In [1]:
```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn import svm
from sklearn.metrics import accuracy_score
```

In [2]:
```python
diabetes_dataset=pd.read_csv("diabetes.csv")
```

In [3]:
```python
diabetes_dataset.head()
```

Out[3]:

| | 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 |

In [5]:
```python
diabetes_dataset.shape
```

Out[5]:
```
(768, 9)
```

In [6]:
```python
diabetes_dataset.describe()
```

Out[6]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedigr |
|---|---|---|---|---|---|---|---|
| **count** | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | 768.000000 | |
| **mean** | 3.845052 | 120.894531 | 69.105469 | 20.536458 | 79.799479 | 31.992578 | |
| **std** | 3.369578 | 31.972618 | 19.355807 | 15.952218 | 115.244002 | 7.884160 | |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | |
| **25%** | 1.000000 | 99.000000 | 62.000000 | 0.000000 | 0.000000 | 27.300000 | |
| **50%** | 3.000000 | 117.000000 | 72.000000 | 23.000000 | 30.500000 | 32.000000 | |
| **75%** | 6.000000 | 140.250000 | 80.000000 | 32.000000 | 127.250000 | 36.600000 | |
| **max** | 17.000000 | 199.000000 | 122.000000 | 99.000000 | 846.000000 | 67.100000 | |

In [8]:
```python
diabetes_dataset['Outcome'].value_counts()
```

Out[8]:
```
Outcome
0    500
1    268
Name: count, dtype: int64
```

In [9]:
```python
diabetes_dataset.groupby('Outcome').mean()
```

Out[9]:

| | Pregnancies | Glucose | BloodPressure | SkinThickness | Insulin | BMI | DiabetesPedi |
|---|---|---|---|---|---|---|---|
| **Outcome** | | | | | | | |
| **0** | 3.298000 | 109.980000 | 68.184000 | 19.664000 | 68.792000 | 30.304200 | |
| **1** | 4.865672 | 141.257463 | 70.824627 | 22.164179 | 100.335821 | 35.142537 | |

In [14]:
```python
x=diabetes_dataset.drop(columns={'Outcome'},axis=1)
y=diabetes_dataset['Outcome']
```

In [11]:
```python
print(x)
```

```
     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
0                       0.627   50
1                       0.351   31
2                       0.672   32
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 8 columns]
```

In [12]:
```python
print(y)
```

```
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 [15]:
```python
scaler=StandardScaler()
```

In [16]:
```
scaler.fit(x)
```
Out[16]:
```
▾ StandardScaler
StandardScaler()
```
In [17]:
```
standardized_data=scaler.transform(x)
```
In [18]:
```
print(standardized_data)
```
```
[[ 0.63994726  0.84832379  0.14964075 ...  0.20401277  0.46849198
   1.4259954 ]
 [-0.84488505 -1.12339636 -0.16054575 ... -0.68442195 -0.36506078
  -0.19067191]
 [ 1.23388019  1.94372388 -0.26394125 ... -1.10325546  0.60439732
  -0.10558415]
 ...
 [ 0.3429808   0.00330087  0.14964075 ... -0.73518964 -0.68519336
  -0.27575966]
 [-0.84488505  0.1597866  -0.47073225 ... -0.24020459 -0.37110101
   1.17073215]
 [-0.84488505 -0.8730192   0.04624525 ... -0.20212881 -0.47378505
  -0.87137393]]
```
In [19]:
```
x=standardized_data
```
In [31]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,stratify=y,random_sta
```
In [32]:
```
print(x.shape,x_test.shape,x_train.shape)
```
```
(768, 8) (154, 8) (614, 8)
```
In [33]:
```
classifier=svm.SVC(kernel="linear")
```
In [34]:
```
classifier.fit(x_train,y_train)
```
Out[34]:
```
▾         SVC
SVC(kernel='linear')
```
In [35]:
```
x_train_prediction=classifier.predict(x_train)
training_data_accuracy=accuracy_score(x_train_prediction,y_train)
```
In [36]:
```
print(training_data_accuracy)
```
```
0.7866449511400652
```
In [37]:
```
x_test_prediction=classifier.predict(x_test)
x_test_accuracy=accuracy_score(x_test_prediction,y_test)
```
In [38]:
```
print(x_test_accuracy)
```
```
0.7727272727272727
```

```
In [56]: input=(5,137,108,0,0,48.8,0.227,37)
         input_data_as_numpy_array=np.asarray(input)
```

```
In [57]: input_data_as_numpy_array_reshape=input_data_as_numpy_array.reshape(1,-1)
```

```
In [58]: import warnings
         warnings.filterwarnings("ignore", category=UserWarning)


         std_data=scaler.transform(input_data_as_numpy_array_reshape)
         std_data
```

```
Out[58]: array([[ 0.3429808 ,  0.5040552 ,  2.01075975, -1.28821221, -0.69289057,
                  2.1331853 , -0.73955549,  0.31985461]])
```

```
In [59]: prediction=classifier.predict(std_data)
```

```
In [60]: print(prediction)
```

```
[1]
```

```
In [61]: if prediction==0:
             print("the person does not have diabetes")
         else:
             print("the person has diabetes")
```

```
the person has diabetes
```

```
In [ ]:
```