Fun Exploratory Python Problems
RAID Problem Set 1

Instructions :

- A google colab file has been shared with you which contains helper code for each of the questions.
- Read questions if you need a particular input of a file a google drive link has been shared at the end of this file which will contain all the files that you can download and upload on google colab.
- To start solving create your own copy of the google collab file by file>save a copy in drive
- Rename the google colab file to whatever you want.
- Fill in the blanks in the google collab file to solve the questions.
- You can download the google collab file by going to file>download>download .ipynb.
- The problems are intended to be tough (for beginners). However each one of them is solvable using googling skills and extensive research!
- The problems are very indicative of your problem solving abilities. If you are able to solve a majority of these problems, you are a good thinker! Programming/AI and beyond.
- If you have never heard of one of these concepts, google! Most of these topics (Image Processing/Linear Regression/Fourier Transforms) have 10 minute YT videos that give you enough information to solve the problems. I have tried to add some resources while I encourage you to find the rest.
- Remember, most problems statements you will see during Inter-IIT, Robocon, Hackathons, Research and your future Job are not solved yet. You need to gather information, do research and implement a solution. That is what engineers do.

Helper code google colab link -
https://colab.research.google.com/drive/1x2XdOSgXrcZ5t6ZmIZ6T5qXyv-mMaUnI?usp=sharing

Q.1 A way to solve Ordinary Differential equations using Euler's method is very popular, especially in computer systems. However it is cumbersome and takes some time. Moreover it does not work on discontinuities very well.

There exists libraries that can solve ODEs as well. Scipy is one of them. Visit the following links and try to figure out how to solve ODEs using Scipy. If these links are not enough, feel free to use any other resource like youtube/google etc.

https://apmonitor.com/pdc/index.php/Main/SolveDifferentialEquations
https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html

Finally solve the following ODEs using the scipy package and show the results using a matplotlib plot.

(i) $\frac{dy}{dx} = e^{x}$ where y = 1 when x = 0 (ii) $\frac{dy}{dx} + x^{2} = x$ where y = 0 when x = 0

(iii) (Bonus) Try to implement a basic ODE solver by scratch (Do not use scipy) and try to plot the ODEs in (i) and (ii).

Q. 2 Fourier Transform lets us transform a signal in the time domain to the frequency domain. We can then view the frequencies that make up that sound/signal.

In musical instruments which produce a sound wave, there exist 12 fundamental notes (A A#/Bb, B, C, C#/Db, D, D#/Eb, E, F, F#/Gb, G, G#/Ab). Each note has a fundamental frequency associated with it.

This website lists notes and their frequencies
https://pages.mtu.edu/~suits/notefreqs.html

When each of these 12 notes are played on an instrument there exists a fundamental frequency and higher frequency harmonics. When the note A is played, there is a spike at the 440 Hz mark on the fourier transform and higher frequencies. They follow a pattern i.e. The consecutive frequencies are equal to $2^{n}f$ where f is the fundamental frequency. For example if we play the note A (440Hz) then we see a spike at 440Hz and then harmonics at 880Hz, 1760Hz and so on.

A numpy array containing a sound wave is provided in the below google drive link. Download the .npy file and open it inside of a python file. Use the link below to find how to open .npy files
https://www.geeksforgeeks.org/numpy-load-in-python/
Plot the wave using matplotlib.

Now apply a fourier transform on the numpy array and plot it on the graph. Make a note of where the frequencies spike. Using the website provided above can you guess what note is being played? Write your answer as a comment at the end of the python file.

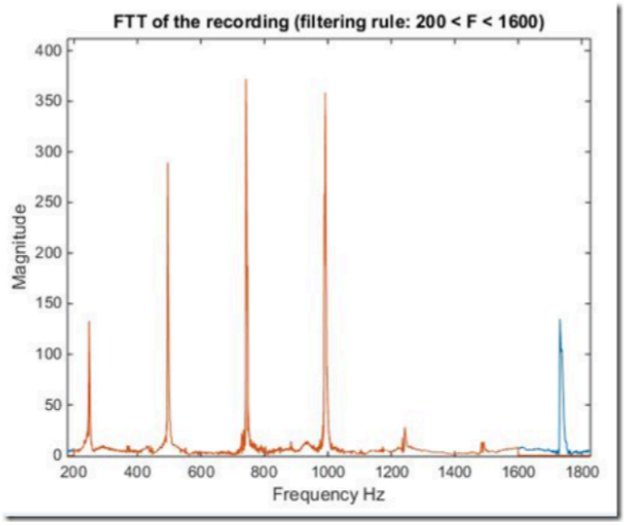
For example, for a fourier transform shown below, we can see spikes at around 250Hz, 500Hz, 750 Hz and 1000 Hz. These frequencies match with B3, B4 and B5 notes and hence the sound

wave contains the note “B”. Perform a similar analysis.

Fourier Transform of a guitar tone

- Frequency of B3 is 246.94Hz
- Frequency of B4 is 493.88Hz
- Frequency of B5 is 987.77Hz

All of the frequencies mentioned above are present in the fourier transform of the guitar’s B note Along with some timbre and harmonic frequencies which give the characteristic sound to the guitar.



Q.3 (Bonus) Given below is a bird's eye view image of farmland. Different crops growing in different areas. Can you find a method to generate an image which shows only the boundaries of different plots of land?

(Hint: It is one out of Blurring kernels/Sharpening Kernels/Edge detection kernels)

The image given below is also shared with you in the google drive folder so you can download the image from the google drive folder.

Q. 4 (Bonus) Linear regression

1. Load the same boston data.
2. Apply Linear regression (using only ‘RM’ as feature)
3. Report mse on test data by varying train-test splits.
4. Plot various lines generated by different models (in 3) in a single plot.
5. Try np.polyfit and fit a degree 2 equation and plot it, and comment on the differences between linear one and this one.
6. Also report mse on test data by using all the available features for linear regression.

Q. 5 Object Oriented Programming allows you to create your own data types.

Create a class representing a 2D vector called “Vector” and add the following features to it.

(i) Two attributes x and y representing the x and y components of the vector.

(ii) Write a __init__ method to initialize the vector object that accepts two inputs x and y. (NOTE: You might have to include self in the function definition)

(iii) Overload the “+” operator in python by defining a method “__add__”. Write the logic to add two vectors and return the new vector.

(iv) Similarly overload the “*” operator in python by defining a method called “__mul__”. Define this to return the dot product of the two vectors. (In vector algebra multiplying 2 vectors can mean dot product or cross product. Here we just want the dot product).

(v) If you print the vector object now you will get the name of the class and the memory location of the object in hexadecimal i.e. 0x0010020. Define a method called “__str__” and return a string in the format “5i + 9j” if the vector has x = 5 and y = 9. (You can use string formatting here). Print your vector variable now. What is the difference?

(vi) (Bonus) Design more helpful functions and methods to help users of your class with utilities and features. Can you add a normalization function? Can you add scalar multiplication? Think and design as many cool features as you can.

Resources to image/npy files are provided in the google drive link below - https://drive.google.com/drive/folders/1mz_9qILx5GLRu5DR7OH_b51uvpLRJLlJ?usp=sharing