

```
In [1]: #Name : Ashwini V Kayande
#Roll No : 60
#Section : 3A
#Date :05/10/2024
```

```
In [3]: #Aim : To perform operation on support vector machine(SVM Classifier)
```

```
In [5]: 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 [7]: import os
```

```
In [9]: os.getcwd()
```

```
Out[9]: 'C:\\\\Users\\user'
```

```
In [11]: os.chdir("C:\\\\Users\\user\\Desktop")
```

```
In [13]: df=pd.read_csv("framingham.csv")
```

```
In [15]: #The "Framingham" heart disease dataset includes over 4,240 records, 15 attributes
#The goal of the dataset is to predict whether the patient has 10-year risk
```

```
In [17]: df.head()
```

```
Out[17]:
```

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

```
In [19]: df.describe()
```

Out[19]:

	male	age	education	currentSmoker	cigsPerDay	
<b>count</b>	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	41
<b>mean</b>	0.429212	49.584946	1.978950	0.494101	9.003089	
<b>std</b>	0.495022	8.572160	1.019791	0.500024	11.920094	
<b>min</b>	0.000000	32.000000	1.000000	0.000000	0.000000	
<b>25%</b>	0.000000	42.000000	1.000000	0.000000	0.000000	
<b>50%</b>	0.000000	49.000000	2.000000	0.000000	0.000000	
<b>75%</b>	1.000000	56.000000	3.000000	1.000000	20.000000	
<b>max</b>	1.000000	70.000000	4.000000	1.000000	70.000000	

In [21]: `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 [23]: `df.isna().sum()`

```
Out[23]: 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 [25]: #Since, only a few rows have null values in them, we are only removing those
#df = df.dropna(subset=['heartRate', 'BMI', 'cigsPerDay', 'totChol', 'BPMeds'])
```

```
In [27]: df
```

```
Out[27]:
```

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

4238 rows × 16 columns

```
In [29]: df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)
```

```
In [31]: df['education'].fillna(value = df['education'].mean(),inplace=True)
```

```
In [33]: df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)
```

```
In [35]: df['BMI'].fillna(value = df['BMI'].mean(),inplace=True)
```

```
In [37]: df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)
```

```
In [39]: df['totChol'].fillna(value = df['totChol'].mean(),inplace=True)
```

```
In [41]: df['BPMeds'].fillna(value = df['BPMeds'].mean(),inplace=True)
```

```
In [43]: df.isna().sum()
```

```
Out[43]: 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 [45]: #Splitting the dependent and independent variables.
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

```
In [47]: x #checking the features
```

```
Out[47]:
```

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

4238 rows × 15 columns

```
In [49]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_st
```

```
In [51]: y_train
```

```
Out[51]: 2754    0
          2184    0
          875    0
          693    0
          3249   0
          ..
          57     0
          3273   0
          2706    1
          578    0
          2439   0
          Name: TenYearCHD, Length: 3390, dtype: int64
```

```
In [53]: 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)
```

```
84.19811320754717
```

```
In [ ]:
```