

```
import numpy as np
import pandas as pd
df=pd.read_csv('/content/diabetes.csv')
df
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabet
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	
...	
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

768 rows × 9 columns

```
print(df.columns)
```

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

```
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabet
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

```
df.tail()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes
763	10	101	76	48	180	32.9	1
764	2	122	70	27	0	36.8	1
765	5	121	72	23	112	26.2	1
766	1	126	60	0	0	30.1	1
767	1	93	70	31	0	30.4	1

```
df.shape
```

(768, 9)

```
print(df.isna().sum())
```

```
Pregnancies      0
Glucose          0
BloodPressure    0
SkinThickness    0
Insulin          0
BMI              0
DiabetesPedigreeFunction  0
Age             0
Outcome          0
dtype: int64
```

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

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

X

```
array([[ 6.    , 148.   , 72.    , ..., 33.6   , 0.627, 50.    ],
       [ 1.    , 85.    , 66.    , ..., 26.6   , 0.351, 31.    ],
       [ 8.    , 183.   , 64.    , ..., 23.3   , 0.672, 32.    ],
       ...,
       [ 5.    , 121.   , 72.    , ..., 26.2   , 0.245, 30.    ],
       [ 1.    , 126.   , 60.    , ..., 30.1   , 0.349, 47.    ],
       [ 1.    , 93.    , 70.    , ..., 30.4   , 0.315, 23.    ]])
```

```
from sklearn.model_selection import train_test_split
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=1
```

\bar{x} test

```
array([[ 7.    , 136.    , 74.    , ..., 26.    , 0.647, 51.    ],
       [ 1.    , 151.    , 60.    , ..., 26.1   , 0.179, 22.    ],
       [ 6.    , 109.    , 60.    , ..., 25.     , 0.206, 27.    ],
       ...,
       [ 1.    , 91.     , 54.     , ..., 25.3    , 0.224, 23.    ],
```

```
[ 1.      , 91.      , 54.      , ..., 25.2      , 0.234, 23.      ],
[ 10.     , 168.     , 74.      , ..., 38.      , 0.537, 34.      ],
[ 9.      , 112.     , 82.      , ..., 34.2     , 0.26 , 36.      ]])
```

```
#standerd scalar (z=(x-u)/s)
# where u=mean of training data,s=std deviation of training dat
from sklearn.preprocessing import StandardScaler
scalar=StandardScaler()
scalar.fit(x_train)
x_train=scalar.transform(x_train)
x_test=scalar.transform(x_test)
x_test
```

```
array([[ 0.96714133,  0.48047259,  0.25121172, ..., -0.76147848,
         0.53734346,  1.47930079],
       [-0.82270897,  0.96100484, -0.44543516, ..., -0.74843705,
        -0.86878331, -0.94617312],
       [ 0.66883295, -0.38448546, -0.44543516, ..., -0.89189275,
        -0.78766061, -0.52798796],
       ...,
       [-0.82270897, -0.96112415, -0.74399811, ..., -0.8658099 ,
        -0.70353337, -0.86253609],
       [ 1.86206648,  1.50560806,  0.25121172, ...,  0.80349276,
         0.20684358,  0.05747126],
       [ 1.5637581 , -0.28837901,  0.64929565, ...,  0.30791853,
        -0.62541522,  0.22474532]])
```

```
#model creation
from sklearn.neighbors import KNeighborsClassifier
classifier=KNeighborsClassifier(n_neighbors=5)
classifier.fit(x_train,y_train)
y_pred=classifier.predict(x_test)
y_pred
print(classifier.predict([[1,89,66,23,94,28.1,0.167,30]]))
```

```
[1]
```

```
from sklearn.metrics._plot.confusion_matrix import confusion_matrix
from sklearn.metrics import classification_report,accuracy_score
result=confusion_matrix(y_test,y_pred)
score=accuracy_score(y_test,y_pred)
print(result)
print(score)
```

```
[[128  18]
 [ 30  55]]
0.7922077922077922
```

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