```
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 sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import SGDClassifier
```

```
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive
%cd /gdrive/My\ Drive/CSE512Data
```

```
Drive already mounted at /gdrive; to attempt to forcibly remount, call drive.mount("/gdr
/gdrive
/gdrive/My Drive/CSE512Data
```

```
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 = GaussianNB()
  clf.fit(x_train, y_train)
  print('Accuracy Gaussian NB ', accuracy_score(y_test, clf.predict(x_test)))
```

```
Accuracy Gaussian NB  0.727257977285019
Accuracy Gaussian NB  0.16722552731206058
Accuracy Gaussian NB  0.17431043807463495
Accuracy Gaussian NB  0.0622769064359113
Accuracy Gaussian NB  0.0
Accuracy Gaussian NB  0.5220118983234181
Accuracy Gaussian NB  0.4290968090859924
Accuracy Gaussian NB  0.2702271498107085
Accuracy Gaussian NB  0.3600054083288264
Accuracy Gaussian NB  0.08956192536506219
```

```
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 = SGDClassifier(loss="hinge", penalty="l2")
  clf.fit(x_train, y_train)
  print('Accuracy SGDClassifier ', accuracy_score(y_test, clf.predict(x_test)))
```

```
Accuracy SGDClassifier  0.7079232017306653
Accuracy SGDClassifier  0.8343428880475933
Accuracy SGDClassifier  0.8255273120605733
Accuracy SGDClassifier  0.937506760411033
Accuracy SGDClassifier  1.0
Accuracy SGDClassifier  0.4781233098972418
Accuracy SGDClassifier  0.5705786911844241
Accuracy SGDClassifier  0.7299621416982153
Accuracy SGDClassifier  0.6399945916711736
Accuracy SGDClassifier  0.9105462412114657
```

```
frame['TAC_reading'].value_counts()
```

```
0    283207
1     86593
Name: TAC_reading, dtype: int64
```

```
response_values  = []
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 = SGDClassifier(loss="log", penalty="l2")
  clf.fit(x_train, y_train)
  response_values.append(accuracy_score(y_test, clf.predict(x_test)))
  print('Accuracy SGDClassifier Logistic regression', accuracy_score(y_test, clf.predict(x_te
print('Avarage Accuracy ', np.mean(response_values))
```

```
Accuracy SGDClassifier Logistic regression 0.7273391022174148
Accuracy SGDClassifier Logistic regression 0.8345051379123851
Accuracy SGDClassifier Logistic regression 0.8255273120605733
Accuracy SGDClassifier Logistic regression 0.937506760411033
Accuracy SGDClassifier Logistic regression 0.9999729583558681
Accuracy SGDClassifier Logistic regression 0.4781233098972418
Accuracy SGDClassifier Logistic regression 0.571254732287723
Accuracy SGDClassifier Logistic regression 0.7298269334775555
Accuracy SGDClassifier Logistic regression 0.6399675500270416
Accuracy SGDClassifier Logistic regression 0.9105732828555976
Average Accuracy: 0.7654597079502434
```