Data Science – Project **IoT using GPS, Hill Detection and Data Visualization**

This project can be done in groups of one or two.

The Internet of Things (IoT) involves the growing trend of having multiple small devices that all collect small amounts of information. Each IoT device uses a small processor. Most of these devices do not even have an operating system on them.

Arrow corporate hires many people to integrate IoT devices which help control power for companies. The buildings are scattered with small IoT devices. The devices monitor heat, light, and water usage, people presence, ventilation, sunlight, and cooling needs. By combining information from many sources, they form a larger picture which tells more of a story then any individual device can. They are now promising to save money for customers, as a blanket statement. This level of IoT is beyond having a computer in your toaster, it is called Industrial IoT, or IIoT for short.

This semester the task is relatively simple. Given a single GPS file, detect and visualize where the car needed to go uphill.

You are provided with a set of GPS data tracks in which Dr. Kinsman drove. Given the GPS data please achieve the following goals:

1. Write a program to convert the GPS data into KML data that can be viewed in Google Earth.
   During the process find a way to clean the data by filtering out and ignoring any redundant or erroneous data.
   For example:
   a. If the vehicle is parked, you do not need multiple data points at that same location.
   b. If the vehicle is traveling in a straight line going down-hill, you can ignore some points on the line.
   c. If the vehicle is traveling in a straight line going up-hill, you can ignore some points on the line.
   d. The Arduino sometimes burps and writes two GPS sentences to the same line of the data file.
      You need to detect and ignore these anomalies. Otherwise it looks like the car jumps to the other side of the planet.
   e. The Arduino sometimes loses its mind, and starts recording GPS values that jump all over the place.
      This especially happens if it loses connection to the antenna. You will need to ignore meaningless junk.
   f. At some point the antenna broke, and the GPS device became less accurate.
   g. When the vehicle first starts up, the GPS device is not moving. Do not worry about the data points when the vehicle has not started moving yet.
   h. When the vehicle parks, the GPS device stops moving. Do not worry about these non-moving data points at the end of a drive.
   i. The "heading" or "direction" information on the GPS is only valid if the car is moving. If the car stops at a stop sign or a traffic light, the "direction" is meaningless. So, you can only use this information if the speed of the vehicle is above some minimum speed.
   j. You need to convert the GPS sentences. You cannot rely on the comments in the code.

   There is a maximum limit to the number of points you can include in any path, so you might need to split one GPS track into multiple paths after 10,000 or 20,000 points. (I don't know what the limit is.)

(continued)

2. Finding Hills:

   Read in all of the given GPS files at once.

   Use any data in the RMC or GGA sentences of the GPS file.
   Do not use any debugging lines that Dr. Kinsman put in the GPS file if they are present. (lng, lat… in the file.)

   Go look at the example KML files from the lectures.

   When the car is going uphill, switch the color of the path to RED.
   When the car is going flat or downhill, switch the color of the road to green.

3. Resources:

   There are lots of web pages about "Python Pynima GPS Visualization". Search the internet for examples. They are out there.

   https://towardsdatascience.com/simple-gps-data-visualization-using-python-and-open-street-maps-50f992e9b676

   https://thatmaceguy.github.io/python/gps-data-analysis-intro/

   https://ozzmaker.com/using-python-with-a-gps-receiver-on-a-raspberry-pi/

   https://blog.jetbrains.com/pycharm/2022/09/visualizing-geospatial-data-in-python/

   https://stackoverflow.com/questions/62645722/track-gps-data-for-a-certain-time-with-python

   https://www.youtube.com/watch?v=bDOoT4nu_oE

(continued)

4. Write-up your results, in a report. This should be about 5 to 7 pages.
   Put the following information in it. (Copy the questions to your work.)

   a. Who worked in your group.

   b. Introduction – what did you do?

   c. Background section (three or four paragraphs), with *references*.
      Discuss the background of GPS: When did GPS come about? How many satellites are there? What other relevant details should we know about GPS? What else did you find out that is interesting about GPS?

   d. List references in your paper. Where did you get code snippets from?
      What python libraries did you use?

   e. Ethical Analysis
      Google has access to all of this information. How could this be misused? Is it ethical for you do analyze this data from Dr. Kinsman? What ethical issues might you run into?

   f. Data Cleaning:
      What kind of data cleaning did you need to do?

   g. Problem definition and strategy:
      How did you define uphill? How did you define downhill? When does it change?
      How do you avoid sudden fluctuations?
      What problems did you need to overcome? What strategy did you use?

   h. Implementation
      Describe how you wrote your program. What program pattern did you use?
      Did you hold all the information in memory at once? Did you parse everything at once?

   i. Create an output *.KML file that has in it:
      i. Roads where Dr. Kinsman went significantly uphill, marked in RED.
         You have to decide what "significantly" means. Define your terms.
      ii. All other roads marked in YELLOW.

   j. A screen capture of image of your results in Google Earth.
   k. A conclusion of what you learned, and how this might be useful for some commercial application. What were you especially proud of? What went well? What did each of you learn?

(continued)

5. **Turn into the dropbox, ONE OF YOU should submit:**

    a. IN THE PAPER FOLDER PUT → A project report, as described above.

    b. In the CODE FOLDER PUT →
        i. Your output *.KML file.
        ii. Your code files.
        iii. Any other supporting files.