We are trying to use various ML techniques to accurately predict arrythmias based on a patient’s ecg record. Currently using the MIT-BIH Arrythmia Database available here:

<https://www.physionet.org/content/mitdb/1.0.0/>

Download the zip file, and more info on the database is available here:

<https://archive.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm>

To read the data you need the physionet WFDB api, then import “wfdb” into your python code

<https://github.com/MIT-LCP/wfdb-python>

I will attach a paper that uses this dataset for research and how they processed the data. “Automated Screening of Arrhythmia Using Wavelet Based Machine Learning Techniques”

Essentially you preprocess the signal, in our case we remove baseline wander, example of before/after removal. Done using ‘heartpy’ found:

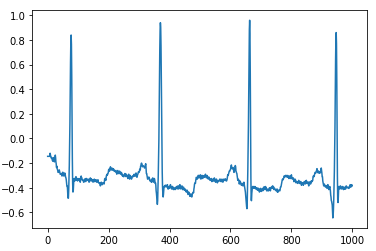
<https://github.com/paulvangentcom/heartrate_analysis_python>

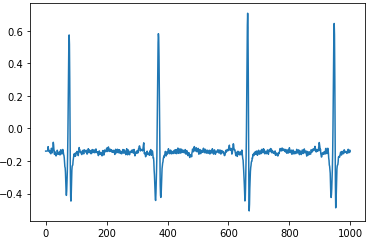
then import ‘heartpy’ into your code

the specific function is found in here, but there are other filters you can try if you wish

<https://python-heart-rate-analysis-toolkit.readthedocs.io/en/latest/heartpy.heartpy.html#filtering>

“heartpy.remove\_baseline\_wander”





Each record has annotations that go with the peak of the pulse, the annotation describes the type of heartbeat. Some datasets may require peak detection, but because this set already identifies where each is, that is currently unnecessary.

After the full signal is processed, each beat must be isolated. We are choosing the midpoint between annotation indices to separate. From here each beat is resampled to 200 datapoints. (done using sklearn)

There are about 112,000 total heart beats in the dataset. We choose a training subset and a test subset.

For now, the full dataset is not necessary, because we just need a rough idea of how different methods affect the accuracy.

I am currently using Support Vector Machines (SVM) for prediction, but there are other methods for classification. But I will soon be trying Linear Discriminant Analysis (LDA). Both of which can be found at the sklearn user guide:

<https://scikit-learn.org/stable/user_guide.html>

1.2 is LDA and 1.4 is SVM

All the functions and examples you need are there.

For SVM the kernel that has so far been the best is ‘linear’ and ‘poly’ (degree 1)

There is a ‘C’ value also that is an optimization parameter, it defaults to 1.0, but should be tested at various levels, for example: 0.001, 0.01, 0.1, 1.0, 2.0, 5.0, 10.0.

I’m using python notebook to run everything as it makes it easier than running a full .py constantly.

As you separate each pulse, I would recommend writing each to a csv file, with each row being a pulse. That can be easily googled if you don’t already know how. Having the csv file makes the it much easier for testing the ML techniques.

I think this covers all the basics to know, start getting a feel for the functions and what they represent. If you have questions definitely let me know, and we’ll find a time next week to meet in person.

It’ll be far easier to talk in person than through any messaging.