```python
In [13]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import sklearn.datasets as ds
         import seaborn as sns
```

Regression

```python
In [14]: data = ds.load_diabetes()
         data.keys()
```

```
Out[14]: dict_keys(['data', 'target', 'frame', 'DESCR', 'feature_names', 'data_filename',
         'target_filename', 'data_module'])
```

```python
In [15]: X = pd.DataFrame(data['data'],columns=data['feature_names'])
```

```python
In [16]: y = data['target']
```

```python
In [17]: from sklearn.model_selection import train_test_split as tts
         X_train , X_test , y_train , y_test = tts(X , y , test_size = 0.2 )
```

```python
In [18]: len(X_train) , len(X_test) , len(y_train) , len(y_test)
```

```
Out[18]: (353, 89, 353, 89)
```

```python
In [19]: data = ds.load_iris()
         data.keys()
```

```
Out[19]: dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'f
         ilename', 'data_module'])
```

```python
In [20]: X = pd.DataFrame(data['data'] ,columns = data['feature_names'])
```

```python
In [21]: y = data['target']
```

```python
In [22]: X_train , X_test , y_train , y_test = tts(X , y , test_size = 0.2 )
         len(X_train) , len(X_test) , len(y_train) , len(y_test)
```
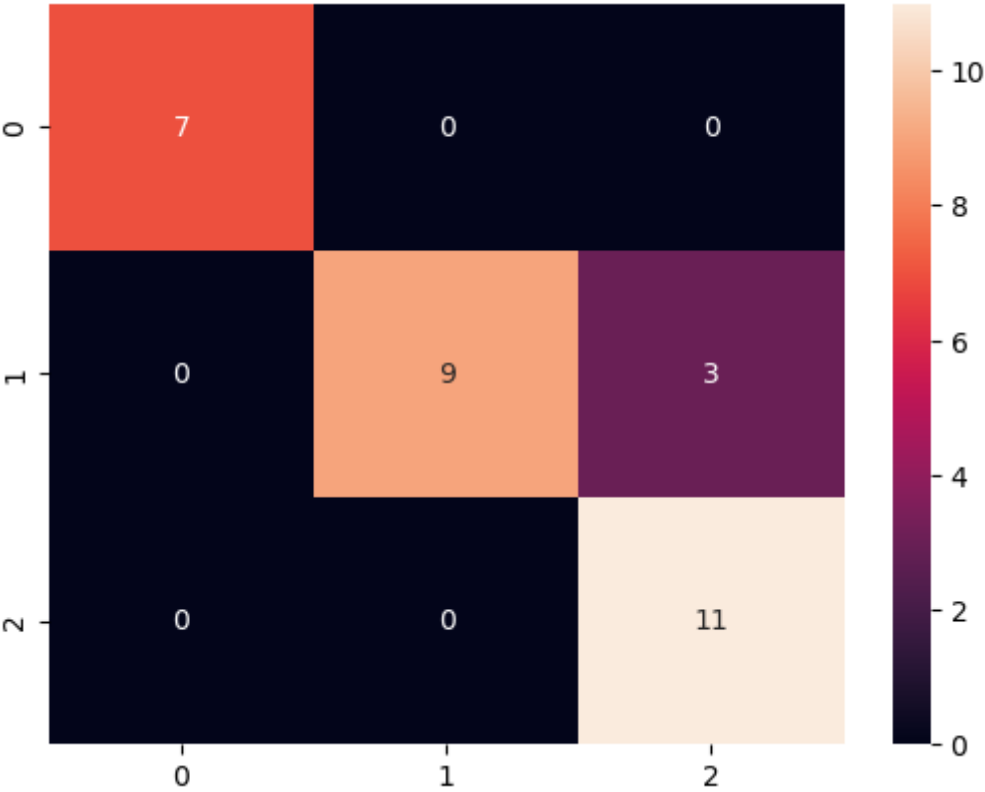
```
Out[22]: (120, 30, 120, 30)
```

```python
In [23]: from sklearn.metrics import classification_report as cr
         from sklearn.metrics import confusion_matrix as cm
         def plot_metrics(y_pred , y_test):
           print(cr(y_test , y_pred))
           sns.heatmap(cm(y_test,y_pred),annot =True)
```

Navie Bayes

```python
In [24]: from sklearn.naive_bayes import MultinomialNB as LR
         lr = LR().fit(X_train , y_train)
         y_pred = lr.predict(X_test)
         plot_metrics(y_pred , y_test)
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	7
1	1.00	0.75	0.86	12
2	0.79	1.00	0.88	11
accuracy			0.90	30
macro avg	0.93	0.92	0.91	30
weighted avg	0.92	0.90	0.90	30

In [ ]: