

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
from sklearn.ensemble import AdaBoostClassifier
```

```
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive
```

Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=9473189](https://accounts.google.com/o/oauth2/auth?client_id=9473189)

Enter your authorization code:

.....

Mounted at /gdrive  
/gdrive

```
%cd /gdrive/My\ Drive/CSE512Data
!ls
```

```
/gdrive/My Drive/CSE512Data
'2020-05-16 00:00:22.1814430.csv'      '2020-05-16 00:00:58.29761911.csv'
'2020-05-16 00:00:24.9546041.csv'    '2020-05-16 00:01:00.10923012.csv'
'2020-05-16 00:00:31.6322502.csv'    all_accelerometer_data_pids_13.csv
'2020-05-16 00:00:36.5933463.csv'    clean_tac
'2020-05-16 00:00:42.6664554.csv'    good_now.csv
'2020-05-16 00:00:46.0082825.csv'    may_be_cleaned.csv
'2020-05-16 00:00:47.8559756.csv'    mega2.csv
'2020-05-16 00:00:47.8559756.gsheets' mega.csv
'2020-05-16 00:00:48.7453017.csv'    phone_types.csv
'2020-05-16 00:00:50.5036288.csv'    pids.txt
'2020-05-16 00:00:51.7498629.csv'    raw_tac
'2020-05-16 00:00:52.77848910.csv'   README.txt
```

```
frame = pd.read_csv('mega2.csv')
def threshold(value):
    def resp(x):
        if x >= value:
            return 1
        else:
            return 0
    return resp
```

```
frame['TAC_reading'] = frame['TAC_reading'].apply(threshold(0.08))
frame = frame[[x for x in frame.columns if x != 'Unnamed: 0']]
frame = frame[[x for x in frame.columns if x != 'pid']]
frame = frame[[x for x in frame.columns if x != 'window10']]
frame = frame[[x for x in frame.columns if x != 'win 10 x FFT spectral centroid spread']]
```

```

frame = frame[[x for x in frame.columns if x != 'win_10_y_FFT_spectral_centroid_spread']]
frame = frame[[x for x in frame.columns if x != 'win_10_z_FFT_spectral_centroid_spread']]
frame = frame[[x for x in frame.columns if x != 'x_FFT_spectral_centroid_spread']]
frame = frame[[x for x in frame.columns if x != 'y_FFT_spectral_centroid_spread']]
frame = frame[[x for x in frame.columns if x != 'z_FFT_spectral_centroid_spread']]
frame = frame[[x for x in frame.columns if x != 'key']]
frame = frame.dropna()
frame.shape

```

↳ (369800, 67)

```

x_values, y_values = frame[[x for x in frame.columns if x != 'TAC_reading']].to_numpy(), fram

```

```

kf = KFold(n_splits=10)

```

```

for train_index, test_index in kf.split(x_values):
    x_test, x_train = x_values[test_index], x_values[train_index]
    y_test, y_train = y_values[test_index], y_values[train_index]
    clf = AdaBoostClassifier(n_estimators=50)
    clf.fit(x_train, y_train)
    print('Accuracy Adaboost ', accuracy_score(y_test, clf.predict(x_test)))

```

↳ Accuracy Adaboost 0.6145754461871282  
 Accuracy Adaboost 0.8228501892915089  
 Accuracy Adaboost 0.7525959978366684  
 Accuracy Adaboost 0.8424824229313143  
 Accuracy Adaboost 0.9599242833964305  
 Accuracy Adaboost 0.5472417522985398  
 Accuracy Adaboost 0.5152514872904272  
 Accuracy Adaboost 0.6639264467279611  
 Accuracy Adaboost 0.7445646295294754  
 Accuracy Adaboost 0.9146295294753921

