

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
```

```
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

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.gsheet' 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
```

```
from sklearn.model_selection import KFold
```

```
import numpy as np
```

```
kf = KFold(n_splits=10)
```

```
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score
```

```
kf.get_n_splits(x_values)
```

```
↳ 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 = MLPClassifier(solver='sgd', random_state=1)
    clf.fit(x_train, y_train)
    print('Accuracy ', accuracy_score(y_test, clf.predict(x_test)))
```

```
↳
```

```

/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
Accuracy 0.5155759870200108
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
Accuracy 0.8356408869659275
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
Accuracy 0.8252028123309897
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
Accuracy 0.9373985938345052
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
Accuracy 0.8008018224224721

Accuracy 0.47798810167658196
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
Accuracy 0.5709031909140075
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
Accuracy 0.7298269334775555
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)
Accuracy 0.6399675500270416
Accuracy 0.9104380746349378
/usr/local/lib/python3.6/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:
  % self.max_iter, ConvergenceWarning)

```