SVM

In [ ]:

```
EXP:12
```

In [2]:

```
#Aim : To perform and find the accuracy of Support Vector Machine Algorithm i.e. SV
```

In [50]:

```
#Name: Leena Rajeshwar Kale
#Roll No.:71
#Sec: C
#Subject:ET - 1
```

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
import os
```

In [3]:

```
os.getcwd()
```

Out[3]:

```
'C:\\Users\\dishi\\Downloads\\ET'
```

In [4]:

```
os.chdir("C:\\Users\\dishi\\Downloads\\ET")
```

In [5]:

```
df=pd.read_csv("framingham.csv")
```

In [15]:

```
df.head()
```

Out[15]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	dia
0	1	39	4.0	0	0.0	0.0	0	0	
1	0	46	2.0	0	0.0	0.0	0	0	
2	1	48	1.0	1	20.0	0.0	0	0	
3	0	61	3.0	1	30.0	0.0	0	1	
4	0	46	3.0	1	23.0	0.0	0	0	

In [17]:

```
df.describe()
```

Out[17]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalent
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.0
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.0
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.0
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.0
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.0
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.0
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.0
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.0

In [19]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   male             4238 non-null   int64  
 1   age              4238 non-null   int64  
 2   education        4133 non-null   float64
 3   currentSmoker    4238 non-null   int64  
 4   cigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   prevalentStroke  4238 non-null   int64  
 7   prevalentHyp     4238 non-null   int64  
 8   diabetes         4238 non-null   int64  
 9   totChol          4188 non-null   float64
 10  sysBP            4238 non-null   float64
 11  diaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
 13  heartRate        4237 non-null   float64
 14  glucose          3850 non-null   float64
 15  TenYearCHD       4238 non-null   int64  
dtypes: float64(9), int64(7)
memory usage: 529.9 KB
```

In [21]:

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

Out[21]:

```
male                 0
age                  0
education          105
currentSmoker        0
cigsPerDay          29
BPMeds              53
prevalentStroke      0
prevalentHyp         0
diabetes             0
totChol             50
sysBP                0
diaBP                0
BMI                 19
heartRate            1
glucose            388
TenYearCHD           0
dtype: int64
```

Missing Value Tretment

In [34]:

```
df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)
```

In [26]:

```
df['education'].fillna(value = df['education'].mean(),inplace=True)
```

In [28]:

```
df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)
```

In [30]:

```
df['BPMeds'].fillna(value = df['BPMeds'].mean(),inplace=True)
```

In [32]:

```
df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)
```

In [36]:

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

Out[36]:

```
male               0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                0
heartRate          0
glucose            0
TenYearCHD         0
dtype: int64
```

In [38]:

```
#Splitting the dependent and independent variables.
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

In [40]:

```
x
```

Out[40]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	1	39	4.0	0	0.0	0.00000	0	0
1	0	46	2.0	0	0.0	0.00000	0	0
2	1	48	1.0	1	20.0	0.00000	0	0
3	0	61	3.0	1	30.0	0.00000	0	1
4	0	46	3.0	1	23.0	0.00000	0	0
...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.00000	0	1
4234	1	51	3.0	1	43.0	0.00000	0	0
4235	0	48	2.0	1	20.0	0.02963	0	0
4236	0	44	1.0	1	15.0	0.00000	0	0
4237	0	52	2.0	0	0.0	0.00000	0	0

4238 rows × 15 columns

Train Test Split

In [43]:

```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

In [45]:

```
x_train
```

Out[45]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
3252	1	40	4.0	1	30.0	0.0	0	0
3946	0	57	2.0	0	0.0	0.0	0	1
1261	0	47	1.0	0	0.0	0.0	0	0
2536	1	41	2.0	1	30.0	0.0	0	0
4089	0	64	1.0	0	0.0	0.0	0	1
...	...	...	...	...	...	...	...	...
3444	0	36	1.0	1	5.0	0.0	0	1
466	0	57	3.0	1	15.0	0.0	0	0
3092	0	60	2.0	0	0.0	0.0	0	1
3772	1	39	2.0	1	10.0	0.0	0	0
860	0	35	2.0	0	0.0	0.0	0	0

3390 rows × 15 columns

SVM Classifier

In [48]:

```
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
svc=SVC()
svc.fit(x_test,y_test)
acc = svc.score(x_test,y_test)*100
print(acc)
```

```
85.37735849056604
```

In [ ]: