```
df = pd.read_csv('/content/drug200.csv')
df.head()
```

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

```
df.isnull().sum()
```

```
Age            0
Sex            0
BP             0
Cholesterol    0
Na_to_K        0
Drug           0
dtype: int64
```

```
df['Drug'].unique()
```

```
array(['DrugY', 'drugC', 'drugX', 'drugA', 'drugB'], dtype=object)
```

```
df['Drug'].value_counts()
```

```
DrugY    91
drugX    54
drugA    23
drugC    16
drugB    16
Name: Drug, dtype: int64
```

```
x = df.iloc[:,1:5].values
x
```

```
['M', 'NORMAL', 'NORMAL', 15.79],
['M', 'NORMAL', 'HIGH', 12.26],
['F', 'NORMAL', 'NORMAL', 12.295],
['F', 'NORMAL', 'NORMAL', 8.107],
['F', 'HIGH', 'HIGH', 13.091],
['M', 'LOW', 'HIGH', 10.291],
['M', 'NORMAL', 'HIGH', 31.686],
['F', 'LOW', 'HIGH', 19.796],
['F', 'HIGH', 'HIGH', 19.416],
['M', 'NORMAL', 'NORMAL', 10.898],
['M', 'LOW', 'NORMAL', 27.183],
['F', 'HIGH', 'NORMAL', 18.457],
['F', 'HIGH', 'NORMAL', 10.189],
['F', 'LOW', 'HIGH', 14.16],
['M', 'HIGH', 'NORMAL', 11.34],
['M', 'HIGH', 'HIGH', 27.826],
['M', 'NORMAL', 'NORMAL', 10.091],
['M', 'HIGH', 'HIGH', 18.703],
['F', 'LOW', 'NORMAL', 29.875],
['M', 'HIGH', 'NORMAL', 9.475],
['M', 'LOW', 'NORMAL', 20.693],
['M', 'LOW', 'NORMAL', 8.37],
['F', 'HIGH', 'HIGH', 13.303],
['F', 'NORMAL', 'NORMAL', 27.05],
['M', 'HIGH', 'HIGH', 12.856],
['M', 'NORMAL', 'NORMAL', 10.832],
['M', 'NORMAL', 'HIGH', 24.658],
['F', 'HIGH', 'NORMAL', 24.276],
['M', 'HIGH', 'HIGH', 13.967],
['F', 'NORMAL', 'HIGH', 19.675],
['F', 'NORMAL', 'HIGH', 10.605],
['F', 'NORMAL', 'NORMAL', 22.905],
['M', 'HIGH', 'NORMAL', 17.069],
['M', 'LOW', 'NORMAL', 20.909],
['F', 'HIGH', 'HIGH', 11.198],
['F', 'HIGH', 'NORMAL', 19.161],
['F', 'HIGH', 'HIGH', 13.313],
['F', 'LOW', 'NORMAL', 10.84],
['M', 'HIGH', 'HIGH', 13.934],
['M', 'NORMAL', 'HIGH', 7.761],
['F', 'LOW', 'HIGH', 9.712],
['F', 'HIGH', 'NORMAL', 11.326],
['F', 'LOW', 'HIGH', 10.067],
['M', 'HIGH', 'HIGH', 13.935],
['F', 'NORMAL', 'HIGH', 13.597],
['M', 'LOW', 'HIGH', 15.478],
['F', 'HIGH', 'NORMAL', 23.091],
['F', 'NORMAL', 'NORMAL', 17.211],
['M', 'NORMAL', 'HIGH', 16.594],
['M', 'HIGH', 'NORMAL', 15.156],
['F', 'HIGH', 'HIGH', 29.45],
['F', 'LOW', 'NORMAL', 29.271],
```

```
y = pd.get_dummies(df.iloc[:,5:]).values
y
```

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

```
            [1, 0, 0, 0, 0],
            [1, 0, 0, 0, 0],
            [0, 0, 1, 0, 0],
            [0, 0, 0, 1, 0],
            [0, 0, 1, 0, 0],
            [1, 0, 0, 0, 0],
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,random_state=21)
xtrain.shape, xtest.shape, ytrain.shape, ytest.shape
```

```
((160, 4), (40, 4), (160, 5), (40, 5))
```

```
# ANN Model

model = Sequential()
model.add(Dense(8,input_dim=4,activation='relu'))
model.add(Dense(26,activation='relu'))
model.add(Dense(3,activation='softmax'))

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

model.summary()
```

```
Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_39 (Dense)            (None, 8)                 40

 dense_40 (Dense)            (None, 26)                234

 dense_41 (Dense)            (None, 3)                 81

=================================================================
Total params: 355
Trainable params: 355
Non-trainable params: 0
_________________________________________________________________
```

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,random_state=21)
xtrain.shape, xtest.shape, ytrain.shape, ytest.shape

X_new = np.random.rand(5,4)

y_pred = model.predict(X_new)

print(y_pred)
```

```
1/1 [==============================] - 31s 31s/step
[[0.41158223 0.27753615 0.3108816 ]
 [0.4466757  0.25308448 0.30023983]
 [0.43947807 0.2730741  0.2874478 ]
 [0.40440667 0.26785207 0.3277413 ]
 [0.38155225 0.301505   0.31694275]]
```