

In [17]: `#Name : Anlita Gulde`

`#Roll no : 44`

`#Section : 3A`

In [18]: `#Aim : To perform operation on support vector machine(SVM Classifier)`

In [20]: `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 [21]: `import os`

```
In [22]: os.getcwd()
```

```
Out[22]: 'C:\\Users\\HP\\Desktop'
```

```
In [23]: os.chdir("C:\\Users\\HP\\Desktop")
```

```
In [24]: df=pd.read_csv("framingham.csv")
```

```
In [25]: #The "Framingham" heart disease dataset includes over 4,240 records, 15 attri  
#The goal of the dataset is to predict whether the patient has 10-year risk of
```

```
In [26]: df.head()
```

```
Out[26]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHypertension
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 [27]: df.describe()
```

```
Out[27]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.000000
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

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentS
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.0

In [28]: `df.info()`

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

Checking for discrepancy in data

In [29]: `df.isna().sum()`

```
Out[29]:
male                0
age                 0
education           105
currentSmoker        0
cigsPerDay           29
BPMeds               53
prevalentStroke      0
prevalentHyp         0
diabetes             0
totChol              50
sysBP                0
diaBP                0
```

In [30]: *#Since, only a few rows have null values in them, we are only removing those*
#df = df.dropna(subset=['heartRate', 'BMI', 'cigsPerDay', 'totChol', 'BPMeds'])

```
heartRate           1
glucose             388
```

TenYearCHD 0
dtype: int64

In [31]: df

Out[31]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	d
	0	1	39	4.0	0	0.0	0.0	0	0

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHypertension
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
...
4233	1	50	1.0	1	1.0	0.0	0	1
4234	1	51	3.0	1	43.0	0.0	0	0
4235	0	48	2.0	1	20.0	NaN	0	0
4236	0	44	1.0	1	15.0	0.0	0	0
4237	0	52	2.0	0	0.0	0.0	0	0

4238 rows × 9 columns

Missing Value Treatment

Since, 'glucose' and 'education' columns had a significant amount of null values, so we replaced them with the mean of values for their respective columns

```
In [32]: df['glucose'].fillna(value = df['glucose'].mean(),inplace=True)

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

In [34]: df['heartRate'].fillna(value = df['heartRate'].mean(),inplace=True)

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

In [36]: df['cigsPerDay'].fillna(value = df['cigsPerDay'].mean(),inplace=True)

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

In [38]: df['BPMeds'].fillna(value = df['BPMeds'].mean(),inplace=True)

In [39]: df.isna().sum()

Out[39]: male                0
age                0
education          0
currentSmoker      0
cigsPerDay         0
BPMeds             0
prevalentStroke    0
```

prevalent
Hyp
0
diabetes
0
totChol
0
sysBP
0
diaBP
0
BMI
0

```
heartRate      0
glucose        0
TenYearCHD     0
dtype: int64
```

Logistic Regression Model

```
In [42]: #Splitting the dependent and independent variables.
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

```
In [43]: x #checking the features
```

```
Out[43]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHypertension
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 [44]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

```
In [46]: y_train
```

```
Out[46]: 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 Clas sifier


```
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 []: