In [ ]:
```
#Aim: To Perform Operation on SVM Classifier
```

In [ ]:
```
# Name : Shruti Anil Dhote
# Roll no : 72
# Sec: C
# Subject : ET1
# Date :27/09/2024
```

In [2]:
```
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 [3]:
```
import os
```

In [4]:
```
os.getcwd()
```

Out[4]:
```
'C:\\Users\\SURUTI DHOTE'
```

In [5]:
```
os.chdir('C:\\Users\\SURUTI DHOTE\\DESKTOP')
```

In [6]:
```
df=pd.read_csv("framingham.csv")
```

In [7]:
```
df.head()
```

Out[7]:

	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 [8]:
```
df.tail()
```

Out[8]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
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	1	40	3.0	0	0.0	0.0	0	1
4239	0	39	3.0	1	30.0	0.0	0	0

In [9]:
```
df.shape
```

Out[9]:
```
(4240, 16)
```

In [10]:
```
df.size
```

Out[10]:
```
67840
```

In [11]:
```
df.info
```

Out[11]:
```
<bound method DataFrame.info of       male  age  education  currentSmoker  cigsPer
Day  BPMeds  \
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
...    ...  ...        ...            ...         ...     ...
4235     0   48        2.0              1        20.0     NaN
4236     0   44        1.0              1        15.0     0.0
4237     0   52        2.0              0         0.0     0.0
4238     1   40        3.0              0         0.0     0.0
4239     0   39        3.0              1        30.0     0.0

      prevalentStroke  prevalentHyp  diabetes  totChol  sysBP  diaBP    BMI  \
0                   0             0         0    195.0  106.0   70.0  26.97
1                   0             0         0    250.0  121.0   81.0  28.73
2                   0             0         0    245.0  127.5   80.0  25.34
3                   0             1         0    225.0  150.0   95.0  28.58
4                   0             0         0    285.0  130.0   84.0  23.10
...               ...           ...       ...      ...    ...    ...    ...
4235                0             0         0    248.0  131.0   72.0  22.00
4236                0             0         0    210.0  126.5   87.0  19.16
4237                0             0         0    269.0  133.5   83.0  21.47
4238                0             1         0    185.0  141.0   98.0  25.60
4239                0             0         0    196.0  133.0   86.0  20.91

      heartRate  glucose  TenYearCHD
0          80.0     77.0           0
1          95.0     76.0           0
2          75.0     70.0           0
3          65.0    103.0           1
4          85.0     85.0           0
...         ...      ...         ...
4235       84.0     86.0           0
4236       86.0      NaN           0
4237       80.0    107.0           0
4238       67.0     72.0           0
4239       85.0     80.0           0

[4240 rows x 16 columns]>
```

In [12]:
```
df.describe
```

Out[12]:
```
<bound method NDFrame.describe of       male  age  education  currentSmoker  cigsP
erDay  BPMeds  \
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
...    ...  ...        ...            ...         ...     ...
4235     0   48        2.0              1        20.0     NaN
4236     0   44        1.0              1        15.0     0.0
4237     0   52        2.0              0         0.0     0.0
4238     1   40        3.0              0         0.0     0.0
4239     0   39        3.0              1        30.0     0.0

      prevalentStroke  prevalentHyp  diabetes  totChol  sysBP  diaBP    BMI  \
0                   0             0         0    195.0  106.0   70.0  26.97
1                   0             0         0    250.0  121.0   81.0  28.73
2                   0             0         0    245.0  127.5   80.0  25.34
3                   0             1         0    225.0  150.0   95.0  28.58
4                   0             0         0    285.0  130.0   84.0  23.10
...               ...           ...       ...      ...    ...    ...    ...
4235                0             0         0    248.0  131.0   72.0  22.00
4236                0             0         0    210.0  126.5   87.0  19.16
4237                0             0         0    269.0  133.5   83.0  21.47
4238                0             1         0    185.0  141.0   98.0  25.60
4239                0             0         0    196.0  133.0   86.0  20.91

      heartRate  glucose  TenYearCHD
0          80.0     77.0           0
1          95.0     76.0           0
2          75.0     70.0           0
3          65.0    103.0           1
4          85.0     85.0           0
...         ...      ...         ...
4235       84.0     86.0           0
4236       86.0      NaN           0
4237       80.0    107.0           0
4238       67.0     72.0           0
4239       85.0     80.0           0

[4240 rows x 16 columns]>
```

In [13]:
```
df
```

Out[13]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
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
...	...	...	...	...	...	...	...	...
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	1	40	3.0	0	0.0	0.0	0	1
4239	0	39	3.0	1	30.0	0.0	0	0

4240 rows × 16 columns

In [14]:
```
df.isna().sum()
```

Out[14]:
```
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 [15]:
```
df.isnull()
```

Out[15]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
4235	False	False	False	False	False	True	False	False
4236	False	False	False	False	False	False	False	False
4237	False	False	False	False	False	False	False	False
4238	False	False	False	False	False	False	False	False
4239	False	False	False	False	False	False	False	False

4240 rows × 16 columns

In [16]:
```
df.isnull().any()
```

Out[16]:
```
male               False
age                False
education           True
currentSmoker      False
cigsPerDay          True
BPMeds              True
prevalentStroke    False
prevalentHyp       False
diabetes           False
totChol             True
sysBP              False
diaBP              False
BMI                 True
heartRate           True
glucose             True
TenYearCHD         False
dtype: bool
```

Missing Value Treatment

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

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

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

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

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

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

In [18]:
```
df.isna().sum()
```

Out[18]:
```
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 [19]:
```
#Splitting the dependent and independent variables.
x = df.drop("TenYearCHD",axis=1)
y = df['TenYearCHD']
```

In [20]:
```
x
```

Out[20]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	1	39	4.0	0	0.0	0.000000	0	0
1	0	46	2.0	0	0.0	0.000000	0	0
2	1	48	1.0	1	20.0	0.000000	0	0
3	0	61	3.0	1	30.0	0.000000	0	1
4	0	46	3.0	1	23.0	0.000000	0	0
...	...	...	...	...	...	...	...	...
4235	0	48	2.0	1	20.0	0.029615	0	0
4236	0	44	1.0	1	15.0	0.000000	0	0
4237	0	52	2.0	0	0.0	0.000000	0	0
4238	1	40	3.0	0	0.0	0.000000	0	1
4239	0	39	3.0	1	30.0	0.000000	0	0

4240 rows × 15 columns

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

In [22]:
```
x #checking the features
```

Out[22]:

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp
0	1	39	4.0	0	0.0	0.000000	0	0
1	0	46	2.0	0	0.0	0.000000	0	0
2	1	48	1.0	1	20.0	0.000000	0	0
3	0	61	3.0	1	30.0	0.000000	0	1
4	0	46	3.0	1	23.0	0.000000	0	0
...	...	...	...	...	...	...	...	...
4235	0	48	2.0	1	20.0	0.029615	0	0
4236	0	44	1.0	1	15.0	0.000000	0	0
4237	0	52	2.0	0	0.0	0.000000	0	0
4238	1	40	3.0	0	0.0	0.000000	0	1
4239	0	39	3.0	1	30.0	0.000000	0	0

4240 rows × 15 columns

In [23]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=42)
```

In [24]:
```
y_train
```

Out[24]:
```
1427    0
3257    0
3822    0
1263    0
3575    0
       ..
3444    0
466     0
3092    0
3772    0
860     0
Name: TenYearCHD, Length: 3392, dtype: int64
```

KNN

In [25]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=5, p=2, metric='minkowski')
knn.fit(x_train, y_train)
acc = knn.score(x_test,y_test)*100
print(acc)
```

```
84.19811320754717
```

In [ ]: