

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import sklearn
4 from sklearn.model_selection import train_test_split
5 from sklearn import linear_model
6 from sklearn.utils import shuffle
7 from sklearn import preprocessing
8 import numpy as np
9
10 # PRE-PROCESSING OF DATA
11 file = pd.read_table('pinch and fist 8 december.txt')
12 feature_names = ['Sensor1', 'Sensor2', 'Sensor3']
13 X = file[feature_names]
14 y = file['grasp']
15
16 # remove nan error
17 import pandas as pd
18 X=X.replace([np.inf,-np.inf])
19 X = X.apply (pd.to_numeric, errors='coerce')
20 X = X.dropna()
21
22 y=y.replace([np.inf,-np.inf])
23 y = y.apply (pd.to_numeric, errors='coerce')
24 y = y.dropna()
25
26
27 # TESTING AND TRAINING
28 X_train, X_test, y_train, y_test = train_test_split(X
    , y, random_state=5)
29 from sklearn.preprocessing import MinMaxScaler
30 scaler = MinMaxScaler()
31 X_train = scaler.fit_transform(X_train)
32 X_test = scaler.transform(X_test)
33
34 # LOGISTIC REGRESSION
35 from sklearn.linear_model import LogisticRegression
36 logreg = LogisticRegression()
37 logreg.fit(X_train, y_train)
38 print('Accuracy of Logistic regression classifier on
    training set: {:.2f}'
39       .format(logreg.score(X_train, y_train)))
40 print('Accuracy of Logistic regression classifier on
    test set: {:.2f}'
41       .format(logreg.score(X_test, y_test)))
```

```
42
43 # DECISION TREE
44 from sklearn.tree import DecisionTreeClassifier
45 clf = DecisionTreeClassifier().fit(X_train, y_train)
46 print('Accuracy of Decision Tree classifier on
      training set: {:.2f}'
47       .format(clf.score(X_train, y_train)))
48 print('Accuracy of Decision Tree classifier on test
      set: {:.2f}'
49       .format(clf.score(X_test, y_test)))
50
51 # KN NEIGHBORS
52 from sklearn.neighbors import KNeighborsClassifier
53 knn = KNeighborsClassifier()
54 knn.fit(X_train, y_train)
55 print('Accuracy of K-NN classifier on training set
      : {:.2f}'
56       .format(knn.score(X_train, y_train)))
57 print('Accuracy of K-NN classifier on test set: {:.2f
      }'
58       .format(knn.score(X_test, y_test)))
59
60 #LINEAR DISCRIMINATION ANALYSIS
61 from sklearn.discriminant_analysis import
      LinearDiscriminantAnalysis
62 lda = LinearDiscriminantAnalysis()
63 lda.fit(X_train, y_train)
64 print('Accuracy of LDA classifier on training set
      : {:.2f}'
65       .format(lda.score(X_train, y_train)))
66 print('Accuracy of LDA classifier on test set: {:.2f
      }'
67       .format(lda.score(X_test, y_test)))
68
69
70 # GAUSSIAN NB
71 from sklearn.naive_bayes import GaussianNB
72 gnb = GaussianNB()
73 gnb.fit(X_train, y_train)
74 print('Accuracy of GNB classifier on training set
      : {:.2f}'
75       .format(gnb.score(X_train, y_train)))
76 print('Accuracy of GNB classifier on test set: {:.2f
      }'
```

```
77         .format(gnb.score(X_test, y_test)))  
78  
79
```