

KNN ALGORITHM

```
import numpy as np
import pandas as pd
df=pd.read_csv("D:/luminar/diabetes.csv")
df
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
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

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1
..
763	0.171	63	0
764	0.340	27	0
765	0.245	30	0
766	0.349	47	1
767	0.315	23	0

[768 rows x 9 columns]

```
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
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

	DiabetesPedigreeFunction	Age	Outcome
0	0.627	50	1
1	0.351	31	0
2	0.672	32	1
3	0.167	21	0
4	2.288	33	1

df.tail()

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
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

	DiabetesPedigreeFunction	Age	Outcome
763	0.171	63	0
764	0.340	27	0
765	0.245	30	0
766	0.349	47	1
767	0.315	23	0

df.isna().sum()

Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0

```
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

```
#sepeate
```

```
x=df.iloc[:,0:8].values
```

```
y=df.iloc[:,8].values
```

```
y
```

```
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)
```

```
x_train
```

```
array([[ 3.    , 180.    , 64.    , ..., 34.    , 0.271, 26.    ],
       [ 4.    , 156.    , 75.    , ..., 48.3 , 0.238, 32.    ],
       [ 3.    , 158.    , 70.    , ..., 35.5 , 0.344, 35.    ],
       ...,
       [ 0.    , 107.    , 62.    , ..., 36.6 , 0.757, 25.    ],
       [ 6.    , 103.    , 66.    , ..., 24.3 , 0.249, 29.    ],
       [ 0.    , 94.    , 0.    , ..., 0.    , 0.256, 25.    ]])
```

```
x_test
```

```
array([[ 1.    , 151.    , 60.    , ..., 26.1 , 0.179, 22.    ],
       [ 3.    , 123.    , 100.    , ..., 57.3 , 0.88 , 22.    ],
       [ 1.    , 163.    , 72.    , ..., 39.    , 1.222, 33.    ],
       ...,
       [ 9.    , 119.    , 80.    , ..., 29.    , 0.263, 29.    ],
       [ 4.    , 132.    , 0.    , ..., 32.9 , 0.302, 23.    ],
       [ 10.   , 75.    , 82.    , ..., 33.3 , 0.263, 38.    ]])
```

```
y_test
```

```
array([0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1,
0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1,
1,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0,
0,
       1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1,
```

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

y_train

```
array([0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
1,
    1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1,
1,
    1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,
0,
    0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0,
1,
    0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1,
1,
    0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1,
1,
    1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0,
    0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1,
    0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
0,
    0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0,
0,
    0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1,
1,
    0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1,
1,
    1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0,
0,
    0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1,
0,
    0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0,
0,
    0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
1,
    0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
```

```

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

```

#scaling

```

from sklearn.preprocessing import StandardScaler
scn=StandardScaler()
scn.fit(x_train)
x_train=scn.transform(x_train)
x_test=scn.transform(x_test)

```

x_train

```

array([[ -0.22589921,  1.8854908 , -0.25299586, ...,  0.2681189 ,
        -0.59080541, -0.59913336],
       [ 0.07436781,  1.13360474,  0.31610031, ...,  2.06237759,
        -0.69102714, -0.0820137 ],
       [ -0.22589921,  1.19626191,  0.05742023, ...,  0.45632785,
        -0.3691028 ,  0.17654613],
       ...,
       [ -1.12670025, -0.40149595, -0.35646789, ...,  0.59434775,
         0.88518729, -0.68531997],
       [ 0.67490184, -0.52681029, -0.14952383, ..., -0.94896567,
        -0.65761989, -0.34057353],
       [ -1.12670025, -0.80876756, -3.56410083, ..., -3.99795071,
        -0.63636074, -0.68531997]])

```

x_test

```

array([[ -0.82643324,  0.97696181, -0.45993992, ..., -0.72311492,
        -0.87021144, -0.9438798 ],
       [ -0.22589921,  0.09976142,  1.60950069, ...,  3.19163131,
         1.258741 , -0.9438798 ],
       [ -0.82643324,  1.35290484,  0.16089226, ...,  0.89548208,
         2.29740253,  0.00417291],
       ...,
       [ 1.57570288, -0.02555292,  0.57478038, ..., -0.35924428,
        -0.61510159, -0.34057353],
       [ 0.07436781,  0.38171869, -3.56410083, ...,  0.130099 ,

```

```
-0.49665773, -0.85769319],  
[ 1.87596989, -1.40401069,  0.67825241, ...,  0.18028806,  
 -0.61510159,  0.43510596]])
```

```
#knn==>model_creation()
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn=KNeighborsClassifier(n_neighbors=7)
```

```
knn.fit(x_train,y_train) #model creation by using training data(inp and  
out)
```

```
y_pred=knn.predict(x_test)
```

```
y_pred
```

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

```
print(knn.predict([[1,0,48,20,0,24.7,0.14,60]]))
```

```
[1]
```

```
from sklearn.metrics import confusion_matrix,accuracy_score
```

```
result=confusion_matrix(y_test,y_pred)
```

```
result
```

```
array([[120,  26],  
       [ 32,  53]], dtype=int64)
```

```
score=accuracy_score(y_test,y_pred)
```

```
score
```

```
0.7489177489177489
```