

SVM classifier

```
In [1]: 1 #Name : Achal Gajanan Ghorad
        2 #Roll no. 39
        3 #Section :3A
        4 #subject:E.T.1
        5 #Date:5/10/2024
```

```
In [2]: 1 #Aim: to perform operation on svm classifier
```

```
In [3]: 1 import pandas as pd
        2 import matplotlib.pyplot as plt
        3 import numpy as np
        4 import seaborn as sns
        5 from sklearn.model_selection import train_test_split
        6 import warnings
        7 warnings.filterwarnings('ignore')
        8
```

```
In [4]: 1 import os
```

```
In [5]: 1 os.getcwd()
```

Out[5]: 'C:\\Users\\ACHAL'

```
In [6]: 1 os.chdir("C:\\Users\\ACHAL\\OneDrive\\Desktop")
```

```
In [7]: 1 df=pd.read_csv("C:\\Users\\ACHAL\\OneDrive\\Desktop\\framingham1.csv")
```

```
In [8]: 1 df.head()
```

Out[8]:

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

In [9]: 1 df.describe()

Out[9]:

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

In [10]: 1 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 [11]: 1 df.isna().sum()

```
Out[11]: 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
```

In [12]: 1 df

Out[12]:

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

4238 rows × 16 columns

Missing value treatment

In [13]: 1 df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)

In [14]: 1 df['education'].fillna(value = df['education'].mean(),inplace=True)

In [15]: 1 df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)

```
In [16]: 1 df['BMI'].fillna(value = df['BMI'].mean(),inplace=True)
        2
```

```
In [17]: 1 df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)
        2
```

```
In [18]: 1 df['totChol'].fillna(value = df['totChol'].mean(),inplace=True)
        2
```

```
In [19]: 1 df['BPMeds'].fillna(value = df['BPMeds'].mean(),inplace=True)
```

```
In [20]: 1 df.isna().sum()
```

```
Out[20]: 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 [21]: 1 #Splitting the dependent and independent variables.
        2 x = df.drop("TenYearCHD",axis=1)
        3 y = df['TenYearCHD']
```

In [22]: 1 x *#checking the features*

Out[22]:

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

4238 rows × 15 columns



Train test split

In [23]: 1 x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,

In [24]: 1 y_train

Out[24]: 3252 0
 3946 0
 1261 0
 2536 0
 4089 0
 ..
 3444 0
 466 0
 3092 0
 3772 0
 860 0
 Name: TenYearCHD, Length: 3390, dtype: int64

SVM classifier

In [25]:



```
1 from sklearn.svm import SVC
2 from sklearn.metrics import accuracy_score
3 svc=SVC()
4 svc.fit(x_test,y_test)
5 acc = svc.score(x_test,y_test)*100
6 print(acc)
```

85.37735849056604

In []:



1