Guided Project Report

Human Activity Recognition

Name: Shruti Verma
Course: AI and ML
(Batch 4)
Duration: 10 months

Problem Statement:Perform activity recognition on the dataset using a hidden markov model.

Prerequisites

What things you need to install the software and how to install them:

Python 3.8 or higher versions This setup requires that your machine has latest version of python. The following url https://www.python.org/downloads/ can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: https://www.pythoncentral.io/add-python-to-path-python-is-not- recognized-as-an-internal-or-external- command/. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url https://www.anaconda.com/download/ You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.8 then run below commands in command prompt/terminal to install these packages pip install -U scikit-learn pip install numpy pip install scipy if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages conda install -c scikit-learn conda install -c anaconda numpy conda install -c anaconda scipy .Install hmm learn.

Video Link

https://drive.google.com/drive/folders/1nKjHKic6ZATLwmXNwcyb1Guxb1LNOrrO

Dataset used

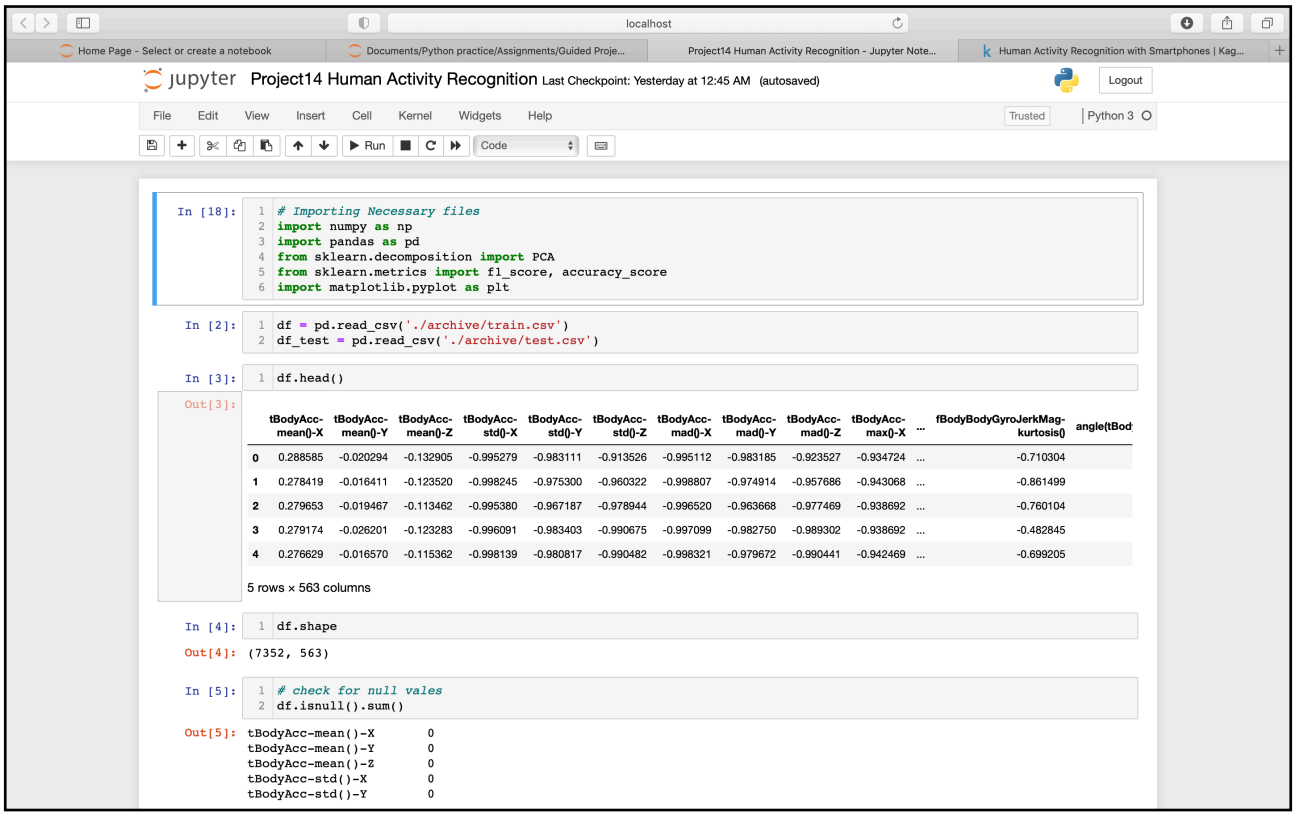
The Human Activity Recognition database was built from the recordings of 30 study participants performing activities of daily living (ADL) while carrying a waist-mounted smartphone with embedded inertial sensors. The objective is to classify activities into one of the six activities performed.Each person performed six activities (WALKING, WALKINGUPSTAIRS, WALKINGDOWNSTAIRS, SITTING, STANDING, LAYING). There are 563 columns and apron 7500 records.

Method used for detection

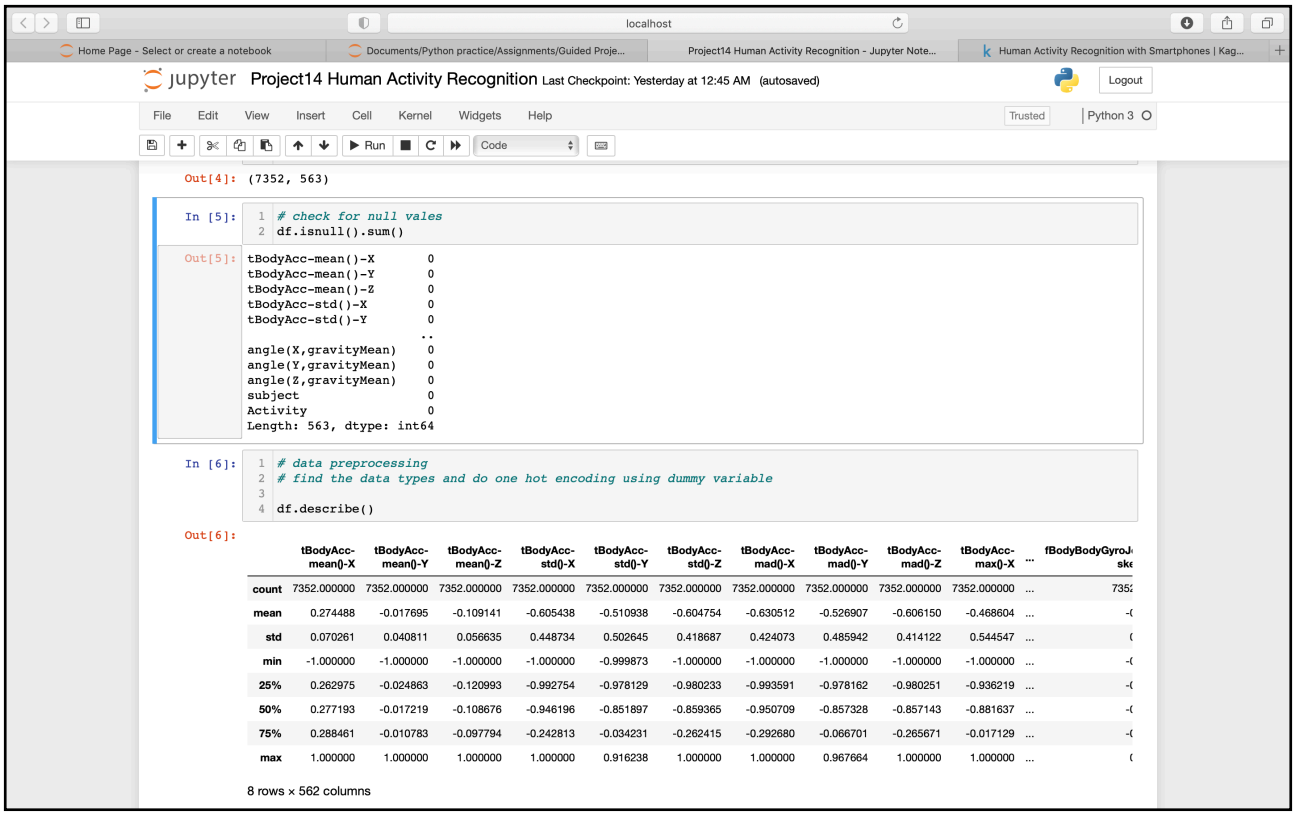
- Reading the dataset, preprocessing the dataset
- Reducing dimensionality for redundant features
- Applying hmm model

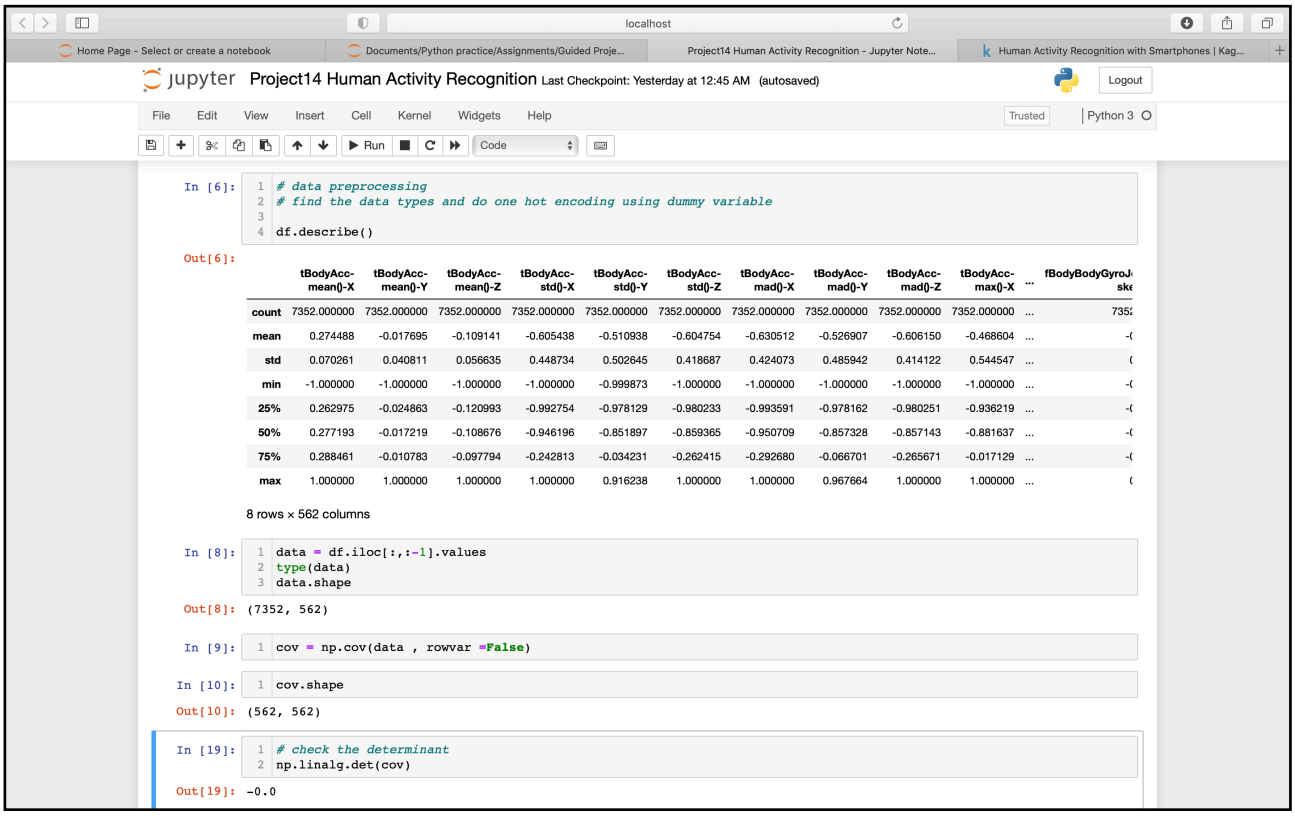
Importing the libraries and capturing images:

Importing the necessary libraries and reading the Human activity recognition with smartphone train and test data

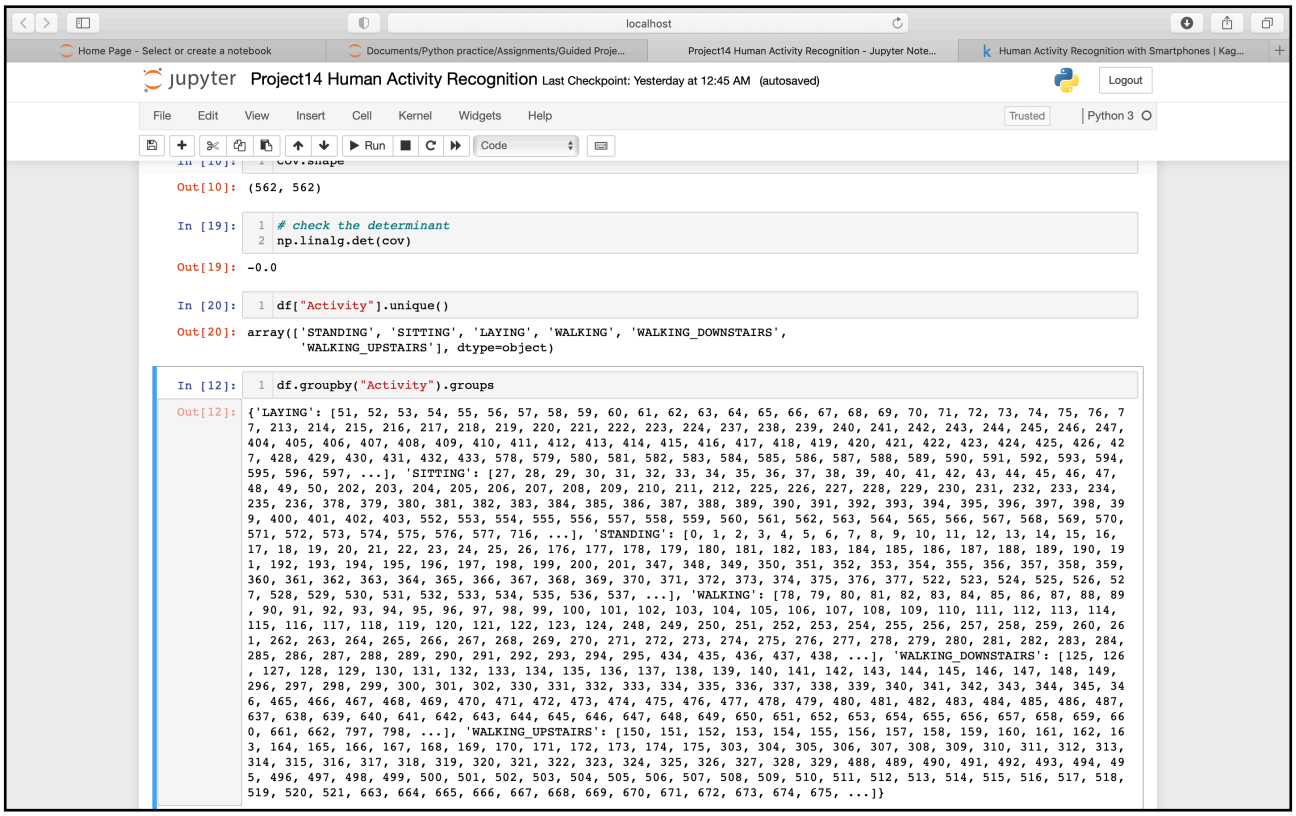


Data processing removing any null values and doing one hot encoder using dummy variable

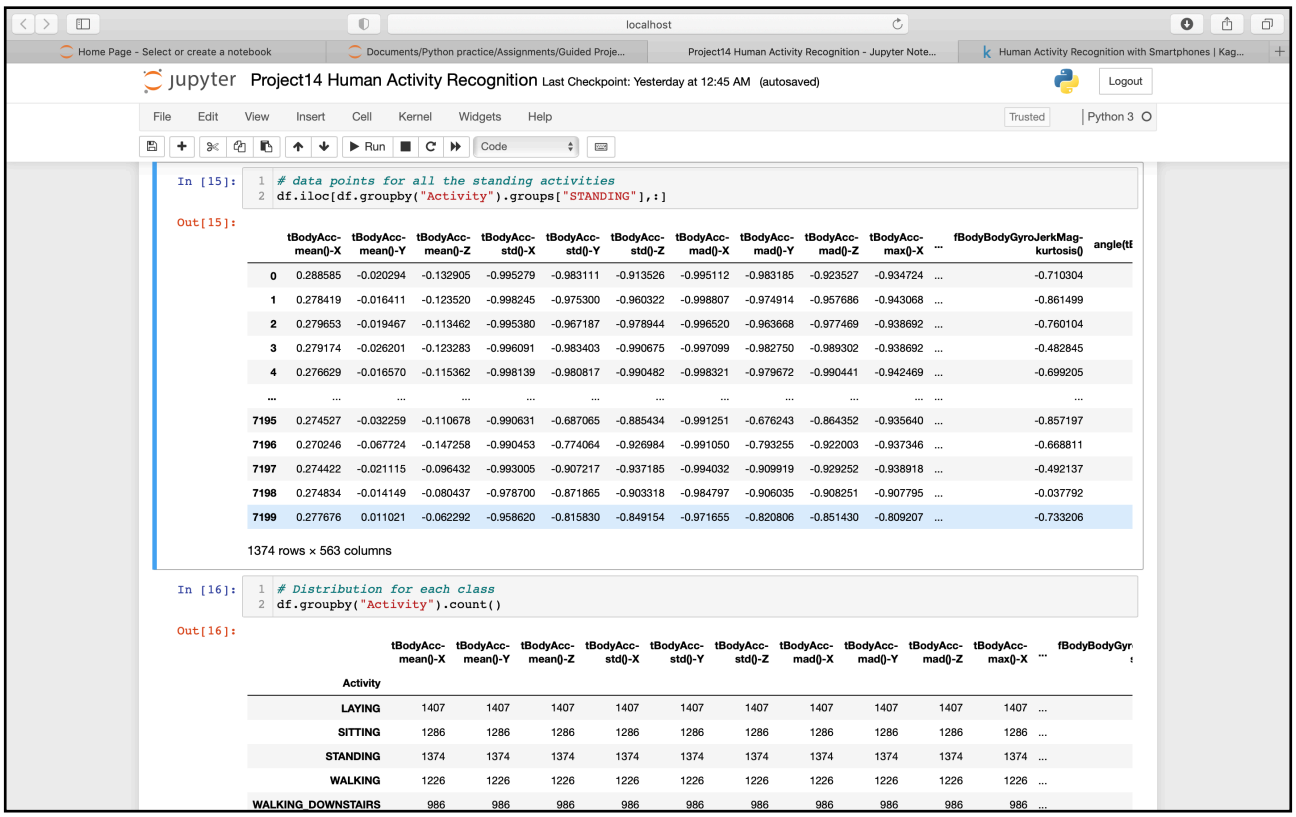




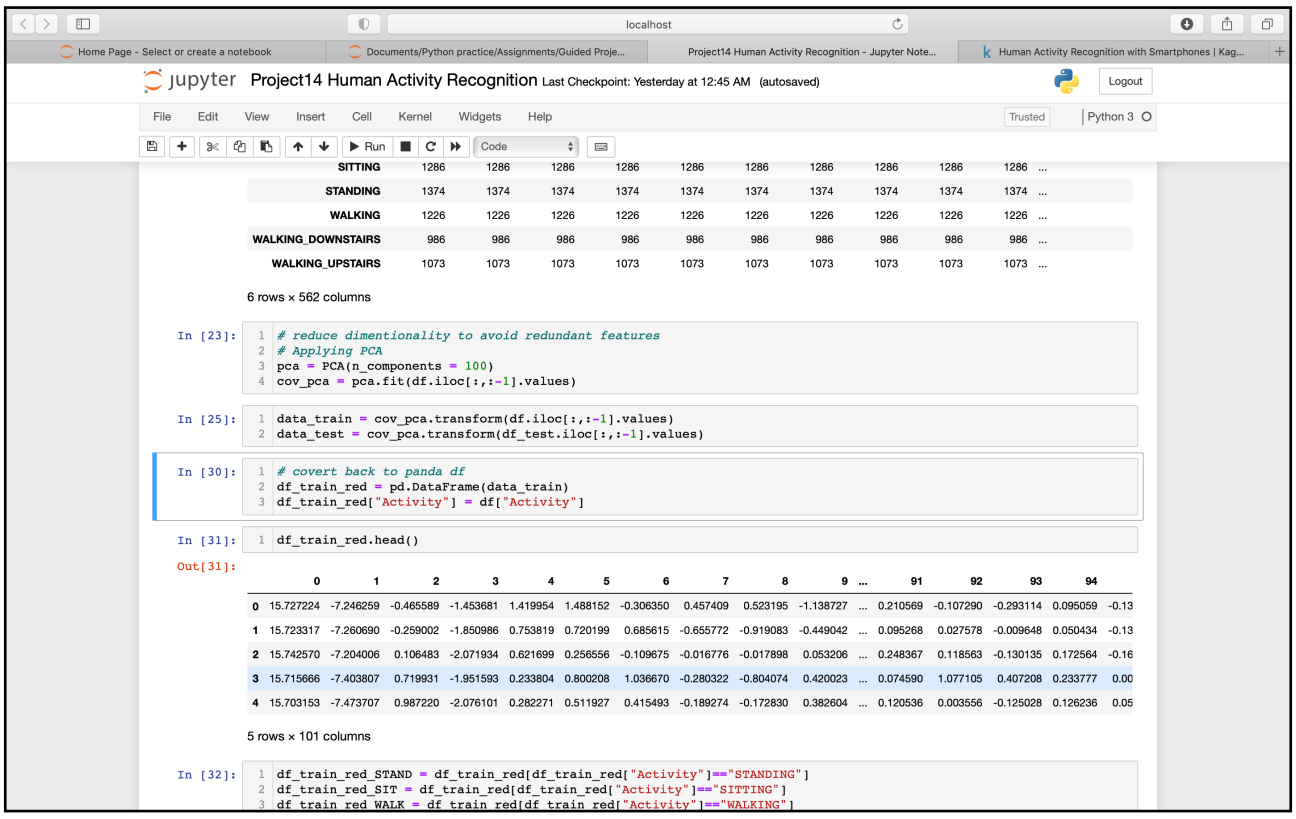
Grouping by each of six activity



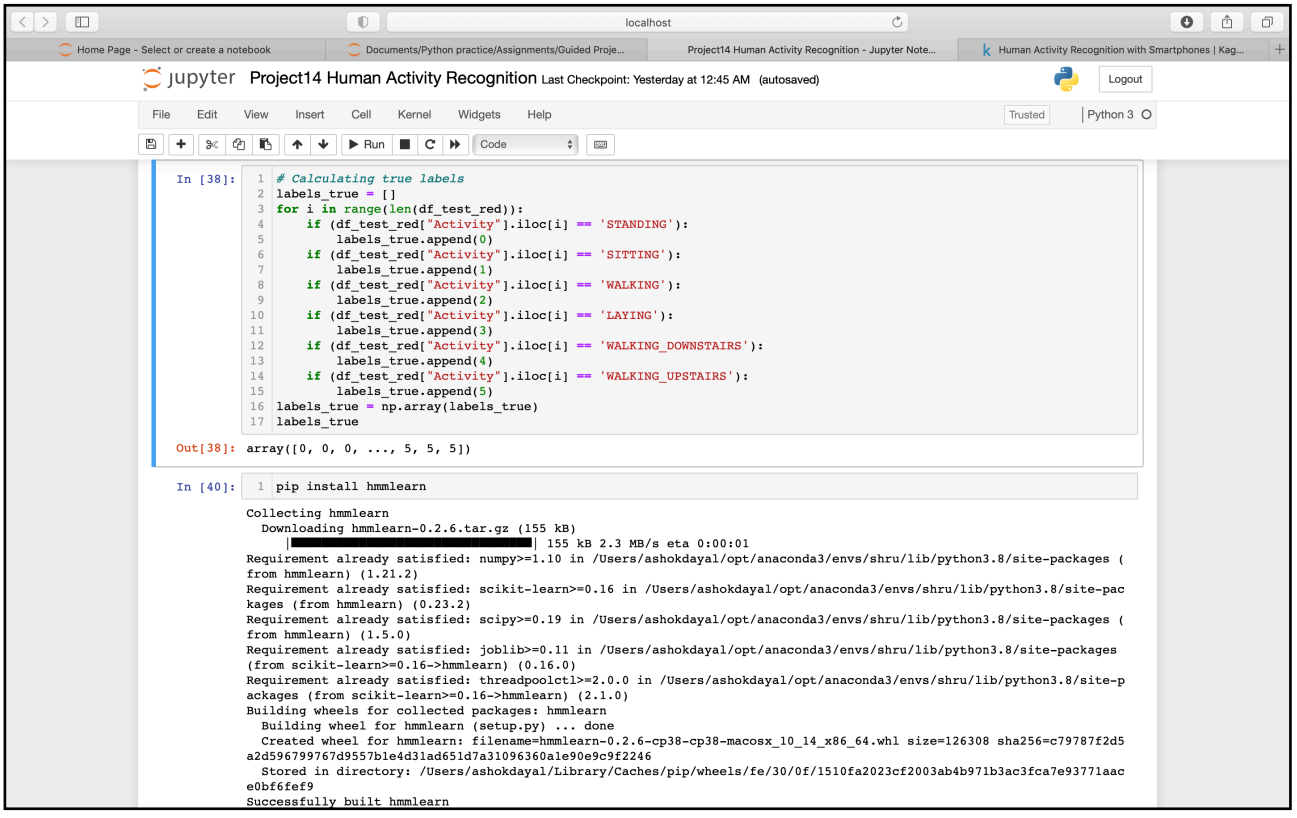
Checking the distribution for each class



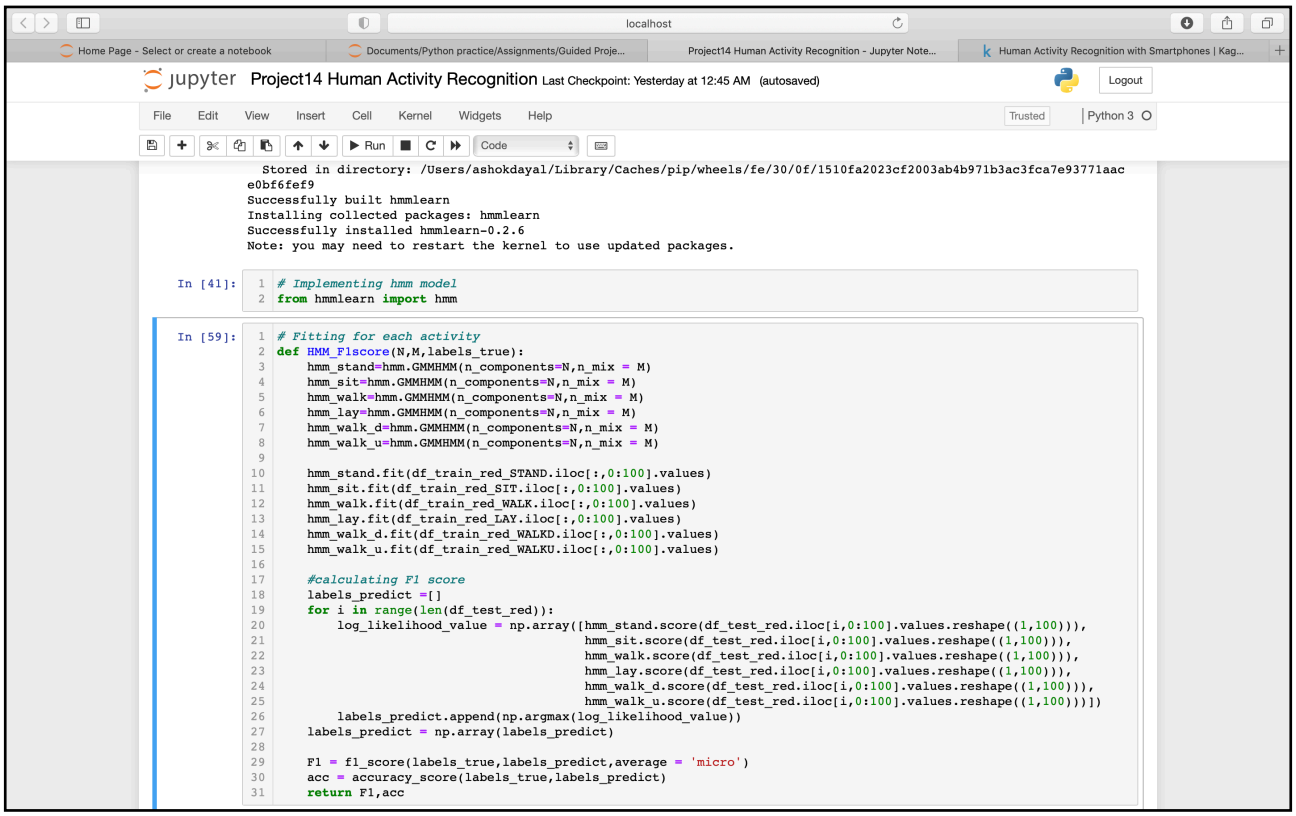
Reduce Dimensionality and covert back to data frame



Calculating true labels and Installing HMM learn



Applying HMM and fitting for each activity



```
  Stored in directory: /Users/ashokdayal/Library/Caches/pip/wheels/fe/30/0f/1510fa2023cf2003ab4b971b3ac3fca7e93771aac
e0bf6fef9
Successfully built hmmlearn
Installing collected packages: hmmlearn
Successfully installed hmmlearn-0.2.6
Note: you may need to restart the kernel to use updated packages.
```

In [41]:
```
# Implementing hmm model
from hmmlearn import hmm
```

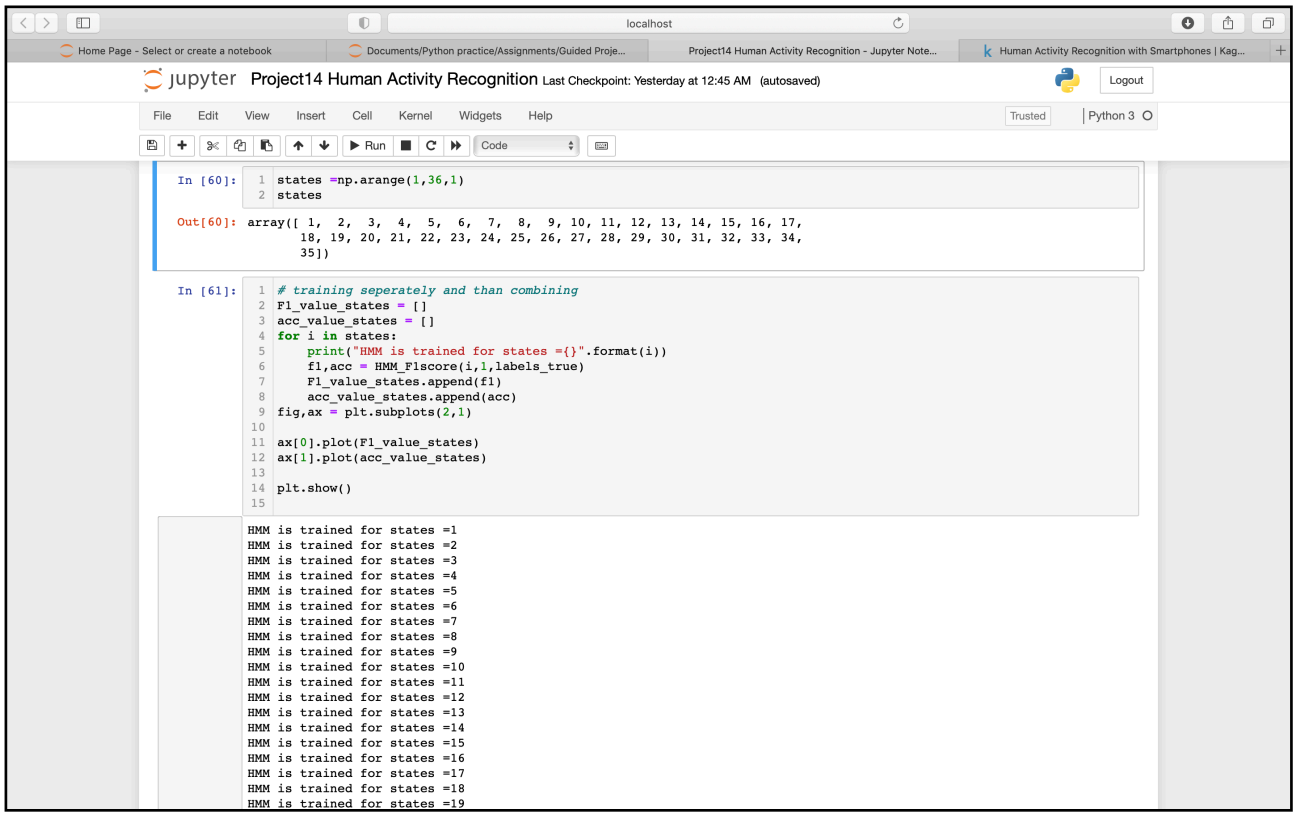
In [59]:
```
# Fitting for each activity
def HMM_F1score(N,M,labels_true):
    hmm_stand=hmm.GMMHMM(n_components=N,n_mix = M)
    hmm_sit=hmm.GMMHMM(n_components=N,n_mix = M)
    hmm_walk=hmm.GMMHMM(n_components=N,n_mix = M)
    hmm_lay=hmm.GMMHMM(n_components=N,n_mix = M)
    hmm_walk_d=hmm.GMMHMM(n_components=N,n_mix = M)
    hmm_walk_u=hmm.GMMHMM(n_components=N,n_mix = M)

    hmm_stand.fit(df_train_red_STAND.iloc[:,0:100].values)
    hmm_sit.fit(df_train_red_SIT.iloc[:,0:100].values)
    hmm_walk.fit(df_train_red_WALK.iloc[:,0:100].values)
    hmm_lay.fit(df_train_red_LAY.iloc[:,0:100].values)
    hmm_walk_d.fit(df_train_red_WALKD.iloc[:,0:100].values)
    hmm_walk_u.fit(df_train_red_WALKU.iloc[:,0:100].values)

    #calculating F1 score
    labels_predict =[]
    for i in range(len(df_test_red)):
        log_likelihood_value = np.array([hmm_stand.score(df_test_red.iloc[i,0:100].values.reshape((1,100))),
                                         hmm_sit.score(df_test_red.iloc[i,0:100].values.reshape((1,100))),
                                         hmm_walk.score(df_test_red.iloc[i,0:100].values.reshape((1,100))),
                                         hmm_lay.score(df_test_red.iloc[i,0:100].values.reshape((1,100))),
                                         hmm_walk_d.score(df_test_red.iloc[i,0:100].values.reshape((1,100))),
                                         hmm_walk_u.score(df_test_red.iloc[i,0:100].values.reshape((1,100)))])
        labels_predict.append(np.argmax(log_likelihood_value))
    labels_predict = np.array(labels_predict)

    F1 = f1_score(labels_true,labels_predict,average = 'micro')
    acc = accuracy_score(labels_true,labels_predict)
    return F1,acc
```

Training separately and combining all six activity



In [60]:
```
states =np.arange(1,36,1)
states
```

Out[60]:
```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35])
```

In [61]:
```
# training seperately and than combining
F1_value_states = []
acc_value_states = []
for i in states:
    print("HMM is trained for states ={}".format(i))
    f1,acc = HMM_F1score(i,1,labels_true)
    F1_value_states.append(f1)
    acc_value_states.append(acc)
fig,ax = plt.subplots(2,1)

ax[0].plot(F1_value_states)
ax[1].plot(acc_value_states)

plt.show()

```

```
HMM is trained for states =1
HMM is trained for states =2
HMM is trained for states =3
HMM is trained for states =4
HMM is trained for states =5
HMM is trained for states =6
HMM is trained for states =7
HMM is trained for states =8
HMM is trained for states =9
HMM is trained for states =10
HMM is trained for states =11
HMM is trained for states =12
HMM is trained for states =13
HMM is trained for states =14
HMM is trained for states =15
HMM is trained for states =16
HMM is trained for states =17
HMM is trained for states =18
HMM is trained for states =19
```

Plotting to show the f1 value and accuracy