assignment_3

May 22, 2022

1 Assignment 3

Joshua Schmidt (jns223)

[ ]:
```
import pandas as pd
import numpy as np

import librosa
import librosa.display

import matplotlib.pyplot as plt

import geopy
from geopy.geocoders import Nominatim
import plotly.graph_objects as go

from IPython.display import Audio
```

1.0.1 Load data

[ ]:
```
# Here are the sample code to load the three types of data. Change it based on␣
↪your needs.

location_dir = "./data/person2-1DBB0F6F-1F81-4A50-9DF4-CD62ACFA4842.
↪absolute_locations.csv.gz"
location = pd.read_csv(location_dir, compression='gzip')

feature_label_dir = "./data/person2-1DBB0F6F-1F81-4A50-9DF4-CD62ACFA4842.
↪features_labels.csv.gz"
feature_label = pd.read_csv(feature_label_dir, compression='gzip')
```

1.0.2 Inspect data

[ ]:
```
location
```

[ ]:
```
        timestamp   latitude   longitude
0      1440627472  32.882277 -117.234632
```

```
1     1440627533  32.882289 -117.234622
2     1440627593  32.882289 -117.234629
3     1440627654  32.882292 -117.234630
4     1440627712  32.882284 -117.234628
…          …          …          …
7370  1441292839  32.878956 -117.231589
7371  1441292931  32.878955 -117.231589
7372  1441292959  32.878956 -117.231589
7373  1441293052  32.878956 -117.231589
7374  1441293083  32.878956 -117.231589

[7375 rows x 3 columns]
```

[ ]:
```
feature_label
```

[ ]:
```
       timestamp  raw_acc:magnitude_stats:mean  raw_acc:magnitude_stats:std  \
0     1440627472                      0.978119                     0.002341
1     1440627533                      0.978315                     0.001636
2     1440627593                      0.978582                     0.002582
3     1440627654                      0.978640                     0.001695
4     1440627712                      0.978938                     0.001981
…          …                             …                            …
7370  1441292839                      1.014379                     0.001043
7371  1441292931                      1.013954                     0.001003
7372  1441292959                      1.014174                     0.000964
7373  1441293052                      1.014009                     0.001029
7374  1441293083                      1.014237                     0.000996

      raw_acc:magnitude_stats:moment3  raw_acc:magnitude_stats:moment4  \
0                            0.002273                         0.005612
1                            0.000510                         0.002482
2                           -0.002260                         0.004173
3                           -0.000799                         0.002433
4                            0.001092                         0.003347
…                               …                                …
7370                         0.000587                         0.001415
7371                         0.000501                         0.001301
7372                        -0.000307                         0.001276
7373                        -0.000293                         0.001391
7374                         0.000476                         0.001335

      raw_acc:magnitude_stats:percentile25  \
0                                 0.976953
1                                 0.977402
2                                 0.977297
3                                 0.977607
4                                 0.977863
```

```
…                                    …
7370                          1.013690
7371                          1.013225
7372                          1.013541
7373                          1.013350
7374                          1.013529

      raw_acc:magnitude_stats:percentile50  \
0                                 0.978090
1                                 0.978380
2                                 0.978632
3                                 0.978664
4                                 0.978961
…                                    …
7370                              1.014335
7371                              1.013953
7372                              1.014194
7373                              1.013978
7374                              1.014236

      raw_acc:magnitude_stats:percentile75  \
0                                 0.979302
1                                 0.979202
2                                 0.980001
3                                 0.979654
4                                 0.979923
…                                    …
7370                              1.015096
7371                              1.014676
7372                              1.014796
7373                              1.014709
7374                              1.014933

      raw_acc:magnitude_stats:value_entropy  \
0                                  1.296142
1                                  2.174581
2                                  2.162259
3                                  2.282570
4                                  1.968584
…                                     …
7370                               2.438258
7371                               2.625195
7372                               2.497071
7373                               2.467187
7374                               2.475133

      raw_acc:magnitude_stats:time_entropy  …  label:STAIRS_-_GOING_DOWN  \
```

```
0                                6.684609  …                  NaN
1                                6.684610  …                  NaN
2                                6.684608  …                  NaN
3                                6.684610  …                  NaN
4                                6.684610  …                  NaN
…                                     …   …                …
7370                             6.684611  …                  NaN
7371                             6.684611  …                  NaN
7372                             6.684611  …                  NaN
7373                             6.684611  …                  NaN
7374                             6.684611  …                  NaN

      label:ELEVATOR  label:OR_standing  label:AT_SCHOOL  label:PHONE_IN_HAND  \
0                NaN                0.0              0.0                  NaN
1                NaN                0.0              0.0                  NaN
2                NaN                0.0              0.0                  NaN
3                NaN                0.0              0.0                  NaN
4                NaN                0.0              0.0                  NaN
…                  …                  …                …                    …
7370             NaN                0.0              0.0                  0.0
7371             NaN                0.0              0.0                  0.0
7372             NaN                0.0              0.0                  0.0
7373             NaN                0.0              0.0                  0.0
7374             NaN                0.0              0.0                  0.0

      label:PHONE_IN_BAG  label:PHONE_ON_TABLE  label:WITH_CO-WORKERS  \
0                    NaN                   NaN                    NaN
1                    NaN                   NaN                    NaN
2                    NaN                   NaN                    NaN
3                    NaN                   NaN                    NaN
4                    NaN                   NaN                    NaN
…                      …                     …                      …
7370                 NaN                   1.0                    NaN
7371                 NaN                   1.0                    NaN
7372                 NaN                   1.0                    NaN
7373                 NaN                   1.0                    NaN
7374                 NaN                   1.0                    NaN

      label:WITH_FRIENDS  label_source
0                    NaN             4
1                    NaN             4
2                    NaN             4
3                    NaN             4
4                    NaN             4
…                      …             …
7370                 NaN             1
7371                 NaN             1
```

```
7372                    NaN              1
7373                    NaN              1
7374                    NaN              1

[7375 rows x 278 columns]
```

1.0.3 remove NaN

[ ]:
```
location.isna().sum()
```

[ ]:
```
timestamp      0
latitude     515
longitude    515
dtype: int64
```

[ ]:
```
newlocation = location.interpolate(method='linear', axis=0).ffill().bfill()
print(newlocation.isna().sum())
newlocation
```

```
timestamp    0
latitude     0
longitude    0
dtype: int64
```

[ ]:
```
       timestamp   latitude   longitude
0     1440627472  32.882277 -117.234632
1     1440627533  32.882289 -117.234622
2     1440627593  32.882289 -117.234629
3     1440627654  32.882292 -117.234630
4     1440627712  32.882284 -117.234628
…              …          …           …
7370  1441292839  32.878956 -117.231589
7371  1441292931  32.878955 -117.231589
7372  1441292959  32.878956 -117.231589
7373  1441293052  32.878956 -117.231589
7374  1441293083  32.878956 -117.231589

[7375 rows x 3 columns]
```

[ ]:
```
location = newlocation
```

1.0.4 overlay on map

[ ]:
```
fig = go.Figure(go.Densitymapbox(lat=location.latitude, lon=location.longitude,
                                 radius=5))
avg_lat, avg_lon = np.mean(location.latitude), np.mean(location.longitude)
fig.update_layout(mapbox_style="open-street-map", mapbox_center_lon=avg_lon)
```

```
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0}, title = "Locations")
fig.show()
```

This person spent some time in the park at Costa Mesa. They drove along the San Diego Freeway and stopped at Irvine, where it looks like they waited for and took a light rail to San Juan. They continued on the light rail through Doheny State Beach and along the Pacific Coast Highway to Oceanside, where they stopped for a bit and waited for new passengers. They then continued to Solana Beach, their destination. In Solana Beach they hopped on a car or bus and headed down highway 101 to San Diego. They ended up at UC San Diego, where they spent some time walking around on campus, going to the various buildings, and making a trip to Trader Joe's for some food. They also went to visit a friend off campus in University City, which was pretty close to campus. To get to Trader Joe's it looks like they walked, but to the friend's house it looks like they drove and got slightly lost. Most likely they had too many things to carry from shopping and ordered an Uber or something.

[ ]:
```
in_car_sitting = feature_label.loc[feature_label['label:IN_A_CAR'].eq(1) &␣
↪feature_label['label:SITTING'].eq(1)]
print('In car sitting:', len(in_car_sitting))

sitting_surfing = feature_label.loc[feature_label['label:SURFING_THE_INTERNET'].
↪eq(1) & feature_label['label:SITTING'].eq(1)]
print('Sitting surfing internet:', len(sitting_surfing))
```

```
In car sitting: 36
Sitting surfing internet: 796
```

1.0.5 naive classifier surfing internet or driving

[ ]:
```
accel_cols = [l for l in list(feature_label) if l.startswith('raw_acc:')]

accel_cols
```

[ ]:
```
['raw_acc:magnitude_stats:mean',
 'raw_acc:magnitude_stats:std',
 'raw_acc:magnitude_stats:moment3',
 'raw_acc:magnitude_stats:moment4',
 'raw_acc:magnitude_stats:percentile25',
 'raw_acc:magnitude_stats:percentile50',
 'raw_acc:magnitude_stats:percentile75',
 'raw_acc:magnitude_stats:value_entropy',
 'raw_acc:magnitude_stats:time_entropy',
 'raw_acc:magnitude_spectrum:log_energy_band0',
 'raw_acc:magnitude_spectrum:log_energy_band1',
 'raw_acc:magnitude_spectrum:log_energy_band2',
 'raw_acc:magnitude_spectrum:log_energy_band3',
 'raw_acc:magnitude_spectrum:log_energy_band4',
 'raw_acc:magnitude_spectrum:spectral_entropy',
```

```
'raw_acc:magnitude_autocorrelation:period',
'raw_acc:magnitude_autocorrelation:normalized_ac',
'raw_acc:3d:mean_x',
'raw_acc:3d:mean_y',
'raw_acc:3d:mean_z',
'raw_acc:3d:std_x',
'raw_acc:3d:std_y',
'raw_acc:3d:std_z',
'raw_acc:3d:ro_xy',
'raw_acc:3d:ro_xz',
'raw_acc:3d:ro_yz']
```

[ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

rand_state = 42

driving_data = in_car_sitting[accel_cols]
surfing_data = sitting_surfing[accel_cols]
X = pd.concat([driving_data, surfing_data])
y = np.concatenate([np.zeros(len(driving_data)), np.ones(len(surfing_data))])

gnb = GaussianNB()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5,␣
↪random_state=rand_state)

model = gnb.fit(X_train, y_train)

y_pred = model.predict(X_test)

print('accuracy of naive bayes classifier:', accuracy_score(y_test, y_pred))
```

```
accuracy of naive bayes classifier: 0.9759615384615384
```

The accuracy of this naive bayes classifier is very high, at 0.976. This is likely due to overfitting, as there are only 36 data points of the person sitting in the car and driving. It would be interesting to see if you would get the same results for a second person as well.

1.0.6 distance travelled

[ ]:
```
import geopy.distance

driving_locations = location.loc[location['timestamp'].
↪isin(in_car_sitting['timestamp'])]

driving_distances = [0]
```

```python
last_row = None
for index, row in driving_locations.iterrows():
  if last_row is None:
    last_row = row
    continue

  prev_coords = last_row['latitude'], last_row['longitude']
  curr_coords = row['latitude'], row['longitude']
  distance = geopy.distance.geodesic(prev_coords, curr_coords).m
  driving_distances.append(distance)

  last_row = row

sorted(driving_distances)
```

[ ]:
```
[0,
 2.7744589817457035,
 5.488644107912544,
 6.15674568869783,
 18.733618019160968,
 29.52663933972229,
 92.88333530947557,
 135.144501768939,
 135.1447419718738,
 135.14498216927285,
 153.8707094159332,
 180.62687104966113,
 675.2745849094122,
 708.990292204785,
 729.8964279851472,
 729.8969603829447,
 729.8974928348063,
 729.8980253446023,
 729.8985579098505,
 729.8990905322992,
 729.899623209466,
 729.9001559452238,
 743.9948928008101,
 1090.8903142322233,
 1131.8747682798428,
 1292.566476764023,
 1452.107032065051,
 1470.2473217762185,
 1543.9954379164212,
 1559.7693640616005,
 1783.34054857259,
 2324.909420615315,
```

```
 2638.61943380624,
 2657.69551163442,
 3069.6220045857267,
 95559.42208817323]
```

[ ]:
```python
driving_threshold = 3500

for i, distance in enumerate(driving_distances):
  if distance > driving_threshold:
    driving_distances[i] = 0

driving_distances
```

[ ]:
```
[0,
 708.990292204785,
 1470.2473217762185,
 2657.69551163442,
 1452.107032065051,
 1090.8903142322233,
 3069.6220045857267,
 1131.8747682798428,
 743.9948928008101,
 1783.34054857259,
 1543.9954379164212,
 675.2745849094122,
 180.62687104966113,
 6.15674568869783,
 2.7744589817457035,
 5.488644107912544,
 29.52663933972229,
 0,
 2638.61943380624,
 1292.566476764023,
 2324.909420615315,
 1559.7693640616005,
 729.9001559452238,
 729.899623209466,
 729.8990905322992,
 729.8985579098505,
 729.8980253446023,
 729.8974928348063,
 729.8969603829447,
 729.8964279851472,
 153.8707094159332,
 135.14498216927285,
 135.1447419718738,
 135.144501768939,
```

```
18.733618019160968,
92.88333530947557]
```

After inspecting the driving data, I set the threshold to 3500m, or 3.5km. I chose this value because after sorting the distances, all of the values seemed to be linearly increasing and reasonably close to one another except for the largest value, at 95.560km. That was a significantly larger value than the next highest at 3km, so I set the threshold at slightly higher than the second highest value.

```
[ ]: import geopy.distance

surfing_locations = location.loc[location['timestamp'].
 ↪isin(sitting_surfing['timestamp'])]

surfing_distances = [0]

last_row = None
for index, row in surfing_locations.iterrows():
  if last_row is None:
    last_row = row
    continue

  prev_coords = last_row['latitude'], last_row['longitude']
  curr_coords = row['latitude'], row['longitude']
  distance = geopy.distance.geodesic(prev_coords, curr_coords).m
  surfing_distances.append(distance)

  last_row = row
```

```
[ ]: surfing_threshold = 5

for i, distance in enumerate(surfing_distances):
  if distance > surfing_threshold:
    surfing_distances[i] = 0
```

When inspecting the sorted surfing data, I found most of distances to be less than 1 meter. When increased to 5 meters, this represents 615 / 795 samples. I chose this number because it represents most of the data, and the distances should be close to 0.

1.0.7 classifier

```
[ ]: # ignore warning
import warnings
from pandas.core.common import SettingWithCopyWarning
warnings.simplefilter(action="ignore", category=SettingWithCopyWarning)

driving_data = in_car_sitting[accel_cols]
driving_data['distance'] = driving_distances
surfing_data = sitting_surfing[accel_cols]
```

```
surfing_data['distance'] = surfing_distances

X = pd.concat([driving_data, surfing_data])
y = np.concatenate([np.zeros(len(driving_data)), np.ones(len(surfing_data))])

gnb = GaussianNB()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5,␣
 ↪random_state=rand_state)

model = gnb.fit(X_train, y_train)

y_pred = model.predict(X_test)

print('accuracy of naive bayes classifier:', accuracy_score(y_test, y_pred))
```

```
accuracy of naive bayes classifier: 0.9807692307692307
```

After adding this distance column and normalizing it (removing any outliers), the accuracy of the model increased from 0.976 to 0.981, which is a small but noticeable improvement. The accuracy of both models is very high, and again may be due to overfitting. However, it is able to correctly characterize this simple two-category system, so maybe it will be accurate with other datasets as well.

1.1 Audio

1.1.1 get samples

[ ]:
```
sitting_eating = feature_label.loc[feature_label['label:SITTING'].eq(1) &␣
 ↪feature_label['label:EATING'].eq(1)]
print('Sitting eating:', len(sitting_eating))

sitting_working = feature_label.loc[feature_label['label:SITTING'].eq(1) &␣
 ↪feature_label['label:COMPUTER_WORK'].eq(1)]
print('Sitting computer work:', len(sitting_working))
```

```
Sitting eating: 425
Sitting computer work: 1622
```

1.1.2 show location

[ ]:
```
geolocator = Nominatim(user_agent="http")
```

Eating Locations

[ ]:
```
eating_locations = location.loc[location['timestamp'].
 ↪isin(sitting_eating['timestamp'])]
eating_locations
```

```
[ ]:        timestamp   latitude   longitude
     364   1440651089  32.878951 -117.231531
     365   1440651171  32.878947 -117.231530
     366   1440651214  32.878943 -117.231530
     367   1440651276  32.878943 -117.231529
     368   1440651352  32.878939 -117.231532
     …          …          …          …
     6372  1441230624  32.878962 -117.231519
     6373  1441230685  32.878962 -117.231521
     6374  1441230749  32.878961 -117.231521
     6375  1441230804  32.878961 -117.231522
     6376  1441230864  32.878961 -117.231522

     [425 rows x 3 columns]
```

```
[ ]: fig = go.Figure(go.Densitymapbox(lat=eating_locations.latitude,␣
      ↪lon=eating_locations.longitude,
                                         radius=5))
     avg_lat, avg_lon = np.mean(eating_locations.latitude), np.mean(eating_locations.
      ↪longitude)
     fig.update_layout(mapbox_style="open-street-map", mapbox_center_lon=avg_lon)
     fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0}, title = "Eating Locations")
     fig.show()
```

```
[ ]: clusters = [(33.667241, -117.899997), (33.446898, -117.652734), (33.088184,␣
      ↪-117.309919), \
                 (32.879628552, -117.233454544), (32.869896, -117.233372), (32.
      ↪864965, -117.232524)]

     for i, (lat, lng) in enumerate(clusters):
       location_data = geolocator.reverse(f"{lat},{lng}")
       print(f'location {i + 1}:', location_data)
```

```
location 1: OC Promenade, 1050, Arlington Drive, Costa Mesa, Orange County,
California, 92626, United States
location 2: 35581, Beach Road, Capistrano Beach, Dana Point, Orange County,
California, 92624, United States
location 3: Leucadia, Encinitas, San Diego County, California, 92011, United
States
location 4: University of California, San Diego, 9500, Gilman Drive, University
City, San Diego, San Diego County, California, 92093, United States
location 5: 3211, Holiday Court, San Diego, San Diego County, California, 92037,
United States
location 6: La Jolla Village Square Shopping Center, 8657, Villa La Jolla Drive,
La Jolla Colony, San Diego, San Diego County, California, 92037, United States
```

I did manual clustering to find these regions of interest. The addresses of the six centroids are shown above. Some of these centroids are along highways / roads, signalling that the person was either driving or walking around while eating. Then there are centroids in UC San Diego and Costa Mesa, showing that this person was eating while studying and at the park. Looking at the map and centroids together, it is clear that the main areas this person is eating is in the car traveling, when studying and when at the park.

working locations

[ ]:
```
working_locations = location.loc[location['timestamp'].
 ↪isin(sitting_working['timestamp'])]
working_locations
```

[ ]:
```
       timestamp   latitude   longitude
391   1440652787  32.878937 -117.231535
392   1440652843  32.878937 -117.231535
393   1440652905  32.878937 -117.231535
394   1440652965  32.878937 -117.231535
395   1440653047  32.878936 -117.231536
…            …          …           …
6985  1441267956  32.878963 -117.231529
6986  1441267979  32.878962 -117.231529
6987  1441268169  32.878962 -117.231530
6988  1441268228  32.878962 -117.231530
6989  1441268302  32.878962 -117.231530

[1622 rows x 3 columns]
```

[ ]:
```
fig = go.Figure(go.Densitymapbox(lat=working_locations.latitude,␣
 ↪lon=working_locations.longitude,
                                 radius=5))
avg_lat, avg_lon = np.mean(working_locations.latitude), np.
 ↪mean(working_locations.longitude)
fig.update_layout(mapbox_style="open-street-map", mapbox_center_lon=avg_lon)
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0}, title = "Working Locations")
fig.show()
```

[ ]:
```
clusters = [(32.881075, -117.238147), (32.881964, -117.240459), (32.8803604,␣
 ↪-117.2363628), \
            (32.87998685, -117.23503884), (32.87949695, -117.233306), (32.
 ↪87902, -117.231626), \
            (32.879414, -117.235188)]

for i, (lat, lng) in enumerate(clusters):
  location_data = geolocator.reverse(f"{lat},{lng}")
  print(f'location {i + 1}:', location_data)
```

```
location 1: Geisel Library, Library Walk, University Center, Torrey Pines, San
Diego, San Diego County, California, 92093-0068, United States
location 2: Economics Building, Ridge Walk, Thurgood Marshall College, Torrey
```

```
Pines, San Diego, San Diego County, California, 92093, United States
location 3: University of California, San Diego, 9500, Gilman Drive, University
City, San Diego, San Diego County, California, 92093, United States
location 4: University of California, San Diego, 9500, Gilman Drive, University
City, San Diego, San Diego County, California, 92093, United States
location 5: University of California, San Diego, 9500, Gilman Drive, University
City, San Diego, San Diego County, California, 92093, United States
location 6: A, Innovation Lane, Pepper Canyon, San Diego, San Diego County,
California, 92093-0068, United States
location 7: University of California, San Diego, 9500, Gilman Drive, University
City, San Diego, San Diego County, California, 92093, United States
```

Looking at the above cluster centroids, computer work is done exclusively in UC San Diego. Some work is done outside in the park by the Geisel Library, while other work is done in the Science and Engineering or Structural and Mechanical Engineering buildings. Work is also done in some smaller buildings scattered throughout campus. This person presumably was using a laptop to go to these different places to work.

Audio

[ ]:
```
mfcc_extension = ".sound.mfcc"
mfcc = "./data/person2-MFCC-1DBB0F6F-1F81-4A50-9DF4-CD62ACFA4842/"

def get_audio_data(timestamp):
  df = pd.read_csv(mfcc + str(timestamp) + mfcc_extension, sep=",",␣
↪header=None, )
  df = df.drop(columns=[13])
  mfcc_coeff = df.T.to_numpy()
  return mfcc_coeff

get_audio_data(1440652787)
```

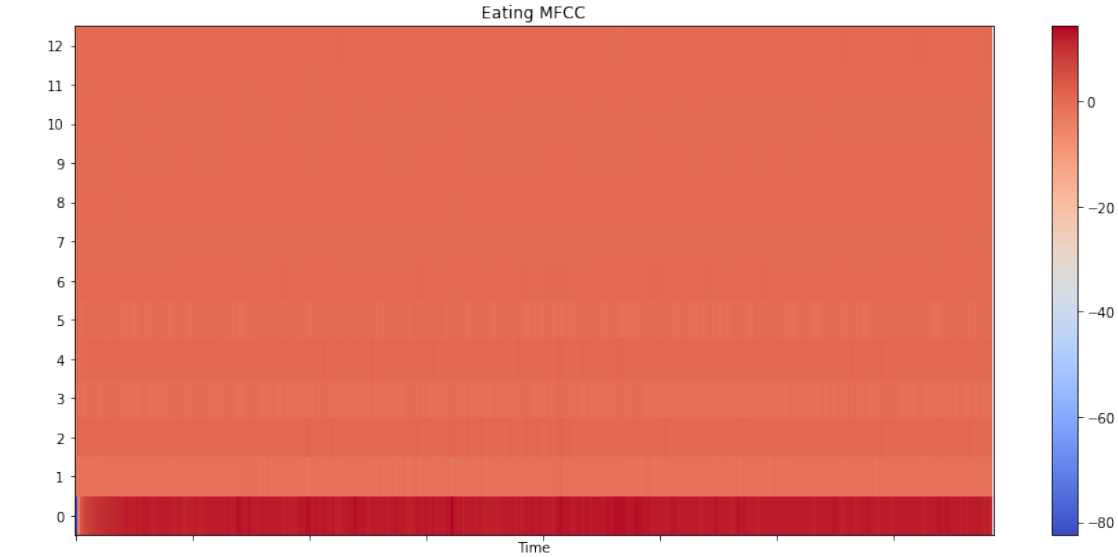
[ ]:
```
array([[-8.24621125e+01, -9.93218702e+00, -4.97305810e+00, …,
         2.14531050e+00,  2.66927492e+00,  1.51974319e+00],
       [ 8.61659633e-16, -1.19795986e+00, -1.28985223e+00, …,
        -5.51409700e-01, -9.40378632e-01, -6.41590596e-01],
       [-4.30829816e-16,  5.51628363e-01,  5.65086712e-01, …,
         6.14282306e-01,  5.58024678e-01,  8.11163369e-01],
       …,
       [ 5.16995780e-15, -2.67070573e-01, -2.49183667e-01, …,
        -1.69268257e-01, -3.08869254e-01, -2.23282110e-01],
       [ 8.18576651e-15, -1.13510040e-01, -1.27752455e-01, …,
        -1.50372953e-01, -1.58451437e-01, -1.95369635e-01],
       [-1.16324050e-14,  2.10808326e-01,  1.17075395e-01, …,
        -3.44395678e-04,  3.65590400e-02, -4.13183255e-02]])
```

[ ]:
```
timestamp = 1440651352
```

```
mfcc_coeff = get_audio_data(timestamp)
mel_data = librosa.feature.inverse.mfcc_to_mel(mfcc_coeff, n_mels=13)

plt.rcParams["figure.figsize"] = (15, 15)

fig, ax = plt.subplots(nrows=2, sharex=True)
img = librosa.display.specshow(librosa.power_to_db(mel_data, ref=np.max), sr =
 ↪22000,
                              x_axis='time', y_axis='mel', fmax=10000,
                              ax=ax[1])
fig.colorbar(img, ax=[ax[1]])
ax[1].set(title='Eating Mel spectrogram')
ax[1].label_outer()
ax[1].set_ylabel('Mel')


img = librosa.display.specshow(mfcc_coeff, x_axis='time', ax=ax[0])
fig.colorbar(img, ax=[ax[0]])
ax[0].set(title='Eating MFCC')
ax[0].set_yticks(np.linspace(0, 12, 13))
```

[ ]:

```
[<matplotlib.axis.YTick at 0x7f492c082380>,
 <matplotlib.axis.YTick at 0x7f492c081c00>,
 <matplotlib.axis.YTick at 0x7f492c0a62f0>,
 <matplotlib.axis.YTick at 0x7f492bf4b8b0>,
 <matplotlib.axis.YTick at 0x7f492bf4b3d0>,
 <matplotlib.axis.YTick at 0x7f492bf744c0>,
 <matplotlib.axis.YTick at 0x7f492bf74c10>,
 <matplotlib.axis.YTick at 0x7f492bf75360>,
 <matplotlib.axis.YTick at 0x7f492bf75ab0>,
 <matplotlib.axis.YTick at 0x7f492bf76200>,
 <matplotlib.axis.YTick at 0x7f492bf76860>,
 <matplotlib.axis.YTick at 0x7f492bf753c0>,
 <matplotlib.axis.YTick at 0x7f492bf768c0>]
```

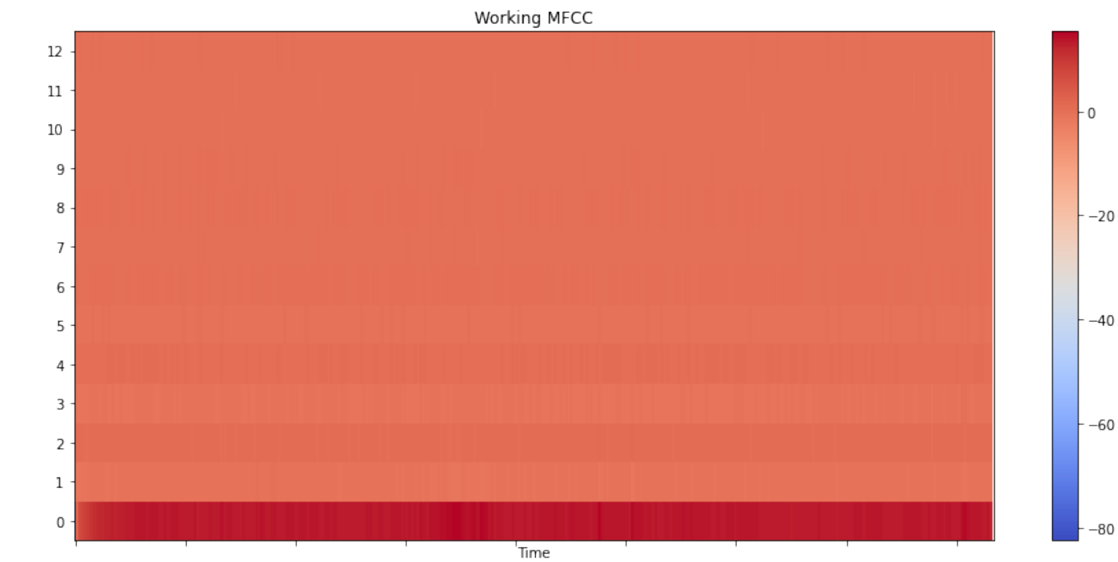


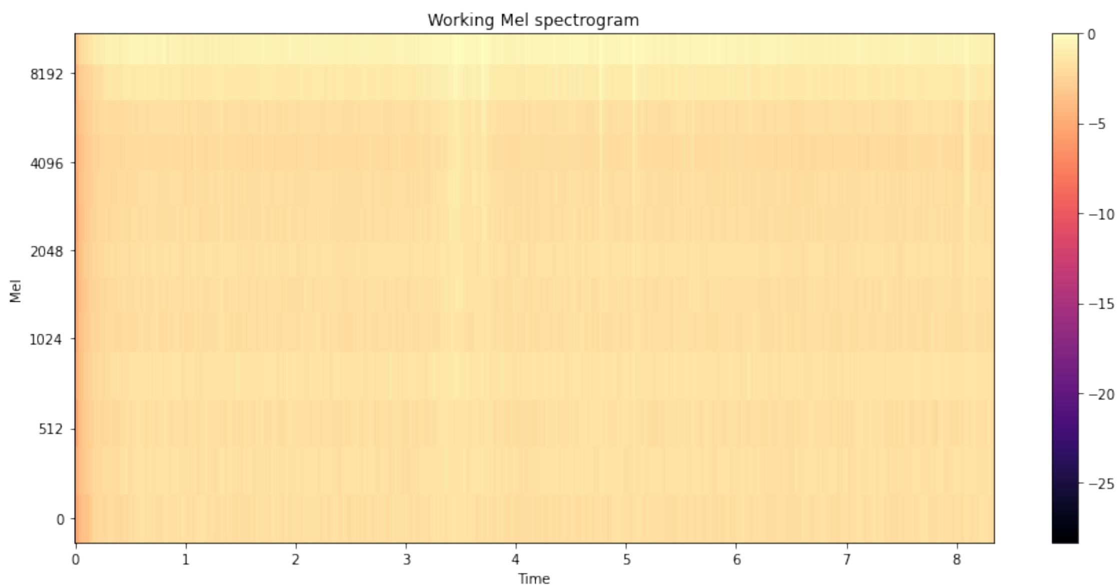
[ ]:
```
timestamp = 1440653047

mfcc_coeff = get_audio_data(timestamp)
mel_data = librosa.feature.inverse.mfcc_to_mel(mfcc_coeff, n_mels=13)

plt.rcParams["figure.figsize"] = (15, 15)

fig, ax = plt.subplots(nrows=2, sharex=True)
img = librosa.display.specshow(librosa.power_to_db(mel_data, ref=np.max), sr =␣
↪22000,
                               x_axis='time', y_axis='mel', fmax=10000,
```

```
                                    ax=ax[1])
fig.colorbar(img, ax=[ax[1]])
ax[1].set(title='Working Mel spectrogram')
ax[1].label_outer()
ax[1].set_ylabel('Mel')


img = librosa.display.specshow(mfcc_coeff, x_axis='time', ax=ax[0])
fig.colorbar(img, ax=[ax[0]])
ax[0].set(title='Working MFCC')
ax[0].set_yticks(np.linspace(0, 12, 13))
```

[ ]:

```
[<matplotlib.axis.YTick at 0x7f492c10d810>,
 <matplotlib.axis.YTick at 0x7f492c10d090>,
 <matplotlib.axis.YTick at 0x7f492c125780>,
 <matplotlib.axis.YTick at 0x7f492c1cae30>,
 <matplotlib.axis.YTick at 0x7f492c1ca950>,
 <matplotlib.axis.YTick at 0x7f492c1cba00>,
 <matplotlib.axis.YTick at 0x7f492c1fc190>,
 <matplotlib.axis.YTick at 0x7f492c1fc8e0>,
 <matplotlib.axis.YTick at 0x7f492c1fd030>,
 <matplotlib.axis.YTick at 0x7f492c1fd780>,
 <matplotlib.axis.YTick at 0x7f492c1cba60>,
 <matplotlib.axis.YTick at 0x7f492c1fd2d0>,
 <matplotlib.axis.YTick at 0x7f492c1fde40>]
```





classifiers

[ ]:
```
eating_location_data = []

for _, curr_location in eating_locations.iterrows():
  lat = curr_location['latitude']
  lng = curr_location['longitude']
  location_data = geolocator.reverse(f"{lat},{lng}")
  eating_location_data.append(location_data)
```

[ ]:
```
df = pd.DataFrame(eating_location_data)
df.to_csv('eating_location_data.csv')
```

[ ]:
```
working_location_data = []

for _, curr_location in working_locations.iterrows():
  lat = curr_location['latitude']
  lng = curr_location['longitude']
  location_data = geolocator.reverse(f"{lat},{lng}")
  working_location_data.append(location_data)
```

[ ]:
```
df = pd.DataFrame(working_location_data)
df.to_csv('working_location_data.csv')
```

encode city

[ ]:
```
cities_eating = [el[0].split(',')[-5] for el in eating_location_data]
sitting_eating['city'] = cities_eating
print(sitting_eating['city'])

cities_working = [el[0].split(',')[-5] for el in working_location_data]
sitting_working['city'] = cities_working
print(sitting_working['city'])

all_encoded = pd.get_dummies(pd.concat([sitting_eating['city'],␣
 ↪sitting_working['city']]))
all_encoded
```

```
364      San Diego
365      San Diego
366      San Diego
367      San Diego
368      San Diego
           …
6372     San Diego
6373     San Diego
6374     San Diego
6375     San Diego
6376     San Diego
Name: city, Length: 425, dtype: object
391      San Diego
392      San Diego
393      San Diego
394      San Diego
395      San Diego
           …
6985     San Diego
6986     San Diego
```

```
6987     San Diego
6988     San Diego
6989     San Diego
Name: city, Length: 1622, dtype: object
```

[ ]:

```
      Agra  Carlsbad  Costa Mesa  Dana Point  Encinitas  Oceanside  \
364      0         0           0           0          0          0
365      0         0           0           0          0          0
366      0         0           0           0          0          0
367      0         0           0           0          0          0
368      0         0           0           0          0          0
…      …        …          …          …         …        …
6985     0         0           0           0          0          0
6986     0         0           0           0          0          0
6987     0         0           0           0          0          0
6988     0         0           0           0          0          0
6989     0         0           0           0          0          0

      San Clemente  San Diego  San Juan Capistrano  Solana Beach  \
364              0          1                    0             0
365              0          1                    0             0
366              0          1                    0             0
367              0          1                    0             0
368              0          1                    0             0
…              …        …                   …           …
6985             0          1                    0             0
6986             0          1                    0             0
6987             0          1                    0             0
6988             0          1                    0             0
6989             0          1                    0             0

      Torrey Pines  Encinitas
364              0          0
365              0          0
366              0          0
367              0          0
368              0          0
…              …        …
6985             0          0
6986             0          0
6987             0          0
6988             0          0
6989             0          0

[2047 rows x 12 columns]
```

[ ]:
```
curr_encoded = all_encoded.iloc[0: len(sitting_eating)]

location_sitting_eating = sitting_eating.join(curr_encoded)
location_sitting_eating = location_sitting_eating.drop('city', axis = 1)
location_sitting_eating
```

[ ]:
```
       timestamp  raw_acc:magnitude_stats:mean  raw_acc:magnitude_stats:std  \
364   1440651089                      0.969981                     0.024200
365   1440651171                      0.973126                     0.001047
366   1440651214                      0.973005                     0.001070
367   1440651276                      0.973191                     0.001023
368   1440651352                      0.973705                     0.001043
…            …                             …                            …
6372  1441230624                      0.978159                     0.057358
6373  1441230685                      0.970195                     0.012532
6374  1441230749                      0.972170                     0.023283
6375  1441230804                      0.973395                     0.039125
6376  1441230864                      0.968523                     0.018076

      raw_acc:magnitude_stats:moment3  raw_acc:magnitude_stats:moment4  \
364                          0.018224                         0.041288
365                         -0.000596                         0.001441
366                         -0.000624                         0.001499
367                         -0.000470                         0.001369
368                         -0.000550                         0.001382
…                                 …                                …
6372                         0.055600                         0.121728
6373                        -0.006951                         0.019416
6374                         0.019654                         0.038548
6375                         0.044037                         0.069762
6376                        -0.011204                         0.027938

      raw_acc:magnitude_stats:percentile25  \
364                               0.959713
365                               0.972483
366                               0.972342
367                               0.972524
368                               0.972999
…                                      …
6372                              0.962143
6373                              0.963655
6374                              0.962927
6375                              0.955612
6376                              0.960779

      raw_acc:magnitude_stats:percentile50  \
364                               0.969943
```

```
365                                0.973123
366                                0.973035
367                                0.973215
368                                0.973745
…                                     …
6372                               0.976692
6373                               0.970648
6374                               0.970257
6375                               0.971782
6376                               0.968867

      raw_acc:magnitude_stats:percentile75  \
364                                0.979780
365                                0.973814
366                                0.973673
367                                0.973904
368                                0.974385
…                                     …
6372                               0.988472
6373                               0.976588
6374                               0.978569
6375                               0.989002
6376                               0.977444

      raw_acc:magnitude_stats:value_entropy  \
364                                 1.999294
365                                 2.472657
366                                 2.341777
367                                 2.419986
368                                 2.463549
…                                      …
6372                                1.278916
6373                                2.124745
6374                                1.952924
6375                                1.867990
6376                                2.131921

      raw_acc:magnitude_stats:time_entropy  …   Costa Mesa   Dana Point  \
364                                6.684301  …            0            0
365                                6.684611  …            0            0
366                                6.684611  …            0            0
367                                6.684611  …            0            0
368                                6.684611  …            0            0
…                                     …   …          …          …
6372                               6.682903  …            0            0
6373                               6.684528  …            0            0
6374                               6.684326  …            0            0
```

```
6375                              6.683817  …          0            0
6376                              6.684437  …          0            0

       Encinitas  Oceanside  San Clemente  San Diego  San Juan Capistrano  \
364            0          0             0          1                    0
365            0          0             0          1                    0
366            0          0             0          1                    0
367            0          0             0          1                    0
368            0          0             0          1                    0
…            …          …             …          …                    …
6372           0          0             0          1                    0
6373           0          0             0          1                    0
6374           0          0             0          1                    0
6375           0          0             0          1                    0
6376           0          0             0          1                    0

       Solana Beach  Torrey Pines  Encinitas
364               0             0          0
365               0             0          0
366               0             0          0
367               0             0          0
368               0             0          0
…               …             …          …
6372              0             0          0
6373              0             0          0
6374              0             0          0
6375              0             0          0
6376              0             0          0

[425 rows x 290 columns]
```

[ ]:
```
curr_encoded = all_encoded.iloc[len(sitting_eating):]

location_sitting_working = sitting_working.join(curr_encoded)
location_sitting_working = location_sitting_working.drop('city', axis = 1)
location_sitting_working
```

[ ]:
```
       timestamp  raw_acc:magnitude_stats:mean  raw_acc:magnitude_stats:std  \
391   1440652787                      0.981915                     0.123744
392   1440652843                      0.978268                     0.007662
393   1440652905                      0.978518                     0.003635
394   1440652965                      0.974498                     0.001502
395   1440653047                      0.975216                     0.001259
…            …                           …                            …
6985  1441267956                      0.981202                     0.001351
6986  1441267979                      0.987082                     0.119474
6987  1441268169                      0.979582                     0.001486
```

6988	1441268228	0.979483	0.015567
6989	1441268302	0.994586	0.104949

	raw_acc:magnitude_stats:moment3	raw_acc:magnitude_stats:moment4 \
391	0.196262	0.289781
392	-0.007976	0.014020
393	-0.001689	0.005214
394	-0.000660	0.002001
395	-0.000859	0.001683
…	…	…
6985	-0.001001	0.001867
6986	0.201659	0.304354
6987	0.000683	0.001977
6988	0.019237	0.034301
6989	0.119046	0.206257

	raw_acc:magnitude_stats:percentile25 \
391	0.963881
392	0.975242
393	0.976515
394	0.973573
395	0.974478
…	…
6985	0.980383
6986	0.964714
6987	0.978601
6988	0.977144
6989	0.973515

	raw_acc:magnitude_stats:percentile50 \
391	0.974787
392	0.978618
393	0.978534
394	0.974486
395	0.975278
…	…
6985	0.981272
6986	0.980208
6987	0.979555
6988	0.979314
6989	0.980731

	raw_acc:magnitude_stats:percentile75 \
391	0.979273
392	0.981514
393	0.980594
394	0.975519

395	0.976036
…	…
6985	0.982141
6986	0.988380
6987	0.980577
6988	0.981152
6989	0.999178

	raw_acc:magnitude_stats:value_entropy \
391	1.019330
392	1.862406
393	2.306569
394	2.578886
395	2.636655
…	…
6985	2.466110
6986	0.904069
6987	2.551983
6988	1.358389
6989	1.422944

	raw_acc:magnitude_stats:time_entropy	…	Costa Mesa	Dana Point \
391	6.677581	…	0	0
392	6.684581	…	0	0
393	6.684605	…	0	0
394	6.684611	…	0	0
395	6.684611	…	0	0
…	…	…	…	…
6985	6.684611	…	0	0
6986	6.678204	…	0	0
6987	6.684611	…	0	0
6988	6.684487	…	0	0
6989	6.679195	…	0	0

	Encinitas	Oceanside	San Clemente	San Diego	San Juan Capistrano \
391	0	0	0	1	0
392	0	0	0	1	0
393	0	0	0	1	0
394	0	0	0	1	0
395	0	0	0	1	0
…	…	…	…	…	…
6985	0	0	0	1	0
6986	0	0	0	1	0
6987	0	0	0	1	0
6988	0	0	0	1	0
6989	0	0	0	1	0

```
      Solana Beach  Torrey Pines  Encinitas
391              0             0          0
392              0             0          0
393              0             0          0
394              0             0          0
395              0             0          0
…              …             …          …
6985             0             0          0
6986             0             0          0
6987             0             0          0
6988             0             0          0
6989             0             0          0

[1622 rows x 290 columns]
```

geoencoding model

```
[ ]: # acceleration and city
cols = [*accel_cols, *list(all_encoded)]

eating_data = location_sitting_eating[cols]
working_data = location_sitting_working[cols]

X = pd.concat([eating_data, working_data])
y = np.concatenate([np.zeros(len(eating_data)), np.ones(len(working_data))])

gnb = GaussianNB()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5,␣
 ↪random_state=rand_state)

model = gnb.fit(X_train, y_train)

y_pred = model.predict(X_test)

print('accuracy of naive bayes classifier:', accuracy_score(y_test, y_pred))
```

```
accuracy of naive bayes classifier: 0.818359375
```

mfcc get data

```
[ ]: mfcc_data_arr = []

for _, row in sitting_eating.iterrows():
  timestamp = row['timestamp']
  mfcc_coeff = get_audio_data(timestamp)

  flat_data = mfcc_coeff.flatten()
  mfcc_data_arr.append(flat_data)
```

```python
mfcc_eating_df = pd.DataFrame(mfcc_data_arr).fillna(0)

curr_data = sitting_eating.reset_index(drop=True)

mfcc_eating_data = pd.concat([curr_data, mfcc_eating_df], axis=1)

mfcc_eating_data
```

[ ]:

```
      timestamp  raw_acc:magnitude_stats:mean  raw_acc:magnitude_stats:std  \
0    1440651089                      0.969981                     0.024200
1    1440651171                      0.973126                     0.001047
2    1440651214                      0.973005                     0.001070
3    1440651276                      0.973191                     0.001023
4    1440651352                      0.973705                     0.001043
..          …                           …                            …
420  1441230624                      0.978159                     0.057358
421  1441230685                      0.970195                     0.012532
422  1441230749                      0.972170                     0.023283
423  1441230804                      0.973395                     0.039125
424  1441230864                      0.968523                     0.018076

     raw_acc:magnitude_stats:moment3  raw_acc:magnitude_stats:moment4  \
0                           0.018224                         0.041288
1                          -0.000596                         0.001441
2                          -0.000624                         0.001499
3                          -0.000470                         0.001369
4                          -0.000550                         0.001382
..                              …                                …
420                         0.055600                         0.121728
421                        -0.006951                         0.019416
422                         0.019654                         0.038548
423                         0.044037                         0.069762
424                        -0.011204                         0.027938

     raw_acc:magnitude_stats:percentile25  \
0                                0.959713
1                                0.972483
2                                0.972342
3                                0.972524
4                                0.972999
..                                   …
420                              0.962143
421                              0.963655
422                              0.962927
423                              0.955612
424                              0.960779
```

```
     raw_acc:magnitude_stats:percentile50  \
0                                0.969943
1                                0.973123
2                                0.973035
3                                0.973215
4                                0.973745
..                                    …
420                              0.976692
421                              0.970648
422                              0.970257
423                              0.971782
424                              0.968867

     raw_acc:magnitude_stats:percentile75  \
0                                0.979780
1                                0.973814
2                                0.973673
3                                0.973904
4                                0.974385
..                                    …
420                              0.988472
421                              0.976588
422                              0.978569
423                              0.989002
424                              0.977444

     raw_acc:magnitude_stats:value_entropy  \
0                                 1.999294
1                                 2.472657
2                                 2.341777
3                                 2.419986
4                                 2.463549
..                                     …
420                               1.278916
421                               2.124745
422                               1.952924
423                               1.867990
424                               2.131921

     raw_acc:magnitude_stats:time_entropy  …  14966  14967  14968  14969  \
0                                6.684301  …    0.0    0.0    0.0    0.0
1                                6.684611  …    0.0    0.0    0.0    0.0
2                                6.684611  …    0.0    0.0    0.0    0.0
3                                6.684611  …    0.0    0.0    0.0    0.0
4                                6.684611  …    0.0    0.0    0.0    0.0
..                                    …   …     …      …     …      …
```

```
420                                   6.682903  ...   0.0   0.0   0.0   0.0
421                                   6.684528  ...   0.0   0.0   0.0   0.0
422                                   6.684326  ...   0.0   0.0   0.0   0.0
423                                   6.683817  ...   0.0   0.0   0.0   0.0
424                                   6.684437  ...   0.0   0.0   0.0   0.0

     14970  14971  14972  14973  14974  14975
0      0.0    0.0    0.0    0.0    0.0    0.0
1      0.0    0.0    0.0    0.0    0.0    0.0
2      0.0    0.0    0.0    0.0    0.0    0.0
3      0.0    0.0    0.0    0.0    0.0    0.0
4      0.0    0.0    0.0    0.0    0.0    0.0
..     ...    ...    ...    ...    ...    ...
420    0.0    0.0    0.0    0.0    0.0    0.0
421    0.0    0.0    0.0    0.0    0.0    0.0
422    0.0    0.0    0.0    0.0    0.0    0.0
423    0.0    0.0    0.0    0.0    0.0    0.0
424    0.0    0.0    0.0    0.0    0.0    0.0

[425 rows x 15255 columns]
```

[ ]:

```python
# no data for timestamp 5389
no_data = [1441156901]
mfcc_data_arr = []

for i, row in sitting_working.iterrows():
  timestamp = row['timestamp']
  if timestamp in no_data:
    continue
  mfcc_coeff = get_audio_data(timestamp)

  flat_data = mfcc_coeff.flatten()
  mfcc_data_arr.append(flat_data)

mfcc_working_df = pd.DataFrame(mfcc_data_arr).fillna(0)

curr_data = sitting_working.drop(sitting_working[sitting_working['timestamp'].
 ↪isin(no_data)].index).reset_index(drop=True)

mfcc_working_data = pd.concat([curr_data, mfcc_working_df], axis=1)

mfcc_working_data
```

[ ]:

```
      timestamp  raw_acc:magnitude_stats:mean  raw_acc:magnitude_stats:std  \
0    1440652787                      0.981915                     0.123744
1    1440652843                      0.978268                     0.007662
2    1440652905                      0.978518                     0.003635
```

```
3     1440652965                    0.974498                    0.001502
4     1440653047                    0.975216                    0.001259
…          …                         …                          …
1616  1441267956                    0.981202                    0.001351
1617  1441267979                    0.987082                    0.119474
1618  1441268169                    0.979582                    0.001486
1619  1441268228                    0.979483                    0.015567
1620  1441268302                    0.994586                    0.104949

      raw_acc:magnitude_stats:moment3  raw_acc:magnitude_stats:moment4  \
0                            0.196262                         0.289781
1                           -0.007976                         0.014020
2                           -0.001689                         0.005214
3                           -0.000660                         0.002001
4                           -0.000859                         0.001683
…                               …                               …
1616                        -0.001001                         0.001867
1617                         0.201659                         0.304354
1618                         0.000683                         0.001977
1619                         0.019237                         0.034301
1620                         0.119046                         0.206257

      raw_acc:magnitude_stats:percentile25  \
0                                 0.963881
1                                 0.975242
2                                 0.976515
3                                 0.973573
4                                 0.974478
…                                    …
1616                              0.980383
1617                              0.964714
1618                              0.978601
1619                              0.977144
1620                              0.973515

      raw_acc:magnitude_stats:percentile50  \
0                                 0.974787
1                                 0.978618
2                                 0.978534
3                                 0.974486
4                                 0.975278
…                                    …
1616                              0.981272
1617                              0.980208
1618                              0.979555
1619                              0.979314
1620                              0.980731
```

```
      raw_acc:magnitude_stats:percentile75  \
0                                 0.979273
1                                 0.981514
2                                 0.980594
3                                 0.975519
4                                 0.976036
…                                    …
1616                              0.982141
1617                              0.988380
1618                              0.980577
1619                              0.981152
1620                              0.999178

      raw_acc:magnitude_stats:value_entropy  \
0                                  1.019330
1                                  1.862406
2                                  2.306569
3                                  2.578886
4                                  2.636655
…                                     …
1616                               2.466110
1617                               0.904069
1618                               2.551983
1619                               1.358389
1620                               1.422944

      raw_acc:magnitude_stats:time_entropy  …  11365  11366  11367  11368  \
0                                 6.677581  …    0.0    0.0    0.0    0.0
1                                 6.684581  …    0.0    0.0    0.0    0.0
2                                 6.684605  …    0.0    0.0    0.0    0.0
3                                 6.684611  …    0.0    0.0    0.0    0.0
4                                 6.684611  …    0.0    0.0    0.0    0.0
…                                    …  …    …    …    …    …
1616                              6.684611  …    0.0    0.0    0.0    0.0
1617                              6.678204  …    0.0    0.0    0.0    0.0
1618                              6.684611  …    0.0    0.0    0.0    0.0
1619                              6.684487  …    0.0    0.0    0.0    0.0
1620                              6.679195  …    0.0    0.0    0.0    0.0

      11369  11370  11371  11372  11373  11374
0       0.0    0.0    0.0    0.0    0.0    0.0
1       0.0    0.0    0.0    0.0    0.0    0.0
2       0.0    0.0    0.0    0.0    0.0    0.0
3       0.0    0.0    0.0    0.0    0.0    0.0
4       0.0    0.0    0.0    0.0    0.0    0.0
…     …    …    …    …    …    …
```

```
1616    0.0    0.0    0.0    0.0    0.0    0.0
1617    0.0    0.0    0.0    0.0    0.0    0.0
1618    0.0    0.0    0.0    0.0    0.0    0.0
1619    0.0    0.0    0.0    0.0    0.0    0.0
1620    0.0    0.0    0.0    0.0    0.0    0.0

[1621 rows x 11654 columns]
```

mfcc model

```
[ ]: # acceleration and mfcc data

cols = [*accel_cols, *list(mfcc_eating_df)]
eating_data = mfcc_eating_data[cols]
cols = [*accel_cols, *list(mfcc_working_df)]
working_data = mfcc_working_data[cols]

X = pd.concat([eating_data, working_data]).fillna(0)
y = np.concatenate([np.zeros(len(eating_data)), np.ones(len(working_data))])

gnb = GaussianNB()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5,␣
↪random_state=rand_state)

model = gnb.fit(X_train, y_train)

y_pred = model.predict(X_test)

print('accuracy of naive bayes classifier:', accuracy_score(y_test, y_pred))
```

```
/home/joshua/Desktop/cornell/ubicomp/hw/a3/.venv/lib/python3.10/site-
packages/sklearn/utils/validation.py:1688: FutureWarning:

Feature names only support names that are all strings. Got feature names with
dtypes: ['int', 'str']. An error will be raised in 1.2.


accuracy of naive bayes classifier: 0.7771260997067448

/home/joshua/Desktop/cornell/ubicomp/hw/a3/.venv/lib/python3.10/site-
packages/sklearn/utils/validation.py:1688: FutureWarning:

Feature names only support names that are all strings. Got feature names with
dtypes: ['int', 'str']. An error will be raised in 1.2.
```

The two models performed fairly similarly, with the geoencoded model being 0.818% accurate and the mfcc model being 0.777% accurate. The geoencoding model only used the name of the city that the user was in, while the mfcc model used all of the data from the audio samples. The mfcc model likely did not have as good performance for a number of reasons. First, there is usually a lot of noise in these samples that we are not filtering out, and while we are trying to get ambient noise, this data could be processed further to distill the signal that has the best representation of this noise. For the geoencoding model, we are again only using the city as an attribute, which is not ideal in this classification model. Ideally, we would use the type of location, whether it is a park, an office, a restaurant or a cafeteria. Classifying the locations into these categories is a different problem in itself, which is why I thought it was out of scope for this assignment. I think that the geoencoded model would be better to continue with for future exploration, despite the audio model also showing some promise.