

```
In [1]: import numpy as np
import pandas as pd
```

```
In [2]: df=pd.read_csv(r"C:\Users\pc\Downloads\diabetes.csv")
```

```
In [3]: df
```

```
Out[3]:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows x 9 columns

```
In [4]: df.columns
```

```
Out[4]: Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
               'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
              dtype='object')
```

```
'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],  
dtype='object')
```

```
In [5]: I=df.drop(['Outcome'],axis=1)
```

```
In [6]: I
```

Out[6]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	6	148	72	35	0	33.6	0.627	50
1	1	85	66	29	0	26.6	0.351	31
2	8	183	64	0	0	23.3	0.672	32
3	1	89	66	23	94	28.1	0.167	21
4	0	137	40	35	168	43.1	2.288	33
...
763	10	101	76	48	180	32.9	0.171	63
764	2	122	70	27	0	36.8	0.340	27
765	5	121	72	23	112	26.2	0.245	30
766	1	126	60	0	0	30.1	0.349	47
767	1	93	70	31	0	30.4	0.315	23

768 rows × 8 columns

```
In [7]: D=df.drop(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',  
                  'BMI', 'DiabetesPedigreeFunction', 'Age'],axis=1)
```

```
In [8]: D
```

Out[8]: Outcome



```
763 0
764 0
765 0
766 1
767 0
```

768 rows x 1 columns

```
In [9]: from sklearn.model_selection import train_test_split
I_train,I_test,D_train,D_test=train_test_split(I,D,test_size=0.25,random_state=0)
```

```
In [10]: from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
dt.fit(I_train,D_train)
```

Out[10]: DecisionTreeClassifier()

```
In [11]: p=dt.predict(I_test)
```

```
In [12]: p
```

```
Out[12]: array([1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0,
0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1,
1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1,
1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1,
0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0], dtype=int64)
```

```
In [15]: D_test
```

Out[15]:

192 rows x 1 columns

Out[27]:

```
0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0,
1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0,
0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0], dtype=int64)
```

In [26]: `com=pd.DataFrame([{'ACTUAL':DT},{'PREDICTED':p}])`

In [27]: `com`

Out[27]:

	ACTUAL	PREDICTED
0	[1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, ...,	NaN
1	NaN	[1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, ...

In [28]: `from sklearn.metrics import confusion_matrix,accuracy_score`

In [29]: `confusion_matrix(DT,p)`

Out[29]: `array([[103, 27],
[25, 37]], dtype=int64)`

In [30]: `accuracy_score(DT,p)`

Out[30]: `0.7291666666666666`

In []: