import time

import board

import busio

import digitalio

import pickle # Use pickle instead of joblib

import numpy as np

import pandas as pd

from adafruit\_adxl34x import ADXL343

#load pre-trained model (decision tree)

with open('dt\_model.pkl', 'rb') as model\_file:

model = pickle.load(model\_file)

#set up connection

i2c = busio.I2C(board.SCL, board.SDA)

accelerometer = ADXL343(i2c)

accelerometer.range = ADXL343.Range.RANGE\_4\_G

# feature extraction function

def extract\_features(data):

if len(data) > 0:

df = pd.DataFrame(data, columns=['x', 'y', 'z'])

# Calculate magnitude

df['magnitude'] = np.sqrt(df['x']\*\*2 + df['y']\*\*2 + df['z']\*\*2)

# Extract features

features = {

'mean\_x': df['x'].mean(),

'mean\_y': df['y'].mean(),

'mean\_z': df['z'].mean(),

'magnitude': df['magnitude'].mean(),

'std\_x': df['x'].std(),

'std\_y': df['y'].std(),

'std\_z': df['z'].std(),

'var\_x': df['x'].var(),

'var\_y': df['y'].var(),

'var\_z': df['z'].var(),

'range\_x': df['x'].max() - df['x'].min(),

'range\_y': df['y'].max() - df['y'].min(),

'range\_z': df['z'].max() - df['z'].min()

}

return pd.DataFrame([features])

return None

# need the code to collect data and make predictions in real time