

Adaboost for disease prediction

```
In [4]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
import warnings
warnings.filterwarnings('ignore')
```

```
In [5]: df = pd.read_csv('D:\Downloads\Testing.csv')
```

In [6]: df

Out[6]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	0
2	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	1	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0
15	0	0	0	0	0	1	0
16	1	1	0	0	0	0	0
17	0	1	0	0	0	1	1
18	0	0	0	0	0	1	0
19	0	0	0	0	0	0	1
20	1	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	1
23	0	0	0	0	0	0	1
24	0	0	0	0	0	0	0
25	0	0	0	0	0	1	0
26	0	0	0	1	0	1	0
27	0	0	0	0	0	1	0
28	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain
32	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0
34	0	0	0	0	0	0	1
35	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0
37	0	1	0	0	0	0	0
38	0	0	0	0	0	0	0
39	0	1	0	0	0	0	1
40	0	1	0	0	0	0	0
41	1	1	0	0	0	0	0

42 rows × 133 columns

In [7]: `df.isnull().sum()`

```
Out[7]: itching          0
skin_rash              0
nodal_skin_eruptions  0
continuous_sneezing   0
shivering              0
..
inflammatory_nails    0
blister               0
red_sore_around_nose  0
yellow_crust_ooze     0
prognosis             0
Length: 133, dtype: int64
```

```
In [8]: df.drop(["prognosis"],axis=1,inplace=True)
df
```

Out[8]:

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain
0	1	1	1	0	0	0	0
1	0	0	0	1	1	1	0
2	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0
4	1	1	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0

```
In [9]: X = df.iloc[:, :-1].values
Y = df.iloc[:, -1].values
```

```
In [10]: from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.45, random_state=10)
```

```
In [11]: from sklearn.ensemble import AdaBoostClassifier
ada = AdaBoostClassifier(n_estimators=25, learning_rate=1.5, random_state=10)
ada.fit(X_train, Y_train)
```

Out[11]: AdaBoostClassifier(learning_rate=1.5, n_estimators=25, random_state=10)

```
In [12]: Y_pred = ada.predict(X_test)
print(np.concatenate((Y_pred.reshape(len(Y_pred),1), Y_test.reshape(len(Y_test),1)
```

```
[[0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]
 [0 0]]
```

```
In [13]: print("Confusion matrix is:")
print(confusion_matrix(Y_test,Y_pred))
print("\nAccuracy score is:")
print(accuracy_score(Y_test,Y_pred))
print("\nClassification report is: ")
print(classification_report(Y_test,Y_pred))
print("\nPrecision score is:")
print(precision_score(Y_test,Y_pred))
print("\nRecall score is:")
print(recall_score(Y_test,Y_pred))
print("\nF1 score is:")
print(f1_score(Y_test,Y_pred))
```

Confusion matrix is:

[[19]]

Accuracy score is:

1.0

Classification report is:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19
accuracy			1.00	19
macro avg	1.00	1.00	1.00	19
weighted avg	1.00	1.00	1.00	19

Precision score is:

0.0

Recall score is:

0.0

F1 score is:

0.0