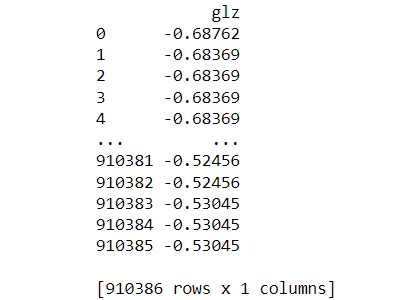
import pandas as pd

import numpy as np

from statistics import mean

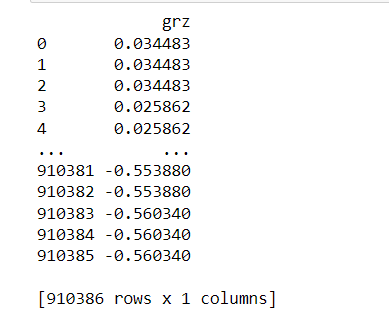
df=pd.read\_csv("Gait\_Data.csv",usecols =['glz'])

print(df)



ef=pd.read\_csv("Gait\_Data.csv",usecols =['grz'])

print(ef)



C=90.123

cfc=0

a=df.values.tolist()

z=ef.values.tolist()

b=np.array(a)

z=np.array(z)

d=df.count()

print(d)



c=[]

e=len(a)/10

for i in range(0,int(e),2):

c.append(a[i])

c.append(a[i+1])

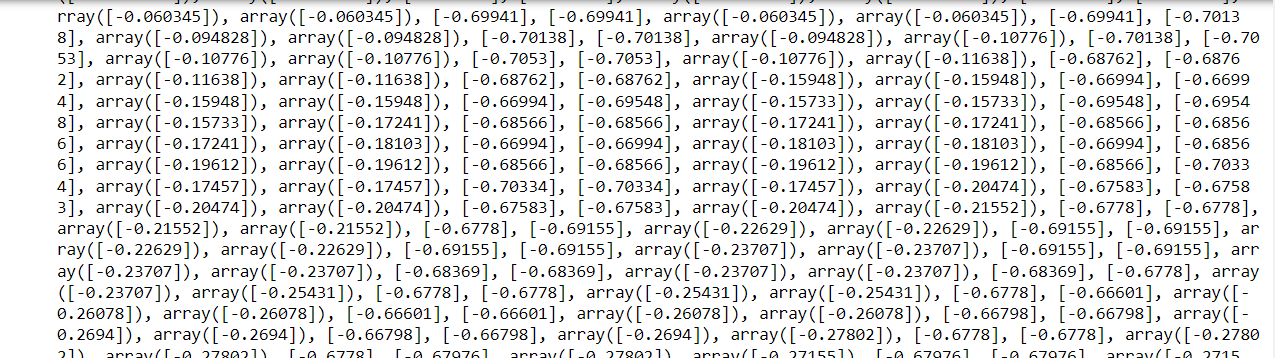
c.append(z[i])

c.append(z[i+1])

f=len(c)

print(f)

print(c)



def find\_peaks(c):

peaks = []

for i in range(1, len(c) - 1):

if c[i] < c[i-1] and c[i] < c[i+1]:

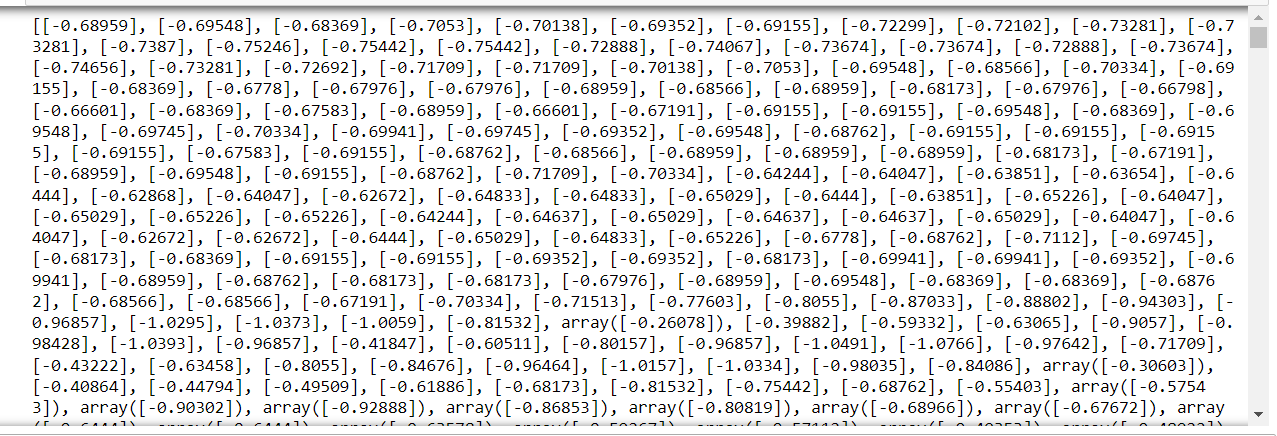
peaks.append(c[i])

return peaks

g=find\_peaks(c)

print(g)

k=len(g)

print(k) 

def find\_valley(c):

valley = []

for i in range(1, len(c) - 1):

if c[i] > c[i-1] and c[i] > c[i+1]:

valley.append(c[i])

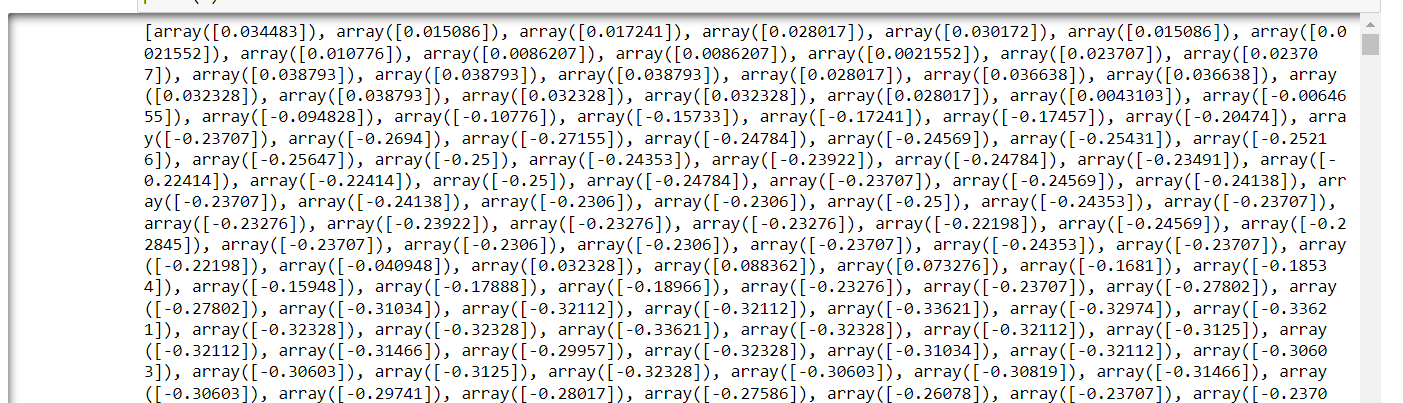
return valley

h=find\_valley(c)

print(h)

l=len(h)

print(l)



m=np.array(c)

def countlist(valley,peak):

return[sub[item] for item in range(len(h))

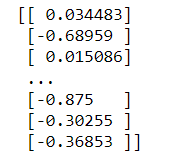
for sub in[valley,peak]]

com=(countlist(find\_valley(m),find\_peaks(m)))

type(com)

g=np.array(com)

print(g)



n=len(g)+len(h)

C=20

temp=[]

cfc=0

i=0

fc=0

lam=0

lc=0

msc=0

print(n)

alg=[]

for i in range(1,n,4):

if((C-m[i])\*(C-m[i-1])>0):

cfc=cfc+1

if(cfc==2):

fc=m[i]

cfc=0

if((C1\*m[i]<=m[i]<=C)and (fc!=0)):

lam=m[i+1]

if((m[i+1]<=C2)and (lam!=0)):

lc=m[i+2]

if((m[i+2]>0)and(lc!=0)):

msc=m[i+3]

alg.append(fc)

alg.append(lam)

alg.append(lc)

alg.append(msc)

cfc==0

else:

i=i-1

else:

i=i-2

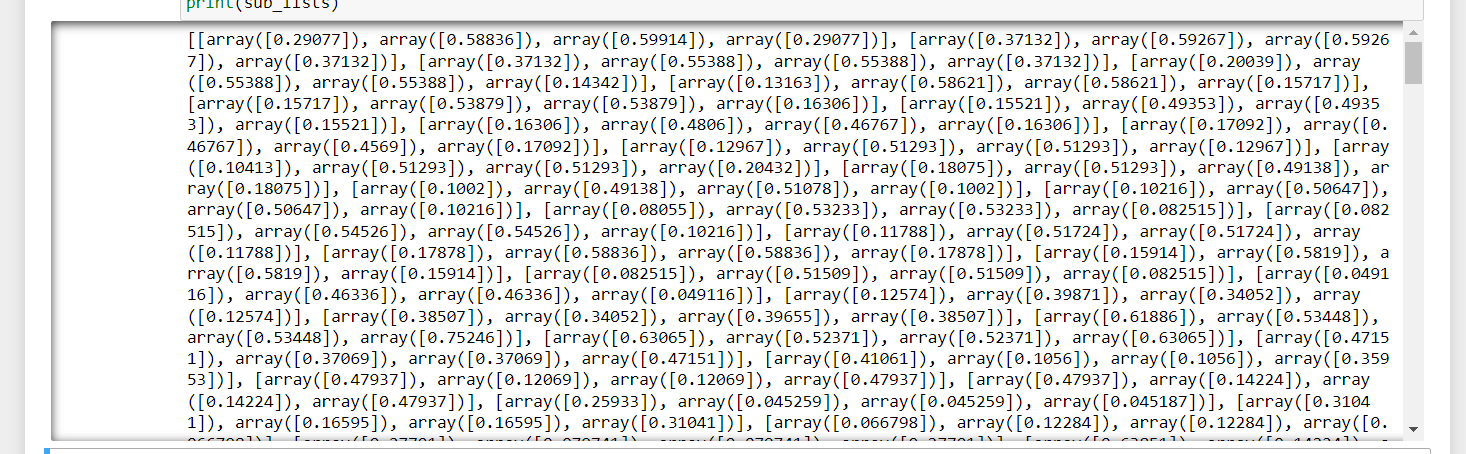


def divide\_list(lst, size):

return [lst[i:i+size] for i in range(0, len(lst), size)]

sub\_lists = divide\_list(alg, 4)

print(sub\_lists)



cc=len(sub\_lists)

print(cc)



cd=[]

lad=[]

for i in range(cc):

cd.append(sub\_lists[i][0])

lad.append(sub\_lists[i][1])

print(lad)



cm=np.average(cd)

print(cm)

lam=np.average(lad)

print(lam)



sd=np.std(cd)

print(sd)



alpha=cm+(2\*sd)

print(alpha)



C2=3

result=["ONGROUND"]

cmax=max(c)

cmin=min(c)

dif=abs(cmax-cmin)

for i in range(cc):

if(lad[i]!=0):

if(lad[i]>0):

result.append("STATE UPDATED")

result.append("WALKING UPSTAIRS")

if(lad[i]>lam):

result.append("STATE UPDATED")

result.append("WALKING DOWNSTAIRS")

if((cd[i]>alpha)and(result[i-1]!="WALKING IN TREADMILL")):

result.append("STATE UPDATED")

result.append("TRANSITION STATE")

if((cd[i]>alpha)and(result[i-1]!="WALKING UPSTAIRS")or(result[i-1]!="WALKING DOWNSTAIRS")):

result.append("STATE UPDATED")

result.append("WALKING IN TREADMILL")

if(cd[i]<=alpha):

result.append("STATE UPDATED")

result.append("ON GROUND")

if(dif<C2):

result.append("STATE UPDATED")

result.append("STATIONARY")

print(result)

