

Importing the libraries

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

Importing the dataset

```
In [2]: data = pd.read_csv('diabetes.csv')
```

```
In [3]: data.head(5)
```

```
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 [4]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   Pregnancies      768 non-null    int64  
 1   Glucose          768 non-null    int64  
 2   BloodPressure    768 non-null    int64  
 3   SkinThickness    768 non-null    int64  
 4   Insulin          768 non-null    int64  
 5   BMI              768 non-null    float64 
 6   DiabetesPedigreeFunction 768 non-null    float64 
 7   Age              768 non-null    int64  
 8   Outcome          768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

```
In [5]: data.describe()
```

```
Out[5]:    Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI  DiabetesPedigreeFunction
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
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
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 [6]: `data.groupby('Outcome').mean()`

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
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	

Separating into features and labels

In [7]: `X = data.iloc[:, :-1].values
Y = data.iloc[:, -1].values`

Train - Test split

In [8]: `X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify = Y)`

feature scaling

In [9]: `sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)`

Training the model

In [10]: `classifier = SVC(kernel = 'linear')`

In [11]: `classifier.fit(X_train, Y_train)`

Out[11]: `SVC(kernel='linear')`

Evaluation

In [12]: `X_train_pred = classifier.predict(X_train)`

In [13]: `accuracy_score(X_train_pred, Y_train)`

Out[13]: `0.7964169381107492`

```
In [14]: X_test_pred = classifier.predict(X_test)
```

```
In [15]: accuracy_score(X_test_pred, Y_test)
```

```
Out[15]: 0.7597402597402597
```

Predictions

```
In [16]: data.head(2)
```

```
Out[16]:    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
```

```
In [17]: std_input_data = sc.transform([[4,110,92,35,0,35,0.45,35]])
```

```
In [18]: prediction = classifier.predict(std_input_data)
```

```
In [19]: prediction
```

```
Out[19]: array([0], dtype=int64)
```

```
In [20]: if (prediction[0] == 0):
    print('Person is non-diabetic')
else:
    print('Person is diabetic')
```

```
Person is non-diabetic
```

```
In [ ]:
```