the environment in terms of the quantity of CO2 emitted to produce the meat that they consume. This is close to the answer of how the user turning vegetarian would impact the environment, however, we have not considered the CO2 would be emitted to produce other food items that would replace the meat. Our results are also able to take into account the location of the user, hence, includes variables in our research question.

Displaying graphs on our program was a limitation in this project. Due to the nature of the libraries, to produce the user interface (TkInter) and the graphs (matplotlib), their interactions with each other are not ideal for displaying a graph like how we wanted. Orignally, we wanted to have graphs embedded into the same window as the starting window. Instead, we had to result to creating a new window by the user pressing a button in order for the user to see the graph. Although at first we had issues with implementing the graph with the TkInter library, we were able to find a viable solution.

From our current product, we can only tell the user how many kilograms of CO2 is required to produce the amount of meat they consume and compare it with their country's per capita average. This, in the grand scheme of things, is not enough to fully answer our question of the environmental impacts of a person turning vegetarian. We don't compare the emission values due to meat consumption to the total global emissions. In addition, we do not know how the reduction of meat consumption will qualitatively impact the word's environment. For example, we don't know how many kilograms of CO2 it would take to slow down climate change, preserving many icebergs in the north and south pole. So, to sum things up, we do know that changing ones diet so that they eat less meat does positively affect their carbon footprint (as in their carbon footprint goes down), but we do not know the extent to which this is true.

In addition to these improvements to be made, we could further enhance the specificity for the meat consumption profile for the user. For instance, we could include age, wealth, and many other factors that affect meat consumption. This would entail finding more data sets that pertain to our areas of interest.

As users are inputting their information, we could use this program as a means to collect meat consumption data. Therefore, this data can be further used in other programs or research. To even further explore our research question, we could have explored the CO2 emissions related more to the user, instead of the animal. To elaborate, we would look at factors such as how much CO2 emissions are released when the user drives to the grocery store, busses to the grocery store, etc. In essence, we would be looking at a even greater amount of detail.

Overall, we were able to create a program that would display the impact of a person's meat consumption on the environment by displaying the CO2 that the person's diet indirectly produces, which helps answer our research question. However, more variables could be included in our computations in the future to make our program closer to reality. Further progressions of this program could also be in the form of data collection and expanding our computational results to qualitative impact on the world's environment.

Some tips

While running our code on several different computers to ensure compatibility, in one instance, on a Windows OS PC, we got an error in the console saying something along the lines of `_tkinter.TclError: image "pyimage2" doesn't exist`. If this error occurs for you as well, the wrap around we discovered is that if you remove `import matplotlib.pyplot as plt` from the top of the file and move it to right under the docstring for the `def graph` function, the program will run correctly.

```
def graph() -> None:
    """Creates graphs of info using matplotlib
    Preconditions:
        - user1.total_emissions_list != []
```

```
    - user1.total_country_emissions_list != []
    - user1.new_total_emissions_list != []
"""
import matplotlib.pyplot as plt
if user1.total_emissions_percentage <= -25:
...
```

References

References

Data to Fish, 21 Dec. 2019, datatofish.com/import-csv-file-python-using-pandas/.

Eatz, Jane of Green. "Food's Carbon Footprint." *Green Eatz*, 27 Sept. 2020, www.greeneatz.com/foods-carbon-footprint.html.

Pandas, *Pandas*, pandas.pydata.org/.

"Per Capita Meat Consumption by Type." *Our World in Data*, United Nations Food and Agricultural Organization (FAO), 2020, ourworldindata.org/grapher/per-capita-meat-consumption-by-type-kilograms-per-year?time=earliest..2017.

"Plotly Python Graphing Library." *Plotly Python Open Source Graphing Library*, plotly.com/python/.

Poore, J., and T. Nemecek. "Reducing Food's Environmental Impacts through Producers and Consumers." Science, vol. 360, no. 6392, Science, 2018, pp. 987–92.

"Sources of Greenhouse Gas Emissions." *EPA*, Environmental Protection Agency, 9 Sept. 2020, www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions.

"TkInter." TkInter - Python Wiki, *Python*, wiki.python.org/moin/TkInter.

Poore, and Nemecek. "Greenhouse Gas Emissions per 100 Grams of Protein." Our World in Data, 2018, ourworldindata.org/grapher/ghg-per-protein-poore.

Arnarson, Atli. "Lamb 101: Nutrition Facts and Health Effects." Healthline, 2019, www.healthline.com/nutrition/foods/lambnutrition.

Arnarson, Atli. "Pork 101: Nutrition Facts and Health Effects." Healthline, 2019, www.healthline.com/nutrition/foods/pork.

Frey, Malia. "Chicken Breast Nutrition Facts and Health Benefits." Verywell Fit, 2020, www.verywellfit.com/how-many-calories-in-chicken-breast-3495665.

Frey, Malia. "Steak Nutrition Facts and Health Benefits." Verywell Fit, 30 June 2020, www.verywellfit.com/the-best-ways-to-eat-steak-on-a-diet-3495219.