Simon Fraser University

CMPT 353
Group Project Report
Sensors, Noise, and Walking

Xizhuol Liu (301384881)
Junchen Li (301385486)

Aug 12, 2020

Abstract: This report focuses on analyzing a set of data obtained from physical sensors in the real world, and completing some of the subject issues through modeling and necessary data analysis. More details on the data and the topic will be discussed later

Accomplishment statement

Xizhuo Liu:

Explained technical approach on building classifier models for training and valid data, deploymented classifier models by using GussianNB(), and unknown data files can plug into classifiers to predict results.

Organized graphical vision which aids the presentation and explanation in the following report.

Responsible for uploading files to gitlab, since the team members network configuration error that can not connect to gitlab.

Junchen Li:

Through the continuous debugging of sensors and the adjustment of data format, the detailed information format about all the data is finally determined and the data is collected according to this format. Since the cell phones of the team members could not download the sensor software, all the data sources were measured by me and my family.

Exploring whether the length of the data affected the score by using a linear regression model. Whether the long data must be in accordance with the model established through the long data can be a satisfactory score

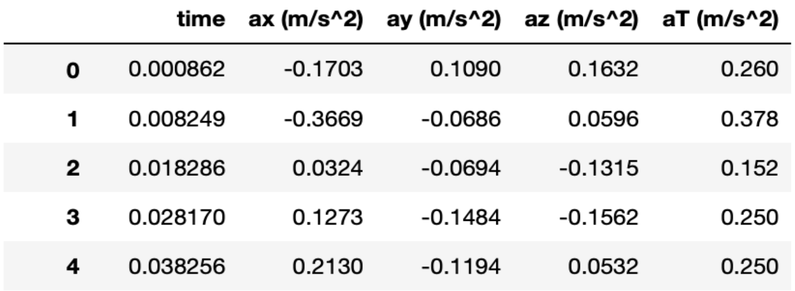
Collaboratively participate in using loops to bring all input files into the normal classifier and injury classifier .

Introduction

Walking is undoubtedly a movement that everyone repeats every day. This experiment is based on the movement of walking to conduct exploration, and collect data of people in different situations when walking through mobile phone sensors. By looking for the best way to collect data and constantly comparing and analyzing the data, we explored whether it was possible to determine whether a person was injured by their gait.

Getting data:

For this experiment, we used a mobile application called Physics Toolbox Sensor Suite. The software records four values at tiny time intervals (Four values are the acceleration in the x, y and z directions of the phone's current position, and the sum of the acceleration at the current time). Note that the formula of total acceleration is $\sqrt{(aX)^2 + (aY)^2 + (aZ)^2}$, since the app will help us to record them (it is the aT), we won't calculate them. The following is a small snippet of the walking data. Using the file called "foot" as an example for the rest of the part which is a ten minutes data file.

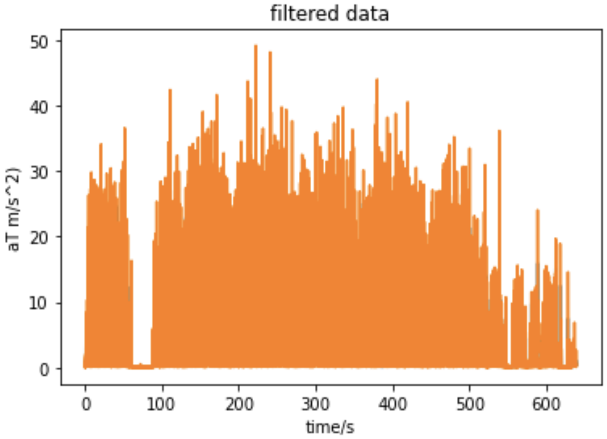


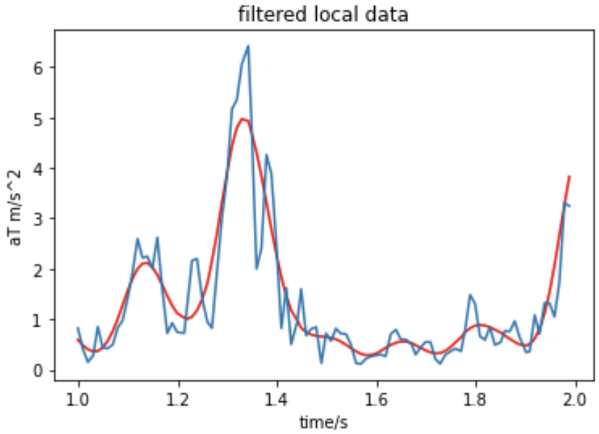
	time	ax (m/s^2)	ay (m/s^2)	az (m/s^2)	aT (m/s^2)
0	0.000862	-0.1703	0.1090	0.1632	0.260
1	0.008249	-0.3669	-0.0686	0.0596	0.378
2	0.018286	0.0324	-0.0694	-0.1315	0.152
3	0.028170	0.1273	-0.1484	-0.1562	0.250
4	0.038256	0.2130	-0.1194	0.0532	0.250

In order to ensure the stability and reliability of the experimental data, the mobile phone was placed in the same position every time the data was measured. For each test, there was a three-second static wait at the beginning and end of the data recording, so that data could be used to eliminate the deviation in reading. The phone was placed at the ankle to better receive real gait information as testers walked. In the situation of injury, the phone also was placed on the right ankle to simulate the injured person walking with a limp on his right foot. Notice that the tests show that a 10-minute session will have more than 65,000 data. Since the amount of data is huge and each test is very time-consuming, we reduced the test time to three minutes and controlled all the data to around 16,000. This not only allows the experiment to proceed more quickly but also ensures that no important data features are lost in the data and we can keep the characteristics of them. Whether the length of collected data will affect the final experimental results will be discussed in depth later.

Simple data analysis:

Filter: For the filter part, a low-pass Butterworth filter is used to further transform the data to eliminate high frequency noise in the sensor reading. It was accomplished through the use of Scipy's signal.butter( ) method, which uses the low pass filter mode of the order of filter used in this function is 3, with the frequency of each sample is 0.15 samples/cycle. This function returns a numerator and a denominator, which are re-plugged into the signal.filtFilt () method to get the filtered file with the same shape as the input. The following two figures show the filtered data respectively. On the left side is the whole filtered data of an input file, and on the right side is a part of the filtered data selected for magnified observation. Right side image can show the filtering is pretty good, eliminating a lot of noise without losing a lot of real signal.

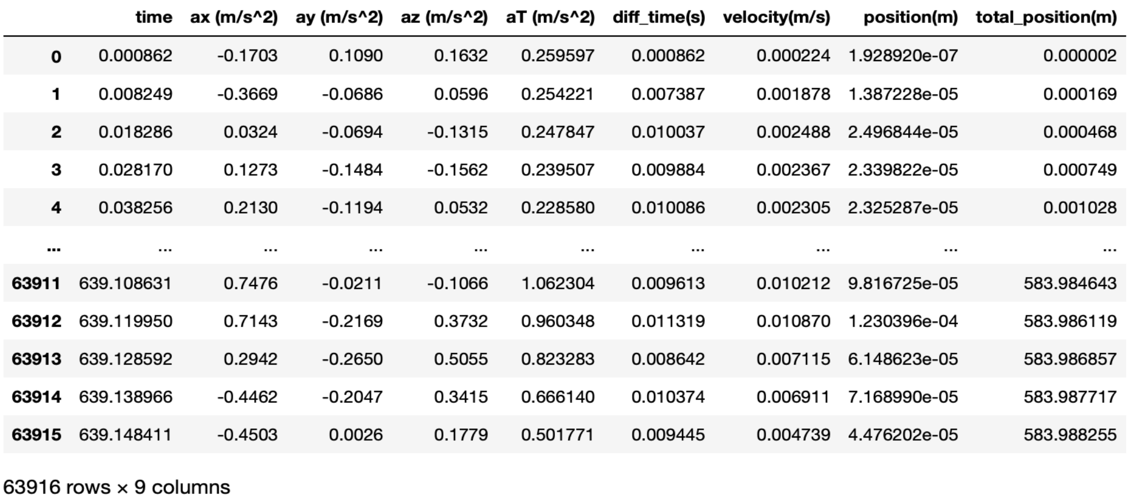




All other files need to be filtered before further analysis.

Dig more information:

According to these formulas: “$\Delta v = a \cdot \Delta t$ and $\Delta p = v \cdot \Delta t$”, more information can be obtained from the known data. By using shift operation, the time difference Δt and ΔaT can be calculated (current one minus previous one) and they can be used to calculate the velocity (m/s) of each time interval. The difference of distance between each time interval can be obtained by multiplying the time difference with velocity. Of course, the distance of test subjects travel is also a very noteworthy issue. Total distance can be calculated by using the function cumsum( ) which is just the cumulative sum of the piecewise distance. The walk speed can be obtained using the velocity formula ($V = \frac{S}{T}$). The image below shows the whole summary of data so far. Note that the column named “velocity(m/s)” is the walking speed for each tiny time interval.

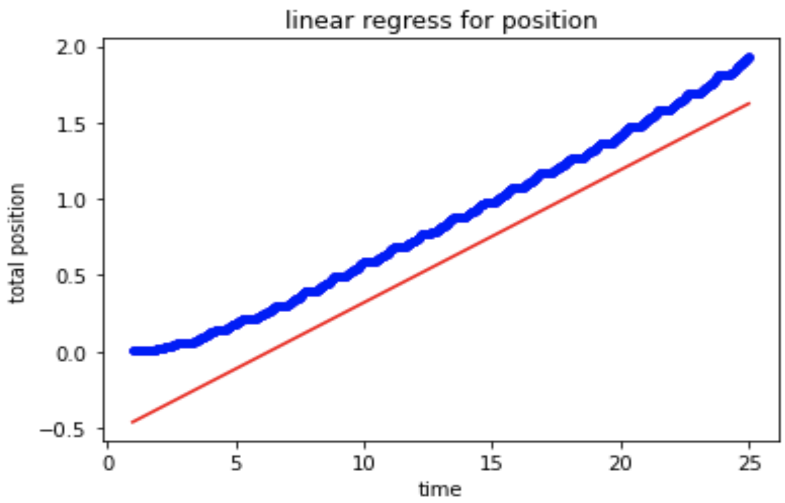


	time	ax (m/s^2)	ay (m/s^2)	az (m/s^2)	aT (m/s^2)	diff_time(s)	velocity(m/s)	position(m)	total_position(m)
0	0.000862	-0.1703	0.1090	0.1632	0.259597	0.000862	0.000224	1.928920e-07	0.000002
1	0.008249	-0.3669	-0.0686	0.0596	0.254221	0.007387	0.001878	1.387228e-05	0.000169
2	0.018286	0.0324	-0.0694	-0.1315	0.247847	0.010037	0.002488	2.496844e-05	0.000468
3	0.028170	0.1273	-0.1484	-0.1562	0.239507	0.009884	0.002367	2.339822e-05	0.000749
4	0.038256	0.2130	-0.1194	0.0532	0.228580	0.010086	0.002305	2.325287e-05	0.001028
...	...	...	...	...	...	...	...	...	...
63911	639.108631	0.7476	-0.0211	-0.1066	1.062304	0.009613	0.010212	9.816725e-05	583.984643
63912	639.119950	0.7143	-0.2169	0.3732	0.960348	0.011319	0.010870	1.230396e-04	583.986119
63913	639.128592	0.2942	-0.2650	0.5055	0.823283	0.008642	0.007115	6.148623e-05	583.986857
63914	639.138966	-0.4462	-0.2047	0.3415	0.666140	0.010374	0.006911	7.168990e-05	583.987717
63915	639.148411	-0.4503	0.0026	0.1779	0.501771	0.009445	0.004739	4.476202e-05	583.988255

63916 rows × 9 columns

Linear Regression:

This section is relevant to the next topic of exploring injuries, and is the basis for starting the experiment. In order to better distinguish whether a new input data is a damaged data, it is necessary to establish a data model that can be used as a reference. Linear regression is a great tool to model filtered data using LinearRegression( ) and fit ( ) functions. The image below is a small part of the intercept:

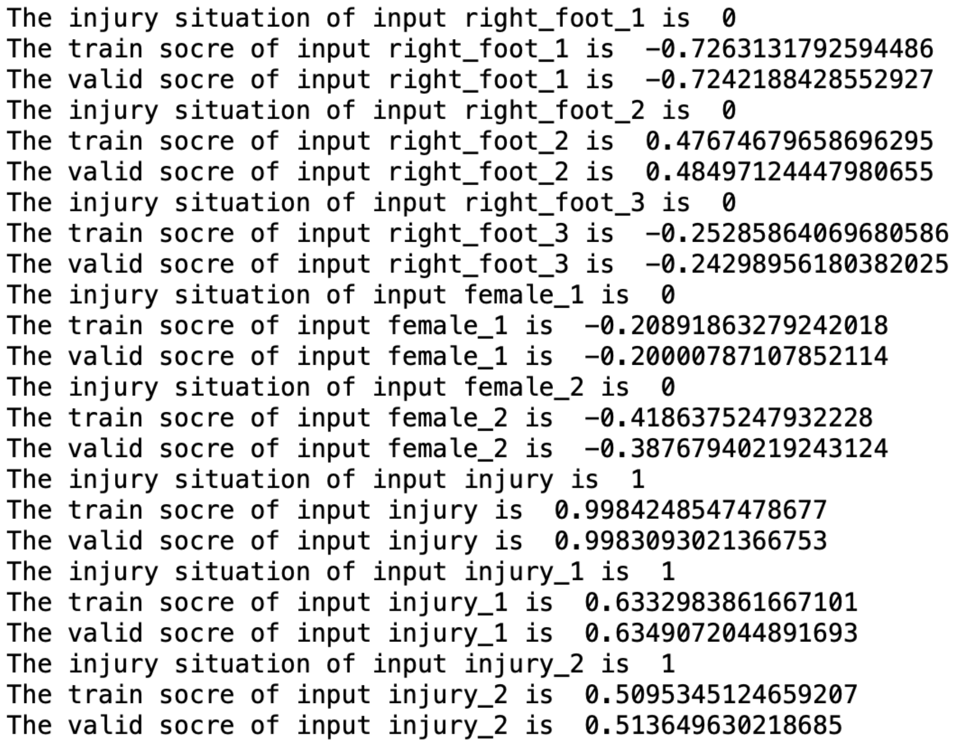


Problem Solving:

The first thing to explore is does the amount of data on the data set used to build the model affect the fit? In other words, if using the 10 minutes data set to build a model, then setting 10 minutes of data and three minutes of data into the model respectively, will get some very different scores? Is the amount of test data consistent in subsequent experiments?
By pouring four groups of files (two data for normal steps, two data for injured steps; The normal gait group and the injured gait group, each consisting of a 10-minute data set and a three-minute data set, were put into two different data models respectively. (The model created with the 10-minute data set and the model created with the three-minute data set) From the data results below, the data model and the length of the input data should match, so that the score is more meaningful.

	Long version model	Short version model
Right Foot (Long version)	0.8541969864910708 (YES)	-0.5678366326635695
Right Foot (Short version)	-0.25827248059251473	0.4552221778850343 (YES)
Injury Foot (Long version)	-9.450070095947572 (YES)	-38.00709471417785
Injury Foot (Short version)	0.3175096683182349	-4.473916943684097 (YES)

Note that all subsequent input files will be three-minute data files. Now that all the uncertainties have been resolved, it's time to get down to the main topic. First of all, as to whether injury can be known by gait, a better solution is to build a classifier that can classify unknown input files by standard. For all data files to be entered, specify "0" for injured input and "1" for injured input, and match them to the names of the files to be entered. The main idea is to use a loop to divide the eight input files into four parts using the train_test_split() function, and then find two scores for the training set and the validation set and save them in a list with an injury situation (0/1). So it generate a summary of 24 data points:

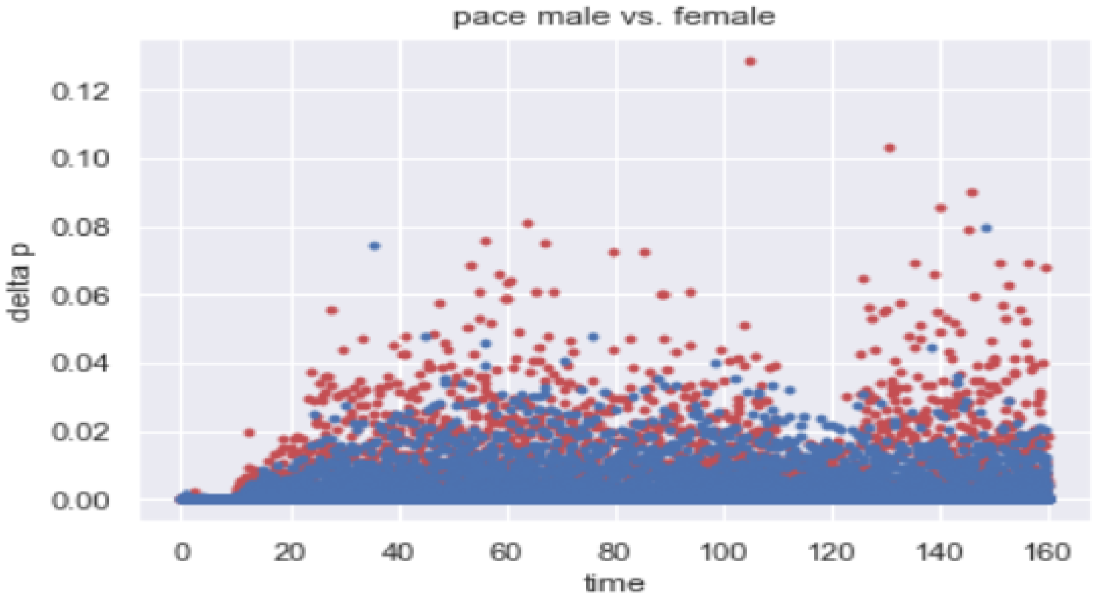


```
The injury situation of input right_foot_1 is  0
The train socre of input right_foot_1 is  -0.7263131792594486
The valid socre of input right_foot_1 is  -0.7242188428552927
The injury situation of input right_foot_2 is  0
The train socre of input right_foot_2 is  0.47674679658696295
The valid socre of input right_foot_2 is  0.48497124447980655
The injury situation of input right_foot_3 is  0
The train socre of input right_foot_3 is  -0.25285864069680586
The valid socre of input right_foot_3 is  -0.24298956180382025
The injury situation of input female_1 is  0
The train socre of input female_1 is  -0.20891863279242018
The valid socre of input female_1 is  -0.20000787107852114
The injury situation of input female_2 is  0
The train socre of input female_2 is  -0.4186375247932228
The valid socre of input female_2 is  -0.38767940219243124
The injury situation of input injury is  1
The train socre of input injury is  0.9984248547478677
The valid socre of input injury is  0.9983093021366753
The injury situation of input injury_1 is  1
The train socre of input injury_1 is  0.6332983861667101
The valid socre of input injury_1 is  0.6349072044891693
The injury situation of input injury_2 is  1
The train socre of input injury_2 is  0.5095345124659207
The valid socre of input injury_2 is  0.513649630218685
```

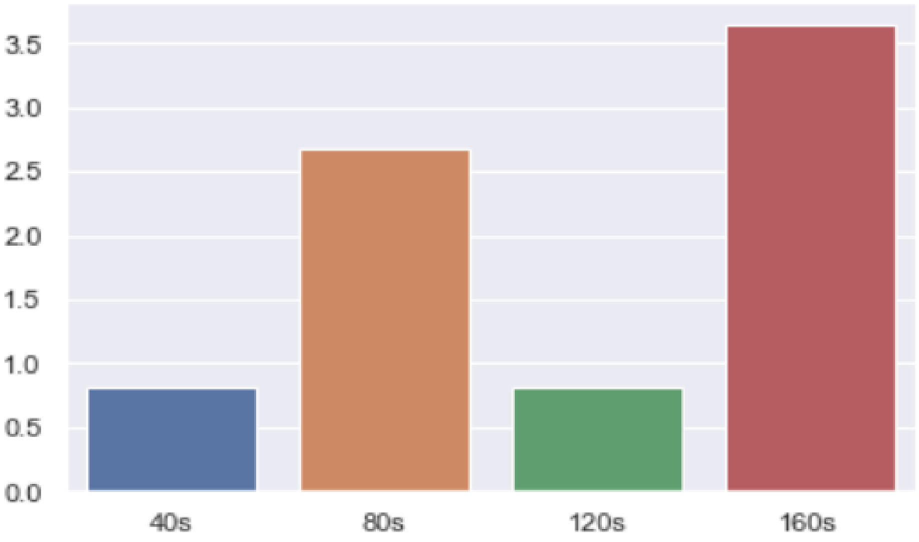
The function GaussianNB( ) is used to put the whole data stack into the classifier. It is worth noting that two parameters are required to make the overall fit. function successfully. Here, the first parameter is the combination of training score and verification score of each file, and the second parameter is the injury situation record. In order to do that, it allows the user to put in an input data set that will return a 1 or 0 to indicate whether the file is the one that was tested by an injured person. Through some test cases, it showed that the classifier is working perfectly. It can also be concluded that the gait data can be used to infer whether a person is injured or not. However, there may have a unknown data that we wanna predict is brought into the normal classifier to predict that the data is biased towards the injured conclusion, and we can not conclude that the data is exactly from a injury person since the normal classifier model is focus on testing whether the person are normal or abnormal, so we still need to plug the unknown data into a injury classifier model. If the data brought into the injury classifier is predict injury, we can conclude that the great probability of this person's data is from the injured person, otherwise, the data predicted to be normal in a injury classifier model then we conclude that this set of data may be abnormal, and we will discard this data since the conclusion is contradictory to the normal classifier prediction. In conclusion, we mainly use counter evidence thinking in mathematics, which is why we need to design two groups of models. In order to verify the authenticity, we will bring the data to be measured into the two groups of models at the same time to improve the accuracy of our data prediction.

Additional Topic:

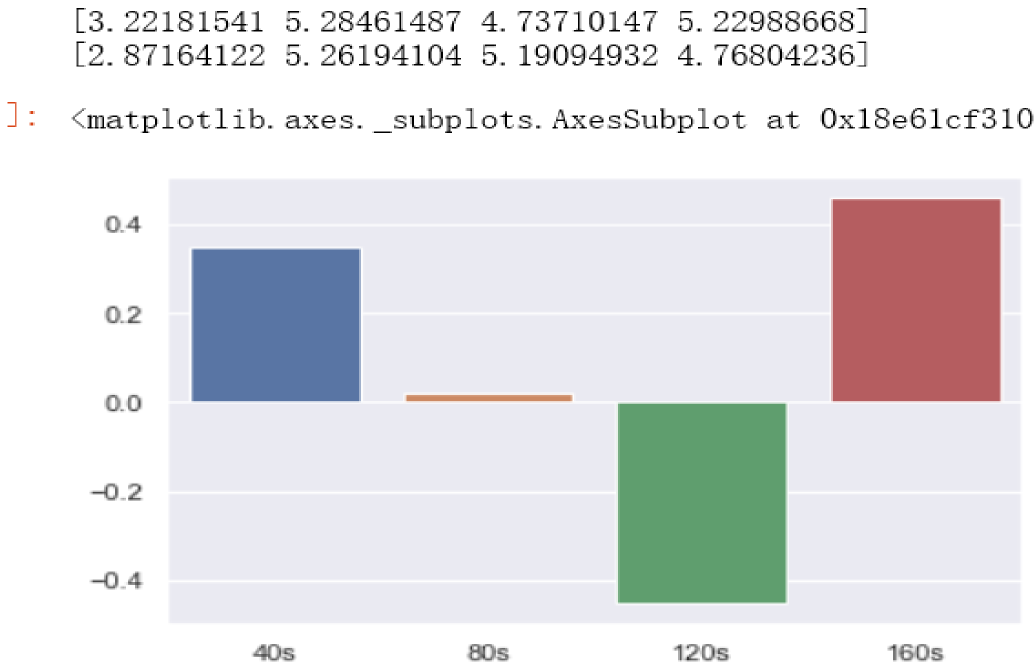
In the following topic, the frequency of steps and the distance traveled in a period of time between males and females are analyzed. First, like the previous project, we first filtered the acceleration, and compared the time and distance samples of a group of male and female sample data.



As can be seen from the data in the above figure, for this control group, the road data of males seems more dense, while the gait of females seems more scattered. It is impossible to confirm this conclusion from a single set of data. In order to better reflect the results, we first collected data between multiple groups of males and females. In order to ensure that the data comparison has more reference significance, we keep consistent for the selected road section and the binding position of mobile phone measurement during the measurement, and the height of the candidates we measured do not differ too much. After collecting several sets of data, we need to cut the time and calculate the distance traveled in a certain time class as a reference unit, so we calculate the distance traveled by males and females every 40 secs. In order to better illustrate this idea, in the graph below, we can see that for the previous comparison of



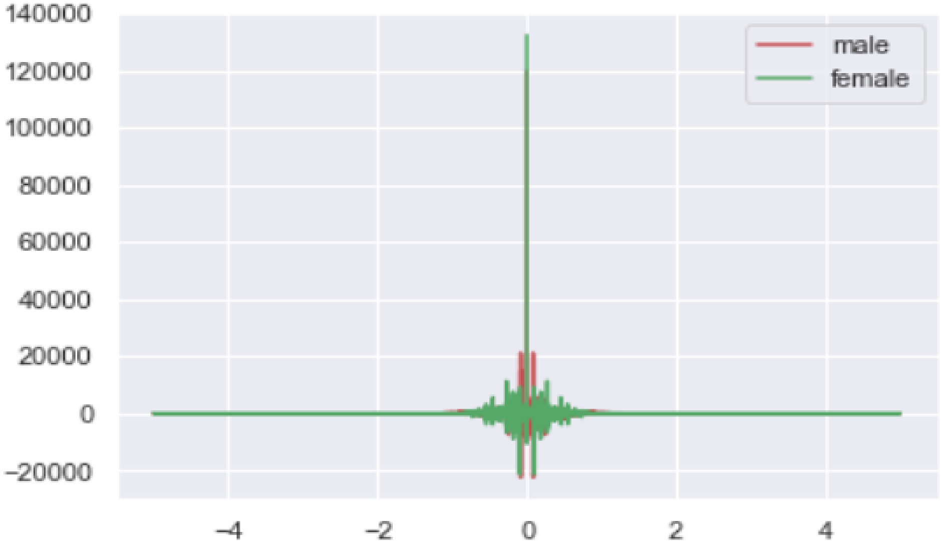
male and female data, we sum the total position for males and females every 40 seconds, and then we subtract the two total positions and get this result. We can see that although the females in the previous figure show that more points are placed directly at the top of the figure (which means more distance at a specific times), however, after the accumulation of time, the total distance of male minus female over a period of time shows that men have traveled more than women in an amount of time. In order to make the analysis more accurate, we measure multiple groups of data and calculate the average value for males and females. The average value of the distance traveled by men every 40 seconds minus the average value of the distance traveled by women every 40 seconds is shown in the following figure.



```
    [3.22181541 5.28461487 4.73710147 5.22988668]
    [2.87164122 5.26194104 5.19094932 4.76804236]

]:  <matplotlib.axes._subplots.AxesSubplot at 0x18e61cf3100>
```

The first row array in the figure represents the distance traveled by men every 40 seconds, and the second row is the distance traveled by women. This set of data is the result of the transformation of distance and time by summing the male and female data respectively and averaging them, then the two sets of numpy data are subtracted to obtain a bar graph of distance difference. In addition, we can infer the dynamics of their steps from the change of acceleration of the two types of genders during this period of time. We decompose the initial accelerations of the two genders by fft function, and obtain the images of the frequency domain of the steps as follows:



The frequency of male is relatively concentrated and fluctuating, and the frequency of female intercourse is extensive and relatively small. Therefore, we draw a conclusion that male may walk more frequently and shorter distances, while women walk more slowly and longer distances.

Introspection and discussion:

Summarize the problems encountered by the team members in the whole experimental project and the possibility of whether the problems can be extended. There are several problems that are worth digging out and digging deeper.

First of all, there are differences in the time format when collecting data. Some time is the information of how many seconds have elapsed, while others are the time information of a certain time on a certain day. This problem was encountered at the beginning of the project, so we had to discard several groups of data and maintain the consistency of time format.
Second, we make a classifier model for both normal data and injured data, however, the file we bring into the classifier may have a difference in the number of normal data and injured data, for example, we may measure the data of 20 groups of normal people and 5 groups of injured people and bring them into the classifier for modeling (in this experiment, we try to ensure the consistency of quantity in the experiment). However, the difference of quantity may lead to the

model not predicting the results of a group of unknown data well when there are many samples, resulting in insufficient data authenticity. This is also a subject that can experiment and analyze the difference between the model and the amount of data brought in. However, it seems that more data sets are needed to verify this result. Therefore, we did not dig into this problem deeply.
Third, the distance data obtained from acceleration and walking speed is not quite accurate, and there may still be a large difference from the actual distance of the real distance. This may be due to the inaccuracy of the data recorded by the tester who passed the up and down slopes and left and right foot turns during the experiment, or the inaccuracy of the distance calculation due to the imperfect calculation formula (The two calculation formulas provided by the teacher were more used)
Fourth, We also think of some interesting experiments. For example, in order to verify whether the sensor is accurate, we want to try to put the sensor on pets to collect data and compare it with human data at the same distance, but since there are no pets that can be taken out of the house among the team members, we abandon this idea temporarily.
In summary, due to the limit of time, we can not get a lot of data to verify the accuracy of the data. We can only get quantitative data within a certain range and analyze it. If we have more time, we will collect more data and bring it into the model to make the experiment more convincing.

Conclusion:

The project flow is to analyze and conceive the data obtained from the sensor, and use the physical formula to obtain the parameters of speed and distance under the condition that of only the sum of time and acceleration. Next, we take random normal person files and injury files as the model of linear expression as the reference item. After using the splitting function, the injured and uninjured files are brought into the model to obtain training and valid score, then we bring the score 0 as normal and 1 as injured into the classifier to get the two classifier models. Finally, if we want to bring in a set of unknown data to predict whether it comes from the injured population, we only need to bring in two classifiers for verification. For additional topics, we compare the distance between the two groups of data of male and female, and analyze the acceleration as the frequency of the pace of male and female.
In this experiment, we summarized the topics that we have insufficient experience and can continue to dig deeply. For example, we can consider measuring the sensitive range of sensors to pets and people, or topics such as distance and calorie consumption. We believe that if we continue to dig deeply into these topics, we will get more interesting phenomena.
In summary, though the amount of overall data is not large enough, we have made the above experimental analysis within the limited data range, and hope that these analyses can help us analyze the changes of the pace of males and females and predict whether a person is injured through the pace data.

Report Reference

scipy.signal.filtfilt¶. scipy.signal.filtfilt - SciPy v1.7.1 Manual. (n.d.). https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.filtfilt.html.

Google. (n.d.). *Physics toolbox sensor suite - apps on Google Play*. Google. https://play.google.com/store/apps/details?id=com.chrystianvieyra.physicstoolboxsuite&hl=en.

Maklin, C. (2019, December 29). *Fast fourier transform*. Medium. https://towardsdatascience.com/fast-fourier-transform-937926e591cb.

Haroon RashidHaroon Rashid 18311 gold badge11 silver badge55 Guillaume ChevalierGuillaume Chevalier 11122 bronze badges. (1966, November 1). *Apply low pass butterworth filter in python*. Signal Processing Stack Exchange. https://dsp.stackexchange.com/questions/49460/apply-low-pass-butterworth-filter-in-python.

Digital low Pass Butterworth filter in Python. GeeksforGeeks. (2020, December 8). https://www.geeksforgeeks.org/digital-low-pass-butterworth-filter-in-python/.

Python numerical methods. Fast Fourier Transform (FFT) - Python Numerical Methods. (n.d.). https://pythonnumericalmethods.berkeley.edu/notebooks/chapter24.03-Fast-Fourier-Transform.html.

Patrick, J. H. (2003, August 21). *The case for gait analysis as part of the management of incomplete spinal cord injury*. Nature News. https://www.nature.com/articles/3101524.