CIS4930 Term Project

Miles Brosz, Natalee Sama, ...

Florida State University

CIS4930; Special Topics

Mithila

4/11/25

Climate Change Impact Analyzer Python Project

This project pools data from the “NOAA National Centers for Environmental Information (NCEI)” weather API, in order to successfully project, predict, and normalize precipitation data for the month of August (chosen given that it is the rainiest month), across the span of 5 years (inclusive of 2020 - 2024). The data was taken from the Tallahassee station in metric units. From this data we employ the “LinearRegression()” from the “Sklearn” library, in order to predict the amount of rainfall in the same month for any number of following years (based on data from our sample size). We present such data in mediums such as: Scatter plots, Pie-Charts, and Bar-Graphs. The Project is separated into main files: “algorithms.py”; which fittingly sees implementation of the algorithms we use for data prediction and detecting anomalies/miscellaneous skews from the data trend; “DataCollection.py”, being responsible for making the API request and sending the data to “/data/climate\_data.json”; “cli.py”, our command-line interface, which parses command-line inputs and loads and processes the data in the aforementioned “climate\_data”, alongside running the code for “visualizer.py” which interprets and displays the data in such called graphs. Extra functionality is also included with the HTML presentation as an optional alternative to the command-line interface, displays and navigation were made using Flask.

# Functionality/Algorithms

It should be noted that because we chose to present the data in the same time frame across different years, this would make other metadata ambiguous. Considering this, we only utilized the data elements that would be variable, the year and the day. For the scatter plot for instance, because of this, representing these elements as a raw datetime object, led for weird uniform clusters along the x axis, as they would all translate to the month of august as a progression in the year, with gaps in the data between. To resolve this, we opted to represent the day as a floating-point variable, and incrementing in 1/31 (31 for each day of August) with the whole number being the year. In doing this, a day like 2024, August 5th is represented as the number 24.1613... or 24 + (5/31). This way the data is nicely separated, and we have x-axis markers for the years 20,21,22, and so on. The y-axis is only changed in that we normalized the data for the amount of precipitation, and this value is represented in this axis.

For prediction, we opted to use the built in linearRegression() method in sklearn. It is easy to grasp conceptually over implementing our own prediction method and it intuitive. The linearRegression just calculates a slope over a set of data, and we feel that this would reflect a strong enough predication with reasonable margins of error for determining future precipitation of years post 2024, as it relates to the trends found in our sample size. This is the easiest way to capture potential climate change increases.

Detecting anomalies, ....

Random clustering, ....

Custom machine learning algorithm, ...

“2. Innovative Algorithms (15 points)  
- Develop a custom machine learning algorithm to predict future trends of climate parameters  
(e.g. temperature, precipitation etc)  
- Implement a novel clustering algorithm to group regions with similar climate patterns  
- Create a time series analysis algorithm to detect anomalies in climate data”