

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
data = pd.read_csv('/home/dslab/Desktop/dataaset/covid.csv')
```

In [3]:

```
data
```

Out[3]:

	no	pc	wbc	mc	ast	bc	ldh	diagnosis
0	1	Low	Low	Low	High	Normal	Normal	True
1	2	Low	Low	Normal	High	Normal	High	True
2	3	Low	High	Normal	High	Normal	Normal	False
3	4	Low	High	Normal	High	High	Normal	True
4	5	Low	Normal	High	High	Normal	Normal	False
5	6	Low	Normal	Normal	High	Normal	High	True
6	7	Normal	Low	Low	High	Normal	Normal	True
7	8	Normal	High	Normal	High	Normal	Normal	False
8	9	Normal	High	Normal	High	High	High	True
9	10	Normal	Normal	High	High	Normal	Normal	False
10	11	Normal	Normal	High	High	Normal	High	True
11	12	High	Low	Low	Normal	Normal	Normal	True
12	13	High	Normal	High	Normal	Normal	Normal	False
13	14	High	Normal	High	Normal	High	High	True
14	15	High	High	Normal	Normal	Normal	High	True
15	16	Low	Normal	High	High	High	Normal	False
16	17	Normal	Normal	High	High	High	Normal	False
17	18	High	Low	Low	Normal	Normal	High	True
18	19	Normal	Normal	Normal	High	Normal	Normal	False
19	20	Normal	High	Normal	High	Normal	High	True
20	21	Normal	Low	Normal	High	Normal	High	True
21	22	Low	High	Normal	High	High	High	True
22	23	Low	Low	Low	High	High	High	True
23	24	High	High	Normal	Normal	Normal	Normal	True
24	25	High	Normal	Normal	Normal	Normal	Normal	False

In [4]:

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
```

In [9]:

```
pc_encoded=le.fit_transform(data['pc'].values)
wbc_encoded=le.fit_transform(data['wbc'].values)
mc_encoded=le.fit_transform(data['mc'].values)
ast_encoded=le.fit_transform(data['ast'].values)
bc_encoded=le.fit_transform(data['bc'].values)
ldh_encoded=le.fit_transform(data['ldh'].values)
Y=le.fit_transform(data['diagnosis'].values)
X=np.array(list(zip(pc_encoded,wbc_encoded,mc_encoded,ast_encoded,bc_encoded,ldh_
```

In [10]:

X

Out[10]:

```
array([[1, 1, 1, 0, 1, 1],
       [1, 1, 2, 0, 1, 0],
       [1, 0, 2, 0, 1, 1],
       [1, 0, 2, 0, 0, 1],
       [1, 2, 0, 0, 1, 1],
       [1, 2, 2, 0, 1, 0],
       [2, 1, 1, 0, 1, 1],
       [2, 0, 2, 0, 1, 1],
       [2, 0, 2, 0, 0, 0],
       [2, 2, 0, 0, 1, 1],
       [2, 2, 0, 0, 1, 0],
       [0, 1, 1, 1, 1, 1],
       [0, 2, 0, 1, 1, 1],
       [0, 2, 0, 1, 0, 0],
       [0, 0, 2, 1, 1, 0],
       [1, 2, 0, 0, 0, 1],
       [2, 2, 0, 0, 0, 1],
       [0, 1, 1, 1, 1, 0],
       [2, 2, 2, 0, 1, 1],
       [2, 0, 2, 0, 1, 0],
       [2, 1, 2, 0, 1, 0],
       [1, 0, 2, 0, 0, 0],
       [1, 1, 1, 0, 0, 0],
       [0, 0, 2, 1, 1, 1],
       [0, 2, 2, 1, 1, 1]])
```

In [11]:

Y

Out[11]:

```
array([1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1,
       1, 1,
       1, 1, 0])
```

In [12]:

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
model = MultinomialNB()
```

In [13]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(X,Y)
```

In [14]:

```
model.fit(X_train, Y_train)
y_pred = model.predict(X_test)
```

In [15]:

```
print("Accuracy:",accuracy_score(Y_test, y_pred))
```

Accuracy: 0.7142857142857143

In [16]:

```
print("\nReport")
print(classification_report(Y_test,y_pred))
```

Report

	precision	recall	f1-score	support
0	0.33	1.00	0.50	1
1	1.00	0.67	0.80	6
accuracy			0.71	7
macro avg	0.67	0.83	0.65	7
weighted avg	0.90	0.71	0.76	7

In []: