CSC110 Project Proposal: Comparison of Real Temperature Values to Predicted Temperature Values for Different Representative Concentration Pathways

Can Yildiz, Alamgir Khan, Eren Findik, Giancarlo Sass Ramos

Friday, November 6, 2020

Problem Description and Research Question

The natural atmospheric greenhouse effect is what makes life on earth possible. However, due to human activities—mainly the burning of fossil fuels—global emission of greenhouse gases (such as carbon dioxide, methane, etc.) has increased significantly over the past few years, and as more of these gases become trapped in the Earth's atmosphere, the average temperature of the world rises. This is because greenhouse gases are transparent to visible light but opaque to infrared radiation, so sunlight can travel through them but heat is blocked. Climate scientists observe this effect (among others) by collecting data (such as temperature, precipitation, etc.) and construct models to predict future temperatures for various Representative Concentration Pathways (abbreviated as RCPs). RCPs make predictions of the concentration of greenhouse gases in the atmosphere by capturing the effect of human activities and future trends on greenhouse gas emissions. The four RCPs range from very high (RCP 8.5) through to very low (RCP 2.6) future concentrations.

One such climate model is the BCCAQv2. This was developed by the Pacific Climate Impacts Consortium and aims to rectify the bias in daily precipitation series that is found in other popular climate models. This creates data that more accurately matches historical records. Essentially, this preserves the precipitation change signal while downscaling daily climate predictions relating to temperature and precipitation. This model is a mathematical equation attempting to estimate the climate trend through interactions within the atmosphere. In order to develop well-structured and long-term plans to deal with climate change, it is crucial that these estimates be as accurate as possible. For this reason, we have decided to test the accuracy of this model by comparing actual values of temperature recorded in Toronto against those predicted by the BCCAQv2 climate model, for various RCPs:**how do real temperature values compare to those predicted by the BCCAQv2 climate model?**

Dataset Description

We will be using the following two datasets (which we have edited slightly for the purposes of this project).

Dataset 1 is a csv file that is a collection of the monthly summaries of the averages and extremes of temperature for each month from the year 2003 to the year 2020 for the city of Toronto. This dataset initially contained other data such as precipitation and snowfall levels but for simplicity's sake we have removed them.

Sourced from Environment and Climate Change Canada (https://climate-change.canada.ca/climate-data/#/monthly-climate-summaries).

Sample data contained in the dataset:

STATION_NAME	PROVINCE_CODE	LOCAL_YEAR	LOCAL_MONTH
TORONTO CITY	ON	2003	1
MEAN_TEMPERATURE	DAYS_WITH_VALID_MEAN_TEMP	MIN_TEMPERATURE	MAX_TEMPERATURE
-6.626666667	30	-20.4	5.5

Dataset 2 is also a csv file containing data of the city of Toronto but is a collection of predicted annual mean temperature (along with a range of low and high mean temperature) calculated using the BCCAQv2 climate model of each RCP (2.6, 4.5, and 8.6) for the years 2003 to 2020. This dataset initially contained longitude and latitude values of the location but as these are unnecessary (since they match the other dataset) we have removed them for simplicity's sake.

Sourced from Climate Data Canada (https://climatedata.ca/).

Sample data contained in this dataset:

Date	RCP 2.6 Range (low)	RCP 2.6 Median	RCP 2.6 Range (high)	RCP 4.5 Range (low)
01-01-00	8.5	9.14	9.92	8.5
RCP 4.5 Median	RCP 4.5 Range (high)	RCP 8.5 Range (low)	RCP 8.5 Median	RCP 8.5 Range (high)
9.15	9.9	8.5	9.15	9.89

Computational Plan

We have two data files to transform: actual monthly temperatures (Dataset 1) and predicted annual temperatures (Dataset 2).

Firstly, we will write a function that uses csv.reader() to read the data in Dataset 1. Using this function, we will store the values from this dataset into a list of our predefined data class which uses the headings from the dataset as attribute names. For example, our data class for the following sample from Dataset1 could look like:

STATION_NAME	PROVINCE_CODE	LOCAL_YEAR	LOCAL_MONTH
TORONTO CITY	ON	2003	1
MEAN_TEMPERATURE	DAYS_WITH_VALID_MEAN_TEMP	MIN_TEMPERATURE	MAX_TEMPERATURE
-6.626666667	30	20.4	5.5

```
@dataclass
class MonthlyTemp
local\_year: datetime.year
local\_month: int
mean\_temperature: float
min\_temperature: float
max\_temperature: float
```

We would then need to use this list of MonthlyTemp in a nested loop (at least one of which would need to be indexed) to mutate the mean temperature for each month to a new list so that we can use the sum of this list to calculate the actual annual mean temperature for each year. Once these annual temperatures are calculated they would be mutated to a final list that will be used later in our program.

Secondly, we will write another function that uses csv.reader() to read the data in Dataset 2. However, this time, we will map the values from the dataset to a dictionary – The given date in the datetime.date format will be mapped to a dictionary which maps RCP values to a list containing the low, median, and high values of that given RCP (The format is basically Dict[datetime.date, Dict[str, List[float]]]]). An example of what this might look like using our sample data from Dataset 2 is shown below.

Date	RCP 2.6 Range (low)	RCP 2.6 Median	RCP 2.6 Range (high)	RCP 4.5 Range (low)
01-01-00	8.5	9.14	9.92	8.5
RCP 4.5 Median	RCP 4.5 Range (high)	RCP 8.5 Range (low)	RCP 8.5 Median	RCP 8.5 Range (high)
9.15	9.9	8.5	9.15	9.89

would convert into:

```
{
datetime.date(2000,01,01):
{ \''RCP 2.6": [8.5, 9.14, 9.92]
\> \RCP 4.5" :[8.5, 9.15, 9.9]
\RCP 8.5": [8.5, 9.15, 9.89]
}
}
```

Finally, using plotly Line and Scatter Plots, we are going to plot a multiple line graph with these predicted values with the dates on the x-axis and the RCP values on the y-axis (each RCP level having its own unique color). Then, we will also plot the actual annual mean temperatures calculated from Dataset 1 onto the same graph with a new color. Additionally, the user would be provided a table that displays the percentage difference between the actual mean temperatures and the ones predicted for the varying RCPs. The user will also be able to check individual percentage differences through clicking on a point on the graph – this will be provided by a click event.

As an extension to this project, we could take data from more cities in Canada and do the same process for them. However, this time we could also create 2 maps that compare the actual and predicted data by colors. The temperature values will be assigned a (r, g, b) value and that color will be used to paint the region on a map of Canada. The regions will be initially white with black borders–like on the example: https://1.bp.blogspot.com/-llh8XcxIAU0/W2rw2m?_n8ZI/AAAAAAAACR8/nPVSrFXKiaA8T7YrtcAFNzNTsC-1T9b-wCLcBGAs/s640/Blank-Outline-Map-Of-Canada-Provinces-challenge-printable-canada-map-canadian-provinces-outline-maps-gif-country-map-fwdpower-.jpg. We will first import the picture and then take a list of points (because some regions have islands) from the region that we are going to address and fill it with color using fillcolor() from turtle library (if we are allowed to use it). This process will be done by using a for loop to fill every region.

References

Higher Temperatures. (2017, September 5). Global Climate Change. https://archive.epa.gov/climatechange/kids/impacts/signs/

James, J. J. (2015, January 15). How do greenhouse gases trap heat? Socratic. https://socratic.org/questions/how-do-greenhouse-gases-trap-heat#: :text=Greenhouse

What are the RCPs? (n.d.). [Infographic]. Coastal Climate Change Infographic Series. https://coastadapt.com.au/sites/default 117-NCCARFINFOGRAPHICS-01-UPLOADED-WEB

Line Charts in Python. (n.d.). Plotly. Retrieved November 6, 2020, from https://plotly.com/python/line-charts/

Scatter Plots in Python. (n.d.). Plotly. Retrieved November 6, 2020, from https://plotly.com/python/line-and-scatter/

Interactive Data Analysis with FigureWidget ipywidgets in Python. (n.d.). Plotly. Retrieved November 6, 2020, from https://plotly.com/python/figurewidget-app/

Monthly climate summaries. (n.d.). Climate Data from Environment and Climate Change Canada. Retrieved November 6, 2020, from https://climate-change.canada.ca/climate-data/#/monthly-climate-summaries

Variable Data (Mean Temperature). (n.d.). Climate Data Canada. Retrieved November 6, 2020, from https://climatedata.ca/do

Climate Data Canada. (n.d.). About. Retrieved November 6, 2020, from https://climatedata.ca/about/