In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import sklearn.datasets as ds
import seaborn as sns
```

Regression

In [2]:
```
data = ds.load_diabetes()
data.keys()
```

Out[2]:
```
dict_keys(['data', 'target', 'frame', 'DESCR', 'feature_names', 'data_filename',
'target_filename', 'data_module'])
```

In [3]:
```
X = pd.DataFrame(data['data'],columns=data['feature_names'])
```

In [4]:
```
y = data['target']
```

In [5]:
```
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 )
```

In [6]:
```
len(X_train) , len(X_test) , len(y_train) , len(y_test)
```

Out[6]:
```
(353, 89, 353, 89)
```

In [7]:
```
data = ds.load_iris()
data.keys()
```

Out[7]:
```
dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'f
ilename', 'data_module'])
```

In [8]:
```
X = pd.DataFrame(data['data'] ,columns = data['feature_names'])
```

In [9]:
```
y = data['target']
```

In [10]:
```
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[10]:
```
(120, 30, 120, 30)
```

In [11]:
```
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)
```

Logistic Regression

In [12]:
```
from sklearn.linear_model import LogisticRegression as LR
from sklearn.metrics import mean_squared_error as mse
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	8
1	0.92	1.00	0.96	11
2	1.00	0.91	0.95	11
accuracy			0.97	30
macro avg	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	30