

# Decision tree for disease prediction

```
In [132]: import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
import warnings
warnings.filterwarnings('ignore')
```

```
In [133]: df = pd.read_csv(r'D:\Downloads\Testing.csv', delimiter = ",")
```

```
In [134]: df.isnull().sum()
```

```
Out[134]: 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 [135]: df.head(10)
```

```
Out[135]:
```

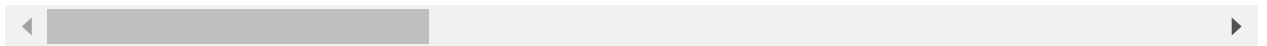
	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	...
0	1	1	1	0	0	0	0	
1	0	0	0	1	1	1	0	
2	0	0	0	0	0	0	0	
3	1	0	0	0	0	0	0	
4	1	1	0	0	0	0	0	
5	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	

```
In [136]: df.drop(["prognosis"],axis=1,inplace=True)
df.head(10)
```

```
Out[136]:
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	ston
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 rows × 132 columns



```
In [137]: data = df.to_numpy()
```

```
In [138]: clf = DecisionTreeClassifier()
```

```
In [139]: X = data[:, :-1]
```

```
Y = data[:, -1]
X
```

```
Out[139]: array([[1, 1, 1, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                [0, 0, 0, ..., 0, 0, 0],
                ...,
                [0, 1, 0, ..., 1, 0, 0],
                [0, 1, 0, ..., 0, 1, 1],
                [1, 1, 0, ..., 0, 0, 1]], dtype=int64)
```

```
In [140]: from sklearn import model_selection
X_train,X_test,Y_train,Y_test = model_selection.train_test_split(X,Y,test_size =
```

```
In [141]: clf.fit(X_train,Y_train)
```

```
Out[141]: DecisionTreeClassifier()
```

```
In [142]: Y_pred = clf.predict(X_test)
```

```
In [143]: 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:  
[[11]]

Accuracy score is:  
1.0

Classification report is:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	11
accuracy			1.00	11
macro avg	1.00	1.00	1.00	11
weighted avg	1.00	1.00	1.00	11

Precision score is:  
0.0

Recall score is:  
0.0

F1 score is:  
0.0