

```
In [5]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas_profiling
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
from sklearn.preprocessing import StandardScaler
```

```
In [7]: df=pd.read_csv("diabetes.csv")
df
```

Out[7]:

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	148	50	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
2	183	64	0	0	23.3	0.672	52	1
3	150	66	23	94	28.1	0.167	21	0
4	150	40	35	168	43.1	2.288	33	1
...
763	101	76	48	180	32.9	0.171	63	0
764	122	70	27	0	36.8	0.340	27	0
765	121	72	23	112	26.2	0.245	30	0
766	126	60	0	0	30.1	0.349	47	1
767	93	70	31	0	30.4	0.315	23	0

768 rows × 8 columns

```
In [8]: df.head()
```

Out[8]:

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	148	50	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
2	183	64	0	0	23.3	0.672	52	1
3	150	66	23	94	28.1	0.167	21	0
4	150	40	35	168	43.1	2.288	33	1

```
In [48]: not_zero_value=["Glucose","BloodPressure","SkinThickness","BMI"]
for column in not_zero_value:
    df[column]=df[column].replace(0,np.NaN)
    mean=int(df[column].mean(skipna=True))
    df[column].replace(np.NaN,mean,inplace=True)
```

```
In [49]: df.head()
```

Out[49]:

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	148.0	50.0	35.0	155.0	33.6	0.627	50	1
1	85.0	66.0	29.0	155.0	26.6	0.351	31	0
2	183.0	64.0	29.0	155.0	23.3	0.672	52	1
3	150.0	66.0	23.0	94.0	28.1	0.167	21	0
4	150.0	40.0	35.0	168.0	43.1	2.288	33	1

```
In [50]: x=df.drop("Outcome",axis=1)
y=df["Outcome"]
print(x)
print(y)
```

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
0	148.0	50.0	35.0	155.0	33.6	
1	85.0	66.0	29.0	155.0	26.6	
2	183.0	64.0	29.0	155.0	23.3	
3	150.0	66.0	23.0	94.0	28.1	
4	150.0	40.0	35.0	168.0	43.1	
..	
763	101.0	76.0	48.0	180.0	32.9	
764	122.0	70.0	27.0	155.0	36.8	
765	121.0	72.0	23.0	112.0	26.2	
766	126.0	60.0	29.0	155.0	30.1	
767	93.0	70.0	31.0	155.0	30.4	

	DiabetesPedigreeFunction	Age
0	0.627	50
1	0.351	31
2	0.672	52
3	0.167	21
4	2.288	33
..
763	0.171	63
764	0.340	27
765	0.245	30
766	0.349	47
767	0.315	23

[768 rows x 7 columns]

```
0    1
1    0
2    1
3    0
4    1
..
763  0
764  0
765  0
766  1
767  0
```

Name: Outcome, Length: 768, dtype: int64

```
In [51]: x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0,test_size=0.2)
print(x_train)
```

	Glucose	BloodPressure	SkinThickness	Insulin	BMI	\
603	150.0	78.0	29.0	126.0	35.2	
118	97.0	60.0	23.0	155.0	28.2	
247	165.0	90.0	33.0	680.0	52.3	
157	109.0	56.0	21.0	135.0	25.2	
468	120.0	72.0	29.0	155.0	30.0	
..	
763	101.0	76.0	48.0	180.0	32.9	
192	159.0	66.0	29.0	155.0	30.4	
629	94.0	65.0	22.0	155.0	24.7	
559	85.0	74.0	29.0	155.0	30.1	
684	136.0	82.0	29.0	155.0	32.0	

	DiabetesPedigreeFunction	Age
603	0.692	54
118	0.443	22
247	0.427	23
157	0.833	23
468	0.183	38
..
763	0.171	63
192	0.383	36
629	0.148	21
559	0.300	35
684	0.640	69

[614 rows x 7 columns]

```
In [52]: sc_x=StandardScaler()
x_train=sc_x.fit_transform(x_train)
x_test=sc_x.transform(x_test)
```

```
In [53]: knn_c=KNeighborsClassifier(n_neighbors=11,p=2,metric="euclidean")
knn_c.fit(x_train,y_train)
```

```
Out[53]: KNeighborsClassifier(metric='euclidean', n_neighbors=11)
```

```
In [54]: knn_c
```

```
Out[54]: KNeighborsClassifier(metric='euclidean', n_neighbors=11)
```

```
In [55]: import math
```

```
In [56]: y_pred=knn_c.predict(x_test)
y_pred
```

C:\Users\HP\anaconda3\lib\site-packages\sklearn\neighbors_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.

```
mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

```
Out[56]: array([[1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1,
1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
dtype=int64])
```

```
In [57]: cm=confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[92 15]
 [18 29]]
```

```
In [58]: f1=f1_score(y_test,y_pred)
print(f1)
```

```
0.6373626373626374
```

```
In [59]: acc=accuracy_score(y_test,y_pred)
print(acc)
```

```
0.7857142857142857
```

```
In [66]: input_data=(150,92,0,0,37.6,0.191,30)
input_data_as_numpy_array=np.asarray(input_data)
input_data_resaped=input_data_as_numpy_array.reshape(1,-1)
prediction=knn_c.predict(input_data_resaped)
print("our prediction is",prediction)
if prediction==1:
    print("The person having diabetes")
else:
    print("the person does not have diabetes")
```

```
our prediction is [1]
The person having diabetes
```

C:\Users\HP\anaconda3\lib\site-packages\sklearn\neighbors_classification.py:228: FutureWarning: Unlike other reduction functions (e.g. `skew`, `kurtosis`), the default behavior of `mode` typically preserves the axis it acts along. In SciPy 1.11.0, this behavior will change: the default value of `keepdims` will become False, the `axis` over which the statistic is taken will be eliminated, and the value None will no longer be accepted. Set `keepdims` to True or False to avoid this warning.

```
mode, _ = stats.mode(_y[neigh_ind, k], axis=1)
```

```
In [ ]:
```