*Project Goals*

The goal of this project was to receive some insight on the traffic conditions in Ann Arbor, and how they correlate with the weather conditions. Using OpenWeather API and TomTom API, we hope to use our newfound Python skills to gather data from the APIs, create databases in SQLite3, edit those databases, and calculate something from the gathered data. We will hopefully get information on live traffic data and incident data and see how the weather conditions (temperature and descriptions) influence that data. Once we have gathered this data, we hope to use Matplotlib to create graphs of the yearly weather in Michigan, the yearly incident data in Michigan, and then a graph showing the correlation between the two.

*Achieved Goals*

Unfortunately, we were not able to use the incident data API from TomTom, nor could we get the previous year’s weather or traffic data because the API did not store a year’s worth of data. Instead, we pivoted our project to get data from 12-04-19 to 12-11-19 on live traffic and weather data. We were able to collect data on traffic conditions, including speed, travel time, road closures, and the confidence of the predicted traffic data. We were also able to collect weather data including the current temperate, the maximum temperature for that day, the minimum temperature for that day, the general weather conditions, and then a description of those weather conditions. From that data, we were able to calculate the average confidence of traffic conditions, the average speed, the average temperature, and average weather condition.

To better understand this calculated data, we created four visualizations. The first is the average confidence level in traffic reports given the weather conditions (bar graph), which showed that the confidence was highest when snowing, second highest when cloudy, and least confident when raining. The second visualization used a double line graph to show the average temperature versus the average speed per day from 12-4-19 to 12-11-19. This highlighted that when the temperature was higher, so was the average speed, and when the temperature was lower, so was the average speed. The third visualization utilized a line graph with two subplots, due to the y axis values being so far apart and graphed the temperature versus the average confidence in traffic conditions per day. This graph showed that when the temperature was higher, the confidence in traffic predictions was lower. The final graph, a pie chart, showed the average weather conditions from 12-4-19 to 12-11-19. It reported that 75% of the time, the weather reported clouds, 12.5% of the time it reported snow, and the other 12.5% of the time, it reported rain.

*The Problems Faced*

One of the first problems we faced in this project was that we were unable to get the live incident data from the TomTom API.This was due to the fact that when we input a versionNumber (as we were required to according to their documentation for the API), it would cause an error. We were also unable to retrieve the past years live traffic data, because the API did not store this data. However, we were still able to complete the project without either of these data points.

In order to collect the live traffic data, we had to create a bounding box, an area for the traffic data to be collected between, in Ann Arbor. We chose coordinates on Washtenaw Ave, Packard & Hill, W. Stadium Blvd, Saline Road, and N. Maple Road. However, because of the way that the TomTom API is set up, we had to convert the latitude and longitude to the World Geodetic System (1984) using an online converter.

* Had to convert the lat/long on map using (website)
* Had to convert the datetime to be the same after the data was gathered
  + This was hard because the code ran from top to bottom, not all at the same time, so the time stamps were different, we ended up just collecting the data per day rather than per hour or second so that when we did our database join we would be able to get matches
* Decided to make pie chart out of weather descriptions, so had to calculate the percent of each weather description using a counter
* For the double line graph, we initially had it as a single line graph with two lines, but we had to separate them because the values of the confidence and the values of the temperature were too different

*Instructions for running the code*

1. Uncomment to collect weather data (line 47/48)
2. Uncomment to collect traffic data (line 111/112)
3. Uncomment to clean 4 data tables (line 178/179) after getting all data
4. Uncomment to create average calculations table (line 360)
5. Uncomment to write data to text file (line ?)

*Documentation for each function written (input/output for each)*

*Documentation for resources used*

1. The visualization that you created
2. Instructions for running your code
3. Documentation for each function you wrote. This includes the input and output for each function.
4. Documentation of all resources that were used

* Mat plot lib pie chart website
* Conversion for lat long website
* Conversion for date time website
* Tomtom data API information website
* Openweather API information website
* Mat plotlib double line chart

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Issue Description** | **Location of Resource** | **Result (did it solve the issue)?** |
|  |  |  |  |