Activity 1: 16ers

Team 16: Anirudh Gattu, Jong Yoon Kim, Cassandra Marshall, Grace Pfohl

Instructions: Use the following template for your submission.

Submissions :

1. Report - pdf format. Name it TeamNumber_SensorDataAnalysisActivity (e.g. Team1_SensorDataAnalysisActivity)
2. A separate Zip folder - with code, jupyter notebook (if you are using it), and data (submit modified GitHub folder). The jupyter notebook should be named TeamNumber_SensorDataAnalysisActivity. You should answer all the questions in this report.

Please make sure the report pdf is separate and NOT in the zip folder, otherwise it gets very hard for us to grade.

TASK 1 [15 points - 3 points each]

a. Open walking_steps.csv collected and check relative_time column. What is the sampling rate for the accelerometer data?

b. Time (s)

- *Sampling Rate = Sampling Frequency = 1 / time (seconds)*
- *1 / (3.817855300E-2 - 3.570844900E-2) = 1/ 0.2470104E-2 = samples / second (Hz)* ≈ 405 Hz
- Total Average Sampling Rate = 28591/(7.065676004E1-3.570844900E-2) ≈ 405 Hz
- Note: While this value is not in the reasonable range for the rubric, we know that some Android phones have higher sampling rates for their accelerometers.

b. Is the sampling rate stable for the file walking_steps.csv? What is the variation? If it is not stable, explain what could be affecting the stability of the rate? (Hint: https://pure.au.dk/ws/files/93103132/sen099_stisenAT3.pdf, no need to thoroughly read the whole paper, this is just to lead you to think in the right direction. Looking at the first two pages should be sufficient)

- *The sampling rate appears unstable because the time between samples changes over time (sampling rate changes over time).*
- Here are some of the sample rates with calculations:
  - 1.000522939E1 - 1.000275923E1 = 0.247016 seconds
    - Sampling rate = 4.049 HZ
  - 1.000769954E1 - 1.000522939E1 = 0.247015 seconds
    - Sampling rate = 4.049 HZ
  - 3.543782051E1 - 3.544029067E1 = 2.47016E−3
    - 404.8 samples per second or Hz
  - The variation is… 1.0777025E−1

- *The sampling rate is semi-stable, but has some variation. After reading through the paper, here are some possibilities:*
  - *Devices come with different hardware and operating system characteristics that can lead to variations in sensor responses.*
  - *The default and supported sampling frequencies for sensors like accelerometers can vary across devices.*
  - *Real-world use cases, such as multitasking or high I/O loads, can lead to unstable sampling rates because the mobile OS may not consistently attach accurate timestamps to measurements.*
  - *Modern sensing strategies that prioritize power conservation, such as dynamic duty-cycling, can introduce further irregularities in sampling rates.*

c. How does your understanding from part (a) affect your data processing pipeline?

The sampling rate informs us of how detailed the accelerometer data we have to work with is. Because the sampling rate is very high and variable, we know that there may be noise in our signal and that we may have more data than we need at certain points. This foreshadows that our data processing pipeline will need to include some preprocessing of the data in order to smooth the signal and reduce unnecessary noise when trying to recognize human activity.

d. How can we make the data easier to work with (hint: re-sampling the data)? Choose one method of your choice and briefly explain how the math works.

One method of resampling is by averaging. We could use some type of moving average based on a predefined window and average their values to get a single data point for that interval. This reduces noise and gives a consistent time frame for analysis. The math is simple: Resampled data = $(a1 + a2 + \ldots an)/n$
We could theoretically downsample from our reading of 405 to something like 50hz. To do this, we would group accelerometer readings from the original data within each 1/50th of a second interval and then calculate the average for each group, and then use the resampled data to recognize activity.
Source
Source

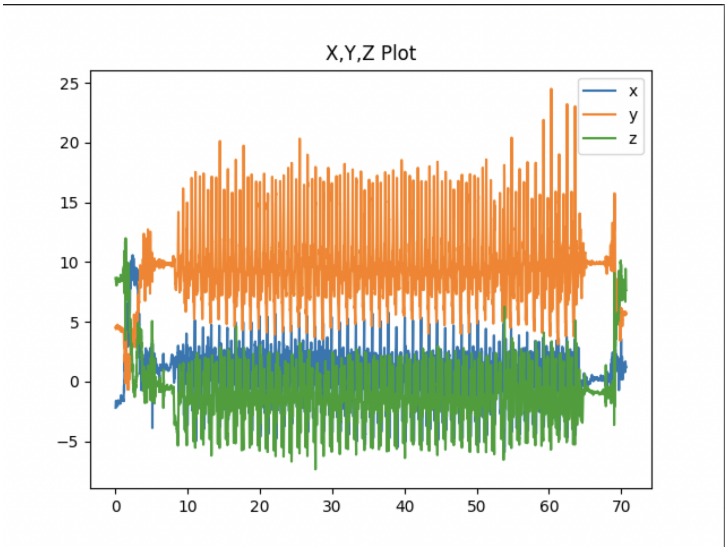
e. Do you think the current sampling rate is a good basis for capturing human movements like walking? Explain your reasons for or against your case.

Yes, a sampling rate of 405 Hz is more than sufficient for capturing human movements, especially walking. Most human movements, including walking, fall within a frequency range of 0.1 to 20 Hz. Even the faster human motions, like sprinting or quick hand movements, seldom exceed this range (from the NIH). According to the Nyquist Theorem, the sampling rate should be at least twice the highest frequency present in the signal to accurately reconstruct that signal from its samples. Given that the highest frequency for human walking is far below 20 Hz, a

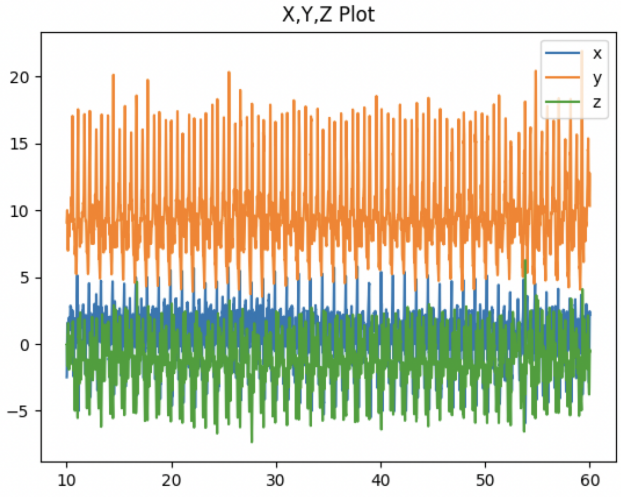
sampling rate of 40 Hz would theoretically be enough. With a rate of 405 Hz, the system is sampling at over 10 times the required rate for walking, ensuring that even finer details are captured (source).

TASK 2 [15 points]

a. [5 points] Plot X, Y, Z for walking_steps_1.csv and walking_steps_1_clean.csv

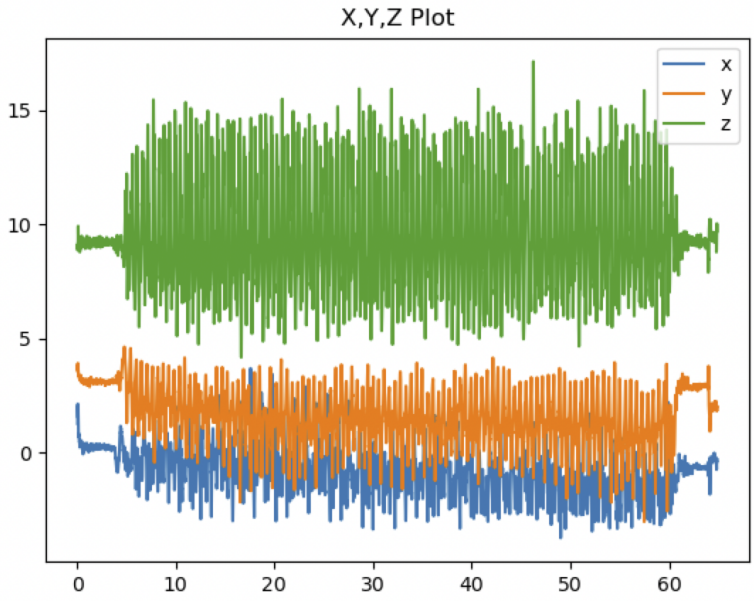


Walking-steps-1-dirty.csv

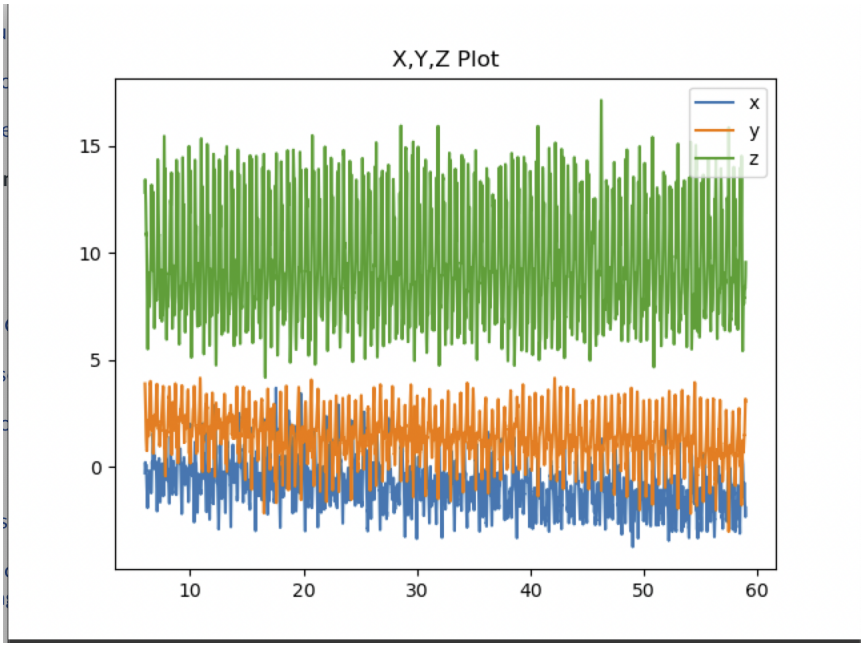


[Walking-steps-1-clean.py]

b. [5 points] Plot X, Y, Z for walking_steps_2.csv and walking_steps_2_clean.csv



[Walking Steps 2 Dirty]



[Walking Steps 2 Clean]

c. [5 points] Reflect how X, Y, Z of walking_steps_1_clean.csv is different from walking_steps_2_clean.csv . Give details as to how you collected the data and how it resulted in movement for each axis.
   a. The starting and stopping points are slightly different in each, so the cleaning process has slightly different time filters. The data for walking-steps-1 was

collected with the phone in Grace's front pocket. The data for walking-steps-2 was collected by being held in front of Grace in her hand.

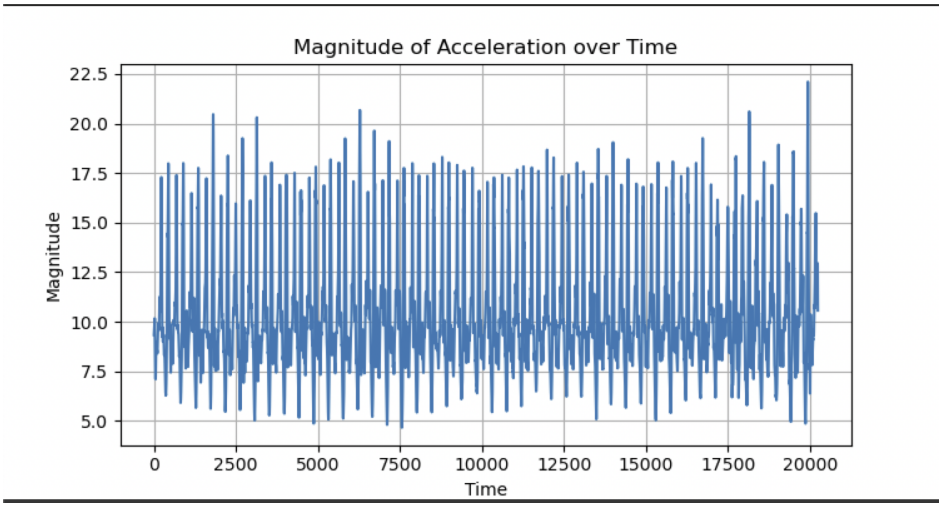
b. In comparing walking_steps_1_clean.csv and walking_steps_2_clean.csv, the plots generally look similar, except two axes (Y and Z) are flipped because the orientation of the phone is different in each. In walking_steps_1, the phone was in the pocket, meaning the phone's orientation was assumed with the top of the phone pointed up. Acceleration due to g in walking_steps_1 is represented by the Y axis, which is approximately 10 m/s^2 while acceleration due to g in waking_steps_2 is represented by Z, which is approx. 10 m/s^2, but a slight tilt while holding the phone slightly alters the Y as well. When the phone is in Grace's hand, the phone is oriented with the screen facing up, meaning that the orientation of the axes is different. In the walking_steps_2, there was less front-to-back movement because the phone was being stabilized in the hand instead of feeling the acceleration of the movement of a leg while walking and shifting in a pocket. This resulted in different amplitudes in the equivalent vertical axes in walking-steps-2 (Z direction) compared to the Y direction in walking_steps_1.

TASK 3 [40 points]

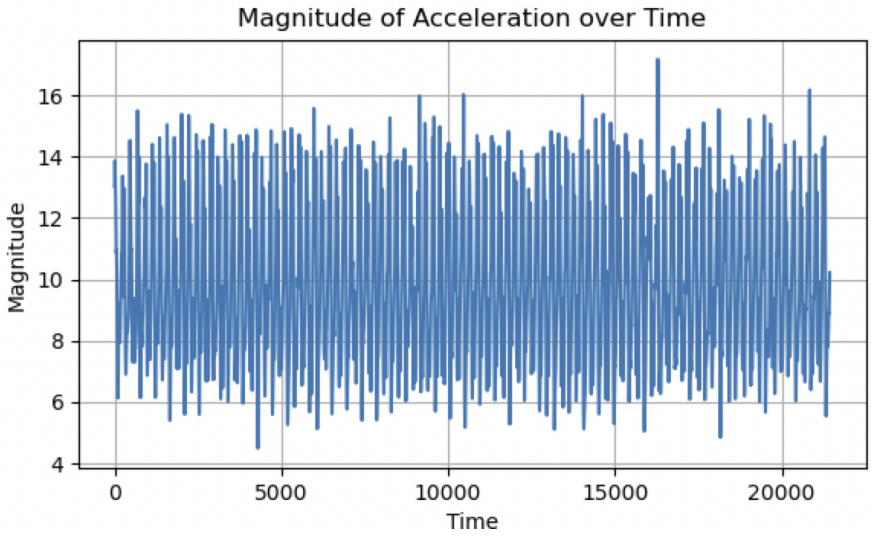
[6 points - 2 for each plot and 2 for reflection]

Part A:

Add plots for moving average for walking_steps_1_clean.csv , walking_steps_2_clean.csv using the method you implemented.

- 

[Walking-Steps-1 magnitude Clean plot]



[Walking-Steps-2 magnitude Clean plot]

Reflect:

1. How magnitude is different from X, Y, Z
   - Magnitude is a unit vector in a direction. X, Y, and Z represent the individual readings or components of a vector in a three-dimensional Cartesian coordinate system. For an accelerometer, the X, Y, and Z readings indicate acceleration values along the x-axis, y-axis, and z-axis, respectively. Each axis measures forces or movements in its designated direction.
   - Magnitude refers to the overall strength of length of a vector in 3D space, which is derived (usually) from the X, Y, and Z components.
   - This provides a scalar value representing the overall intensity or size of the vector without the directional information. In the context of an accelerometer, the magnitude provides a holistic view of the acceleration force experienced by the device (phone), regardless of its orientation.
   - Source 1 used for background information
   - Source 2 used for background information
2. How are the magnitudes different in the 2 files? (hint: orientation)
 [6 points - 2 for each plot and 2 for reflection]

Upon analyzing the two files, it's evident that the magnitude for 'walking-steps-1' is, in general, greater than the other. This file pertains to the scenario where the phone was tucked away in Grace's pocket. A pocketed phone is likely to experience more varied orientations due to the natural sway and motion of the leg during walking.

In contrast, when a phone is held in hand (as one would expect in the other data set), there's a subconscious effort by the individual to stabilize it. The human hand acts as a sort of damper or shock absorber, minimizing abrupt changes in orientation. Therefore, one would expect smaller magnitude variations in this scenario.

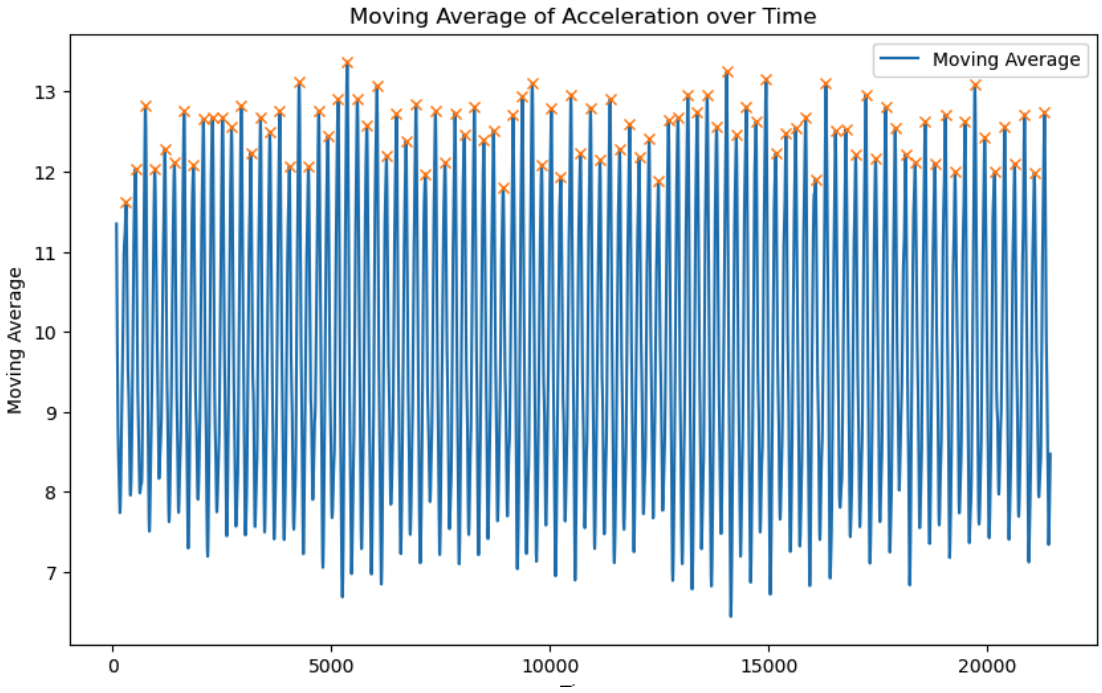
Furthermore, pockets, especially those in loose-fitting clothing, allow the phone to toss and turn freely with each step. This is in stark contrast to when it's in one's hand, where the grip and the inherent damping properties of the human arm come into play.

In reflection, these findings show the importance of considering the context and environment in which data is collected. The same action (in this case, walking) can yield very different data sets based on seemingly minor changes in conditions, such as the placement of the recording device. It serves as a reminder to always account for potential variables that could influence the data.

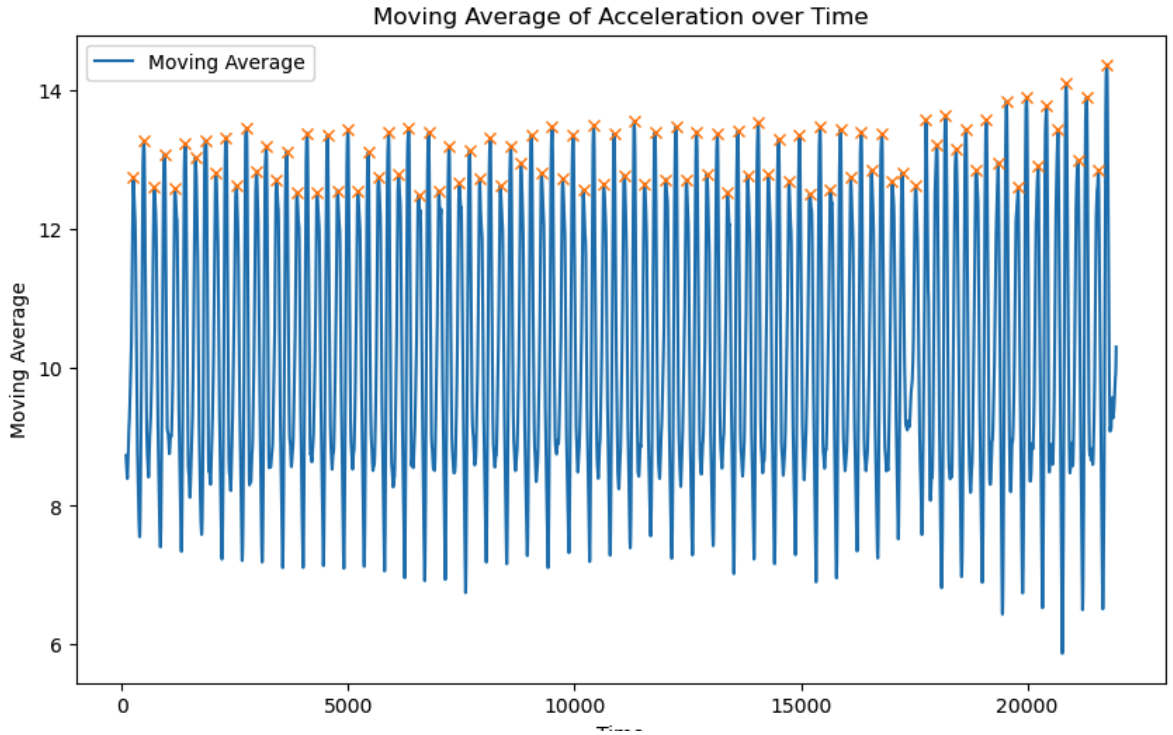
B. Moving Average Plots

[Walking-Steps-1 Moving Average Plot]

Walking-steps-1



[Walking-steps-2]



[Walking-Steps-2 Moving Average Plot]

Reflect

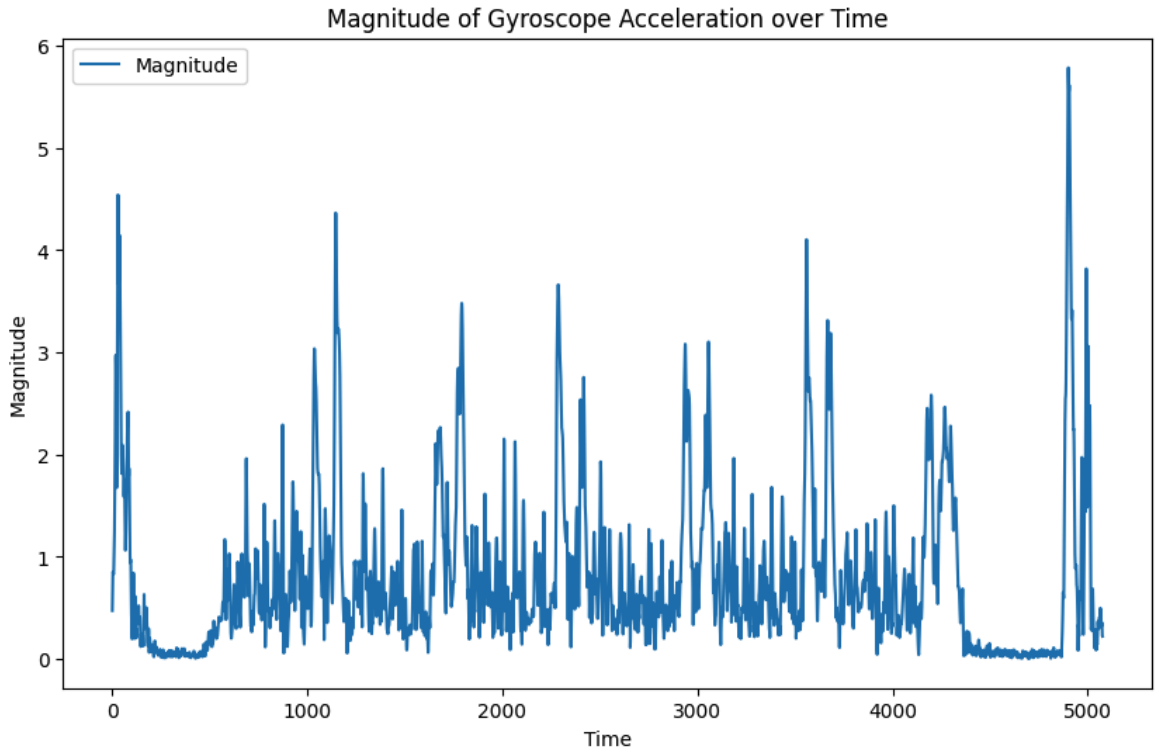
1. How the moving average is different from magnitude.
    - While both magnitude and moving average provide insight into the data's nature, their objectives differ. Magnitude provides a sense of the combined intensity of multi-dimensional data (like from a 3D accelerometer). It gives an instantaneous snapshot of the force at a specific point in time. In contrast, the moving average is a tool to smooth out data and highlight more general patterns over time by smoothing out data fluctuations. By averaging data points over a specified window, it helps identify underlying trends and reduces short-term volatility or noise.

    - Source
2. On what factor(s) does moving average depend?
    - The moving average is primarily influenced by two factors: the underlying data (in our case, magnitude) and the chosen window size. The window size determines how many data points are used to calculate each average. A larger window size will produce a smoother curve, averaging out momentary spikes or drops. In contrast, a smaller window will retain more of the original data's detail, possibly at the risk of capturing noise. The challenge lies in choosing a window size that provides a balance: retaining significant details while also filtering out unnecessary noise.
    - Choosing the right window size is a balance between retaining useful detail and filtering out noise. Too large a window might lose essential details, while too small a window might not effectively filter out noise.

- Source

- [8 points] Describe your step counting process.

For our step counting, we began by plotting the magnitude vectors to visualize the intensity of the accelerometer readings. To better identify individual steps and reduce noise, we then plotted moving averages of these magnitudes. We employed a function aptly named "moving average" for this. This function takes the data (magnitude) and a specified window size as inputs, and returns the average of the latest set of readings based on that window size. We chose a window size of 90, which effectively means that the function calculates the average of the last 90 values. This approach smoothed our curve, ensuring a clearer, singular peak for each step, making step identification more accurate.



d. [10 points - 5 each] How many steps did you count? Plot of your raw data with steps labeled on it

Walking Steps 1: 96 steps
Walking Steps 2: 95 steps
See plots above in part B (moving average plots also show the steps)
Note: We could not figure out why the plots are printed twice.

1. walking_steps_1_clean.csv through your algorithm.
2. walking_steps_2_clean.csv through your algorithm.

e. [10 points] Reflect how d.1 and d.2 are different from ground truth data (100 steps).

- There are several factors that can cause differences between the data recorded by the phone (d.1 and d.2) and the ground truth data of 100 steps:
- **Walking Pace Variation:** The speed at which you walked might have varied during the experiment. Rapid accelerations or decelerations could influence the peak identification, especially if the step frequency changed drastically in a short period.
- **Phone Orientation:** The orientation of the phone can play a significant role. When the phone is in the pocket, its orientation could vary greatly, especially if the pocket is loose. This might affect the magnitude of acceleration readings. Similarly, if the phone was held differently in hand during the walk, it might record varying magnitudes.
- **Turning and Maneuvering:** Turning a corner due to space constraints to continue walking can result in a short and swift step which might not produce a clear peak in the accelerometer data. Such abrupt maneuvers, especially if done quickly, might distort the representation of that step in the data.
- **Footwear and Surface Irregularities**: The type of footwear used and the surface's irregularities can also cause discrepancies. Walking on uneven surfaces might produce irregular step patterns, potentially causing the step count algorithm to miss or double-count steps.
- **Algorithm Sensitivity:** The sensitivity settings of the get_peaks function can impact step count accuracy. If the algorithm is too sensitive, it might register noise as steps. Conversely, if it's not sensitive enough, it might miss actual steps. The algorithm's accuracy would depend on the threshold settings and other parameters that determine a 'peak.'

BEFORE COMING TO CLASS #2

In folder

/data - Add climbing_steps.csv

TASK 4 [30 points]

a. [10 points] Explain your algorithm for segmentation, with rationale.

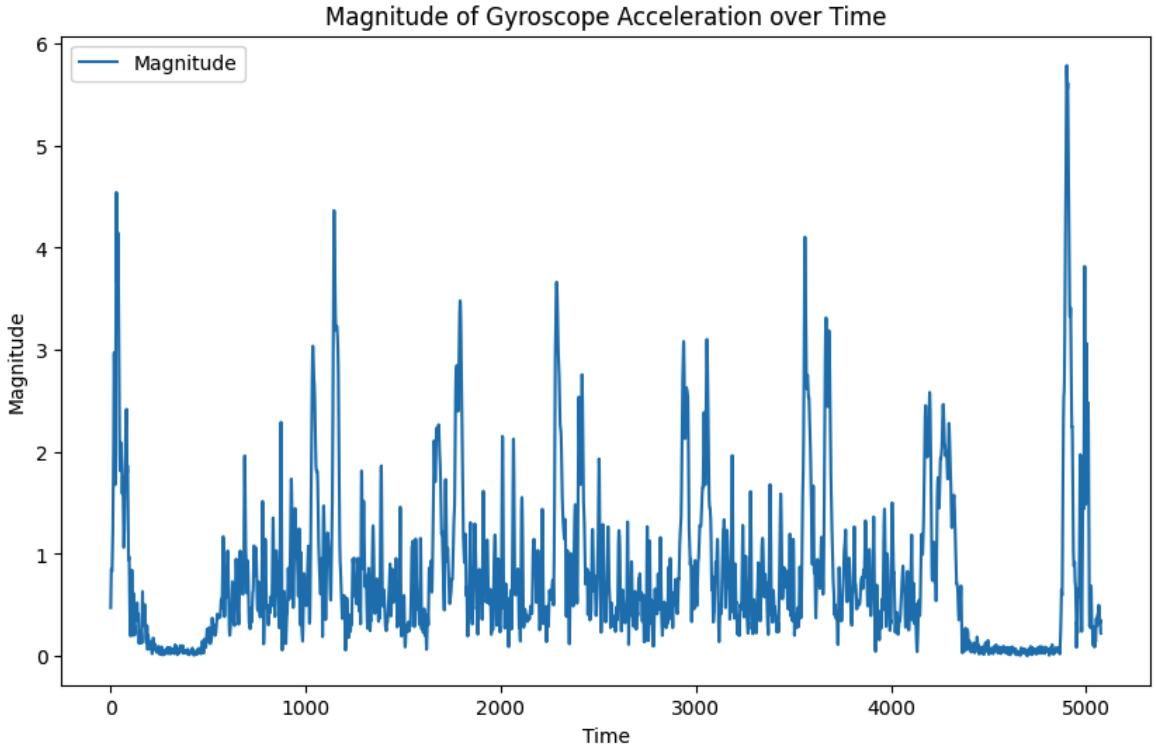
Algorithm for Segmentation:

a. **Data Source Selection:** We begin by primarily focusing on the accelerometer and gyroscope data. The decision to bypass barometer data arises from its unreliability in this context. While it can offer altitude changes, its sensitivity to environmental conditions, like changes in atmospheric pressure, can lead to misleading results.
b. **Utilizing Accelerometer for Step Detection**: The accelerometer measures the linear acceleration of the phone. When walking or taking a step, there's a characteristic motion of a sequence of upward and downward forces. These create recognizable 'peaks' in the accelerometer data. However, while the accelerometer is excellent for detecting the act of taking a step, it struggles to distinguish between different types of steps, such as those taken on flat ground versus those taken while ascending or descending stairs.

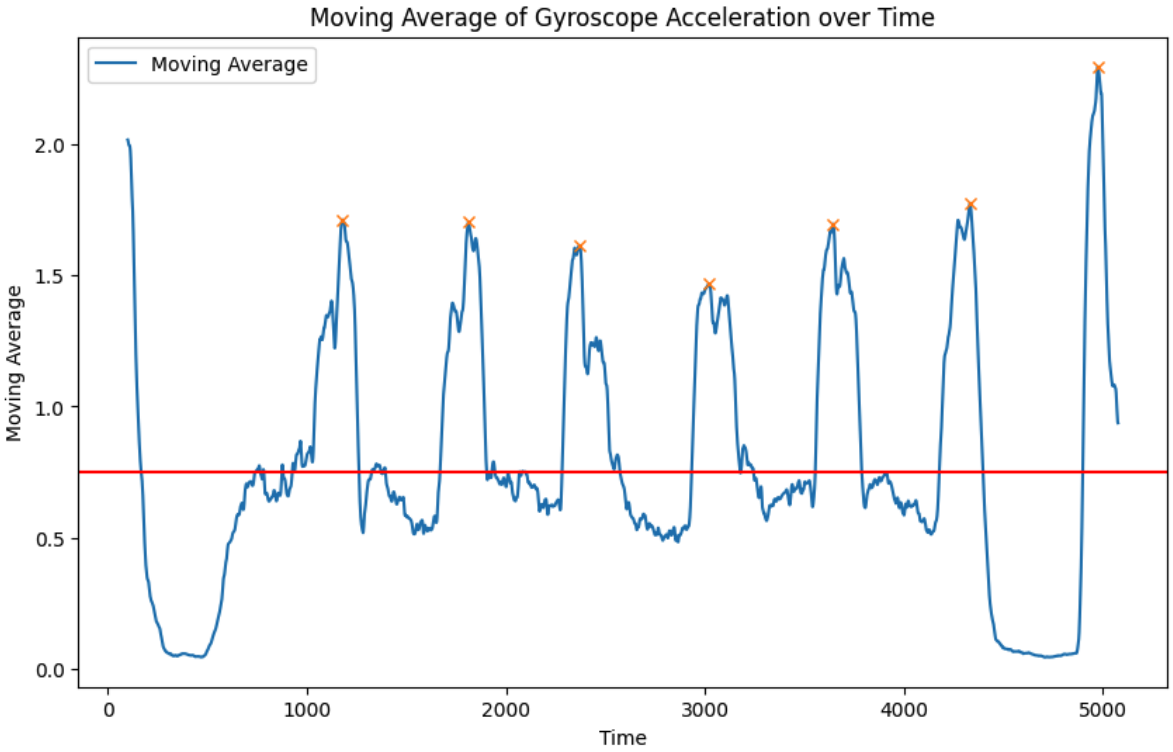
c. **Gyroscope for Detecting Flat Walking Periods:** The gyroscope measures angular velocity, which can help us identify rotations or changes in orientation. As a person walks on a flat surface, there's minimal twisting or turning of the phone (especially if in the pocket). However, when transitioning between walking and stair-climbing, or when reaching the midpoint or end of a floor, there might be subtle changes in orientation. To pinpoint the specific rotational movements, we focused on the consistent changes in the vertical axis acceleration. These changes can manifest as peaks in gyroscope data, signaling these transition points or 'flat' walking times.
d. **Segmentation Process:** With the above knowledge:
    i. We use the gyroscope data to identify periods of flat walking on stair platforms. These are sections where we notice consistent high peaks indicating rotations or orientation changes.
    ii. After identifying flat walking periods, we segment the accelerometer data to exclude these portions identified by the gyroscope data. The rationale here is that these segments mainly contain walking on a flat surface and not stair climbing.
    iii. Once we have segmented the accelerometer data, we then apply the count_peaks function from SciPy and used in Task 3 to count the steps in these segments with local maxima. This gives us a focused count of steps, primarily representing stair-climbing actions.

e. Use both the accelerometer and gyroscope data because the barometer is unreliable. Accelerometer will give peaks that we can consider walking, but don't differentiate well from stair climbing steps and platform steps. The gyroscope will indicate full rotations (peaks) that indicate the flat points at the midpoint and end of each “floor.” The segmentation involves removing the flat walking times indicated by the gyroscope from the accelerometer data before counting steps using the count_peaks function used in Task 3.
f. Note: See Docs for Programming Implementation in references

b. [5 points] Plot of pre-segmented data. Plot segmented data for climbing and walking. Make sure the way you plot (colors/ boxes/lines) easily differentiates the two.



Gyroscope Acceleration Magnitude Plot (Pre-Segmented)

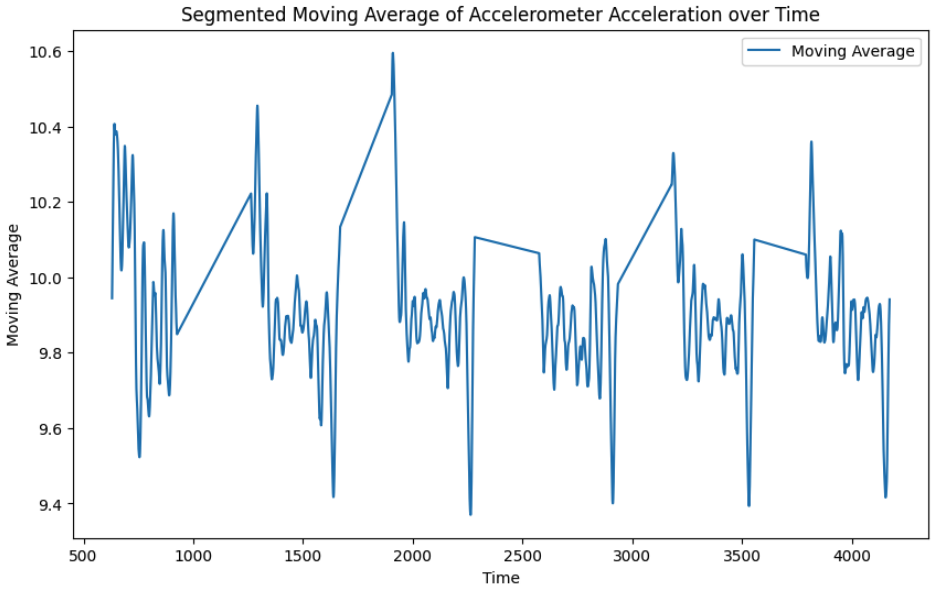


Moving Average Plot of Gyroscope Acceleration over Time

Note: We chose to disregard the intervals where the graph for the moving average was above the red line (y, or the moving average = 0.75 is our threshold)

c. [5 points] What was the difference in the step counting method from task 3?
    a. Needed to use two different sensors and remove instances where walking on the six platforms between floors before counting peaks from accelerometer data
    b. In task 3, the emphasis was primarily on the accelerometer data to detect steps. However, for the advanced step counting in this task, the process harnessed the capability of two different sensors - the accelerometer and the gyroscope. The integration of both sensors provide a more comprehensive picture of the movements, as each sensor captures different aspects of motion. One of the key challenges addressed in this task was the differentiation between regular walking and climbing. They have similar peaks in accelerometer data, however, they have different patterns in the gyroscope data that can help shed light on the way to differentiate them. By taking into account the gyroscope's ability to detect changes in orientation, it became possible to recognize when the user is walking on flat surfaces, like platforms between floors, as opposed to ascending or descending stairs. The step counting method used in this task is inherently more complex than that of Task 3. While Task 3 was more about direct analysis of the magnitude and its moving average, this method required a multi-layered approach to ensure accuracy. By integrating data from multiple sensors and implementing a segmentation process, the method achieves a higher level of precision in step counting, particularly in environments with mixed walking scenarios.
d. [5 points] How many steps of stairs did you count? Plot labeled data for climbed steps.

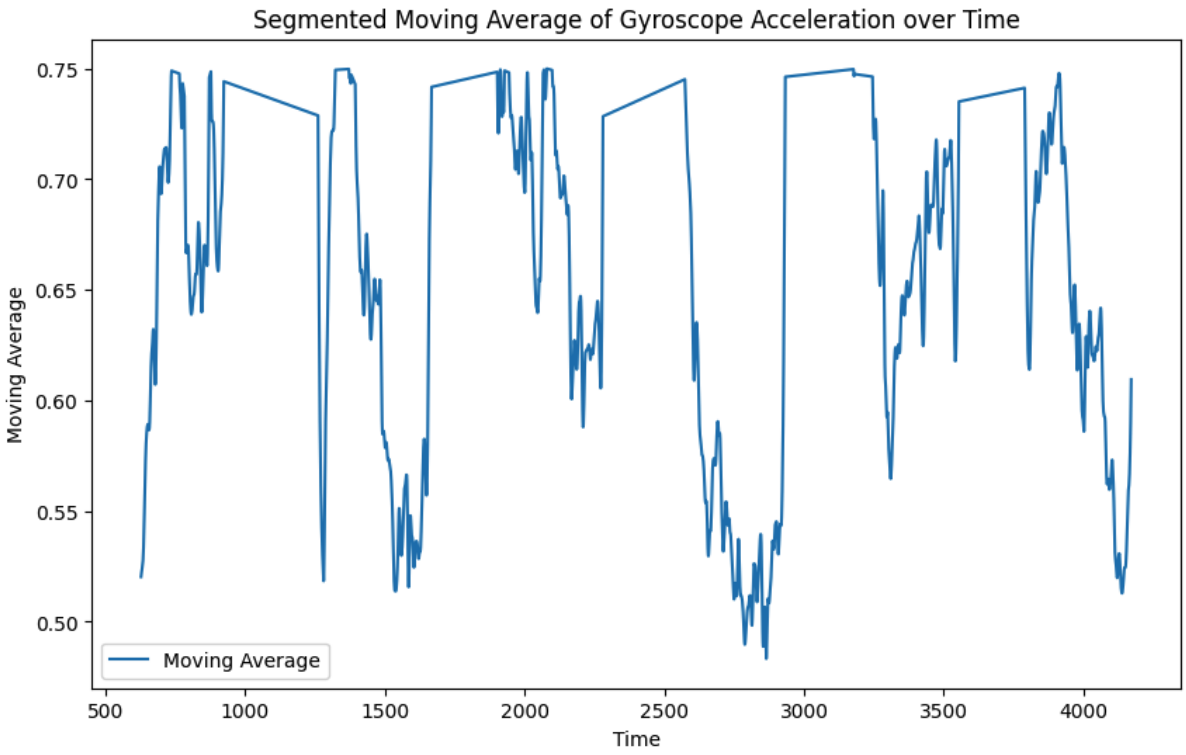
Number of Stairs counted: 84 steps. Note: We were not able to plot the individual steps for this graph, but it was determined as the number of local maxima with the find_peaks function.



Segmented Plot of Accelerometer Moving Average over Time (data over the flat areas is removed)

e. [5 points] Reflect on this exercise and compare it with what you learned in class. (hint: bulling's activity recognition chain)

- Sensing in the real world is imperfect. Bulling's Activity Recognition Chain makes it clear that real work activity recognition is a challenging multi-step process with many possible approaches.
- Exercise gets into the loops of data observation when using accelerometer and gyroscope measurements. This information is particularly applicable/helpful in terms of day-to-day phenomena and what an individual does after obtaining the sensor data itself. In essence, a subsequent iterative process is required to decipher: starting from data cleansing, applying proper calculations for the low pass filter for magnitude (vector magnitude plus moving average), and then the step count after analyzing the signal proceeded to see the peaks, and then making a judgment on that while having some ground truth.
- Our estimation of the actual number of steps based on local maxima performed worse in this case than when using flat walking data. This is likely due to the added complications of using two sensors instead of one and being unable to use the most obvious sensor (barometer).



Additional Helpful Plot: Segmented Plot of Gyroscope Moving Average over Time (data over the flat areas is removed)

TASK 5: Bonus Question [10 points]

a. [5 points] Match the dataset to the TA! Write the name of the dataset as provided to you next to the TA's name.
   i. Dylan -
   ii. Ming -
   iii. Shlok -

b. [5 points] Explain your process and methods - how did you go about solving this problem?

- Although we did not have the proper background to take this task all the way to the finish line, we began to do some initial study of the iterative process based on pure interest only.
- The iterative process will definitely have to be staged over the following:
  - Data Preprocessing: This step involves similar work as in the previous tasks starting with data cleaning. The data could be re-sampled, and noise would again need to be reduced with a moving average function (as explained by Prof. Thomas Ploetz in Multiple Channels lecture video).

- Feature extraction, as mentioned in the comments for the task, would involve transforming raw data to identify features that are indicative of each TA's walking behavior nuances. As per the Google Slides, we would probably jot interest in points for timing and magnitude of acceleration during walking. This is definitely a import we would make:
  - *from sklearn.feature_extraction import DictVectorizer*
- We probably would need to use some ML classifier to train the data points or even take advantage of Weka's GUI (although we don't have first-hand experience with it)
- Once we examine that model and validate it, we can take a prediction and compare it to the ground truth to indicate which sessions in the test data were recorded by each TA.

CONTRIBUTIONS

Please write down the names of the team members that contributed to the deliverable and declare what each contributed to the exercise.

Team Member	Contributions
Anirudh Gattu	- Filling in the gaps, docs, coordination, task2, task4 (attempt), task5 part b
Jong Yoon Kim	- Cleaning the data. - Task 1 Coding - Task 4 Segmentation
Cassandra Marshall	- Task 1 + 2 + 3 expanded documentation - Documenting general answers in the Report - Task 3 coding - Reviewing all documents - Submitting
Grace Pfohl	- Set up Files/GitHub Enterprise - Task 1 Calculations - Task 3 Programming and beginning of Task 4 - Task Documentation - Collected Walking-steps-1.csv and Walking-steps-2.csv

REFERENCES

Docs for Conceptual Understanding:
https://pure.au.dk/ws/files/93103132/sen099_stisenAT3.pdf
https://math.stackexchange.com/questions/413260/on-the-magnitude-of-vectors
https://www.khanacademy.org/math/precalculus/x9e81a4f98389efdf:vectors/x9e81a4f98389efdf:component-form/a/vector-magnitude-and-direction-review
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3859040/#:~:text=As%20shown%20in%20%5B56%5D%2C,is%20contained%20below%2010%20Hz.
https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem
https://stats.stackexchange.com/questions/367910/how-do-i-identify-the-magnitude-of-the-difference-between-two-moving-averages
https://www.wallstreetmojo.com/moving-average-formula/
https://www.earthdatascience.org/courses/use-data-open-source-python/use-time-series-data-in-python/date-time-types-in-pandas-python/resample-time-series-data-pandas-python/#:~:text=As%20previously%20mentioned%2C%20resample(),output%20value%20for%20that%20period.
https://en.wikipedia.org/wiki/Downsampling_(signal_processing)
Docs for Programming Implementation:
https://www.geeksforgeeks.org/plot-the-magnitude-spectrum-in-python-using-matplotlib/#

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.magnitude_spectrum.html
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.filter.html
https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html
https://numpy.org/doc/stable/reference/generated/numpy.array.html
Docs for ML classifiers and related analysis:
https://www.tutorialspoint.com/weka/weka_quick_guide.htm
https://medium.com/@jaz1/holdout-vs-cross-validation-in-machine-learning-7637112d3f8f
https://scikit-learn.org/stable/modules/feature_extraction.html