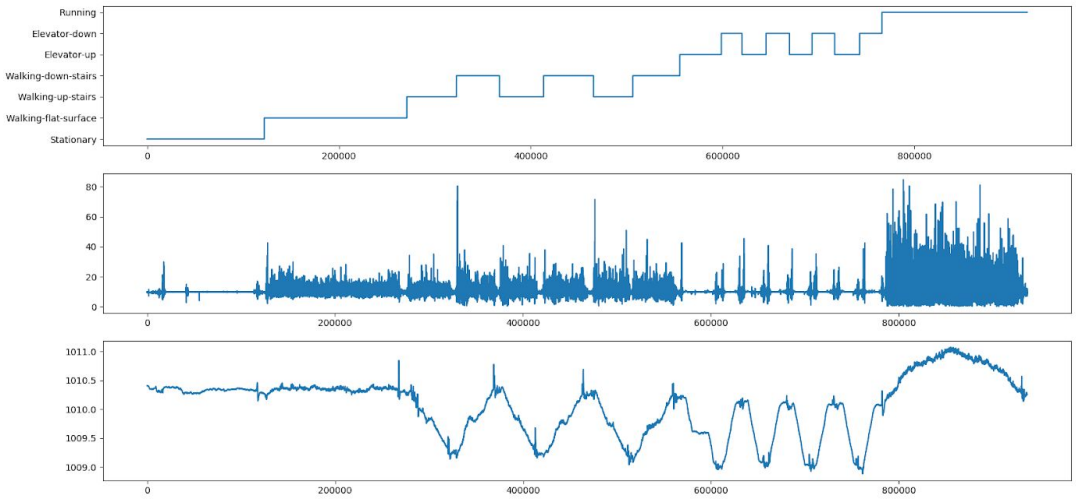
Dennis Shen
CS590U assignment 2 report

Step 1:
To get a larger version, you can run visualize_data.py in the uploaded code and recreate the graph. There will also be an attached .png image in the uploaded homework files



The x-axis is always time in milliseconds, the first graph is activity vs time, the second is accelerometer magnitude vs time and the last one is barometric pressure vs time.

The data is my own stitched together activities. (dshen)

The mean and variance of the accelerometer magnitude seems like they should be effective for identifying stationary and running. It may also help in determining walking and moving up and down stairs. The range can also be useful for using the stairs and elevator. Medium range + low mean implies elevator usage. The slope of the barometric pressure seems like it will be effective at finding activities which have vertical movement like walking on stairs and using the elevator and whether the direction is up or down. Although there is a large change in barometric pressure during the run, this is because I made a mistake and did not run on a flat area. I thought I might try stitching together the activities randomly for feature extraction, but I realized there would be a problem because barometric pressure would start to suddenly jump huge amounts.

Step 2:
The feature extraction code is in data_extraction.py

Step 3-5 note: in the uploaded code, running train_models.py will run all the code for steps 3-5. To isolate and run only one, comment out the necessary code at the bottom of the file

Step 3: the function for step 3 is located in train_models.py, it is called step_3_classifier()
I tried using the KNN classifier and decision tree classifiers.
First I made a stratified split of the data into 80% training and 20% testing.
I used stratified 5 fold cross validation on the training set to determine the best number of neighbors/best maximum depth for each classifier respectively.
For the KNN classifier, I found that the 1 nearest neighbor classifier had the highest overall accuracy.

After training on the entire training set and testing on the test set got these results
Stationary
recall 0.8285714285714286
precision 0.8903508771929824
accuracy 0.9530154277699859
Walking-flat-surface
recall 0.7813620071684588
precision 0.8288973384030418
accuracy 0.9256661991584852
Walking-up-stairs
recall 0.8349514563106796
precision 0.7644444444444445
accuracy 0.9389901823281908
Walking-down-stairs
recall 0.7760416666666666
precision 0.7760416666666666
accuracy 0.9396914446002805
Elevator-up
recall 0.7070063694267515
precision 0.7655172413793103
accuracy 0.9438990182328191
Elevator-down
recall 0.7967479674796748
precision 0.7
accuracy 0.9530154277699859
Running
recall 0.9285714285714286
precision 0.8927038626609443
accuracy 0.9712482468443198

For the decision tree classifier I used a max depth of 31 and got these results:
Stationary
recall 0.8938775510204081
precision 0.9125
accuracy 0.9670406732117812
Walking-flat-surface
recall 0.9139784946236559
precision 0.9107142857142857
accuracy 0.9656381486676017
Walking-up-stairs
recall 0.8543689320388349
precision 0.8341232227488151
accuracy 0.9544179523141655
Walking-down-stairs
recall 0.7708333333333334
precision 0.783068783068783
accuracy 0.9403927068723703
Elevator-up
recall 0.7452229299363057
precision 0.740506329113924
accuracy 0.9431977559607293
Elevator-down
recall 0.6991869918699187
precision 0.7478260869565218
accuracy 0.9537166900420757
Running
recall 0.9330357142857143
precision 0.8969957081545065
accuracy 0.9726507713884993

The KNN classifier was better in some aspects, but the decision tree seemed better overall. The easiest to classify activities were stationary and running which isn't very surprising since they often have the most extreme values for both the accelerometer frequency and time domain features. Even though the barometric pressure seems like it should be a huge help for the stair and elevator values, I think the data collection for those might be somewhat messed up. I suspect other students recorded their values in a similar way to me and there were not many prolonged periods for the elevator data. Plus, going immediately up and down the elevator probably did not help since in real life a person would likely not immediately move up and down the elevator. There is another problem for walking up and down stairs where there was more than a few steps of flat-walking-distance between each flight of stairs between floors in the CS building. Finally, barometric pressure might not be providing useful data for activities like walking flat, stationary and running and could just be creating noise.

Step 4: you can run the code with the function step_4_classifier() located in train_models.py
Again I tried using both KNN and decision tree for both parts

Step 4a:
KNN classifier
best number of neighbors 26

Stationary
recall 0.6913123844731978
precision 0.7540322580645161
accuracy 0.9051214707813526
Walking-flat-surface
recall 0.4896
precision 0.392811296534018
accuracy 0.7399868680236376
Walking-up-stairs
recall 0.40835266821345706
precision 0.55
accuracy 0.8690085357846355
Walking-down-stairs
recall 0.3325581395348837
precision 0.5437262357414449
accuracy 0.8663821405121471
Elevator-up
recall 0.6600660066006601
precision 0.45351473922902497
accuracy 0.8870650032829941
Elevator-down
recall 0.45454545454545453
precision 0.5555555555555556
accuracy 0.927774130006566
Running
recall 0.8375527426160337
precision 0.7231329690346083
accuracy 0.9248194353250164

Decision tree classifier

best max_depth 9
Stationary
recall 0.49722735674676527
precision 0.840625
accuracy 0.8939592908732764
Walking-flat-surface
recall 0.3552
precision 0.5068493150684932
accuracy 0.7967826657912016
Walking-up-stairs
recall 0.6218097447795824
precision 0.6077097505668935
accuracy 0.8896913985554826
Walking-down-stairs
recall 0.7023255813953488
precision 0.41483516483516486
accuracy 0.8181221273801708
Elevator-up
recall 0.7491749174917491
precision 0.49134199134199136
accuracy 0.8978988837820092
Elevator-down
recall 0.6446280991735537
precision 0.639344262295082
accuracy 0.942875902823375
Running
recall 0.8354430379746836
precision 0.9588377723970944
accuracy 0.968811556139199

Aside from stationary, it seems that the decision tree performs way better than the KNN classifier. The location mismatch definitely hurt the classifier

Part 4b.
KNN classifier
best number of neighbors 24

Stationary
recall 0.706997084548105
precision 0.6170483460559797
accuracy 0.8770812928501469
Walking-flat-surface
recall 0.3823146944083225
precision 0.3295964125560538
accuracy 0.7372673849167483
Walking-up-stairs
recall 0.1480865224625624
precision 0.3852813852813853
accuracy 0.8398628795298727
Walking-down-stairs
recall 0.3931947069943289
precision 0.2810810810810811
accuracy 0.7911361410381978
Elevator-up
recall 0.36721991701244816
precision 0.5822368421052632
accuracy 0.8942213516160626
Elevator-down
recall 0.24731182795698925
precision 0.30564784053156147
accuracy 0.8802644466209598
Running
recall 0.8589147286821706
precision 0.6674698795180722
accuracy 0.9101371204701273

DT classifier
best max_depth 4

Stationary
recall 0.7682215743440233
precision 0.7772861356932154
accuracy 0.9240940254652301
Walking-flat-surface
recall 0.2925877763328999
precision 0.7575757575757576
accuracy 0.8491674828599413
Walking-up-stairs
recall 0.7487520798668885
precision 0.5005561735261401
accuracy 0.8530852105778648
Walking-down-stairs
recall 0.6257088846880907
precision 0.47421203438395415
accuracy 0.8616552399608227
Elevator-up
recall 0.46887966804979253
precision 0.578005115089514
accuracy 0.8969147894221352
Elevator-down
recall 0.49193548387096775
precision 0.23735408560311283
accuracy 0.809745347698335
Running
recall 0.5255813953488372
precision 0.9685714285714285
accuracy 0.9223800195886386

I feel like the decision tree was better again, but both classifiers were pretty terrible for many activities. I think training on the hand and testing on the pocket was worse than training on the pocket and testing on the hand. Not sure why this is the case, maybe because hand movement is less consistent from person to person. The obvious fix to location mismatch is to not mismatch the location. However if a classifier must work with both locations, then adding location data to the list of things to predict and doing some sort of multi label classification might help. Also making use of the gyroscope data might help since I imagine that there are different patterns in pocket orientation and orientation.

Step 5:
Again I attempted both parts using both a KNN classifier and decision tree classifier

Time domain features only
KNN classifier
best number of neighbors 1

Stationary
recall 0.9183673469387755
precision 0.9221311475409836
accuracy 0.9726507713884993
Walking-flat-surface
recall 0.9068100358422939
precision 0.8815331010452961
accuracy 0.9579242636746143
Walking-up-stairs
recall 0.8446601941747572
precision 0.8285714285714286
accuracy 0.9523141654978962
Walking-down-stairs
recall 0.765625
precision 0.7903225806451613
accuracy 0.94109396914446
Elevator-up
recall 0.6878980891719745
precision 0.72
accuracy 0.9361851332398317
Elevator-down
recall 0.6666666666666666
precision 0.6612903225806451
accuracy 0.9417952314165497
Running
recall 0.9285714285714286
precision 0.9244444444444444
accuracy 0.9768583450210379

This classifier performs similarly to the ones using all the features in step 3. It's even better in stationary and running than the step 3 classifiers

Time domain features only
DT classifier
best max_depth 15

Stationary
recall 0.8938775510204081
precision 0.9087136929460581
accuracy 0.9663394109396914
Walking-flat-surface
recall 0.8960573476702509
precision 0.8802816901408451
accuracy 0.955820476858345
Walking-up-stairs
recall 0.8155339805825242
precision 0.7813953488372093
accuracy 0.9403927068723703
Walking-down-stairs
recall 0.78125
precision 0.819672131147541
accuracy 0.9474053295932678
Elevator-up
recall 0.7006369426751592
precision 0.7534246575342466
accuracy 0.9417952314165497
Elevator-down
recall 0.7235772357723578
precision 0.6953125
accuracy 0.9488078541374474
Running
recall 0.9241071428571429
precision 0.9039301310043668
accuracy 0.9726507713884993

Again I feel like the decision tree is slightly better overall than the KNN classifier.

Frequency domain features only
KNN classifier
best number of neighbors 24

Stationary
recall 0.4857142857142857
precision 0.44074074074074077
accuracy 0.8057503506311361
Walking-flat-surface
recall 0.26523297491039427
precision 0.2846153846153846
accuracy 0.7258064516129032
Walking-up-stairs
recall 0.5631067961165048
precision 0.42962962962962964
accuracy 0.8288920056100981
Walking-down-stairs
recall 0.484375
precision 0.34572490706319703
accuracy 0.8071528751753155
Elevator-up
recall 0.46496815286624205
precision 0.553030303030303
accuracy 0.8997194950911641
Elevator-down
recall 0.4634146341463415
precision 0.5588235294117647
accuracy 0.9221598877980365
Running
recall 0.2857142857142857
precision 0.5203252032520326
accuracy 0.8464235624123422

Frequency domain features only
DT classifier
best max_depth 10

Stationary
recall 0.7020408163265306
precision 0.8557213930348259
accuracy 0.9284712482468443
Walking-flat-surface
recall 0.7956989247311828
precision 0.6082191780821918
accuracy 0.8597475455820477
Walking-up-stairs
recall 0.6796116504854369
precision 0.7608695652173914
accuracy 0.9228611500701263
Walking-down-stairs
recall 0.6614583333333334
precision 0.5879629629629629
accuracy 0.8920056100981767
Elevator-up
recall 0.6114649681528662
precision 0.64
accuracy 0.9193548387096774
Elevator-down
recall 0.6260162601626016
precision 0.77
accuracy 0.9516129032258065
Running
recall 0.5625
precision 0.6
accuracy 0.8723702664796634

This time the decision tree clearly performs far better than the KNN classifier. I'm not surprised, but this without the time domain features the classifiers aren't doing as well. I think the frequency domain features are also more susceptible to being thrown off by the poor data collection (up and down the elevator). Although I experimented a little bit with different spectral rolloff amounts 80-95%, I didn't feel like it made a huge difference.

It seems like the classifiers get the most out of the time domain features since the time domain features alone are able to make a classifier about as good as the step 3 all-feature classifiers. Plus the time domain features decision tree classifier performs better across all activities compared to either of the frequency domain classifiers.