

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder, MinMaxScaler
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report
```

```
irr=pd.read_csv('/content/data.csv')
irr.head()
```

	crop	moisture	temp	pump
0	cotton	638	16	1
1	cotton	522	18	1
2	cotton	741	22	1
3	cotton	798	32	1
4	cotton	690	28	1

```
lb=LabelEncoder()
irr['crop']=lb.fit_transform(irr['crop'])
```

```
# this step is not necessary because all crops are same here
```

```
irr.tail()
```

	crop	moisture	temp	pump
195	0	941	13	1
196	0	902	45	1
197	0	894	42	1
198	0	1022	45	1
199	0	979	10	1

```
irr.isna().sum()
```

```
crop      0
moisture  0
temp      0
pump      0
dtype: int64
```

```
# checking balanced or not
irr['pump'].value_counts()
```

✓ 0s completed at 8:59 PM



```
1    150
0     50
Name: pump, dtype: int64
```

```
x=irr.iloc[:, :-1].values
y=irr.iloc[:, -1].values
```

```
xtr,xts,ytr,yts=train_test_split(x,y,test_size=.30,random_state=42)
```

```
# standardscaler gives negative values multinomial doesnt take it
std=MinMaxScaler()
std.fit(xtr)
xtr=std.transform(xtr)
xts=std.transform(xts)
```

```
model=MultinomialNB()
model.fit(xtr,ytr)
ypr=model.predict(xts)
```

```
res=confusion_matrix(yts,ypr)
res
```

```
array([[ 0, 15],
       [ 0, 45]])
```

```
print(classification_report(yts,ypr))
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	15
1	0.75	1.00	0.86	45
accuracy			0.75	60
macro avg	0.38	0.50	0.43	60
weighted avg	0.56	0.75	0.64	60

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:
_warn_prf(average, modifier, msg_start, len(result))
```

```
from sklearn.metrics import precision_recall_fscore_support
```

```
# Specify the zero_division parameter
precision_recall_fscore_support(yts,ypr, zero_division=1)
```

```
(array([1. , 0.75]),  
 array([0., 1.]),  
 array([0. , 0.85714286]),  
 array([15, 45]))
```

```
score=accuracy_score(yts,ypr)  
score
```

```
0.75
```

[Colab paid products](#) - [Cancel contracts here](#)