In [332…
```
#Name:Khushi Chandrashekhar Satpute
#Aim:-To perform and analysis of Logistic Regression Algorithm
#Roll no:43
#Section : B
#Subject : ET-II
```

In [334…
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
```

In [336…
```
os.getcwd()
```

Out[336… 'C:\\Users\\asus\\Downloads'

In [338…
```
data=pd.read_csv("framingham.csv")
```

In [340…
```
data.head()
```

Out[340…

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP	
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	106.0	70.0	26
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	121.0	81.0	28
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	127.5	80.0	25
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	150.0	95.0	28
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	130.0	84.0	23

In [342…
```
data.tail()
```

Out[342…

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP
4233	1	50	1.0	1	1.0	0.0	0	1	0	313.0	179.0	92.0
4234	1	51	3.0	1	43.0	0.0	0	0	0	207.0	126.5	80.0
4235	0	48	2.0	1	20.0	NaN	0	0	0	248.0	131.0	72.0
4236	0	44	1.0	1	15.0	0.0	0	0	0	210.0	126.5	87.0
4237	0	52	2.0	0	0.0	0.0	0	0	0	269.0	133.5	83.0

In [344…
```
data.shape
```

Out[344… (4238, 16)

In [346…
```
data.size
```

Out[346… 67808

In [348…
```
data.ndim
```

Out[348… 2

In [350…
```
data.columns
```

Out[350…
```
Index(['male', 'age', 'education', 'currentSmoker', 'cigsPerDay', 'BPMeds',
       'prevalentStroke', 'prevalentHyp', 'diabetes', 'totChol', 'sysBP',
       'diaBP', 'BMI', 'heartRate', 'glucose', 'TenYearCHD'],
      dtype='object')
```

In [352…
```
data.describe()
```

Out[352…

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabete
count	4238.000000	4238.000000	4133.000000	4238.000000	4209.000000	4185.000000	4238.000000	4238.000000	4238.00000
mean	0.429212	49.584946	1.978950	0.494101	9.003089	0.029630	0.005899	0.310524	0.02572
std	0.495022	8.572160	1.019791	0.500024	11.920094	0.169584	0.076587	0.462763	0.15831
min	0.000000	32.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
25%	0.000000	42.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
50%	0.000000	49.000000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000
75%	1.000000	56.000000	3.000000	1.000000	20.000000	0.000000	0.000000	1.000000	0.00000
max	1.000000	70.000000	4.000000	1.000000	70.000000	1.000000	1.000000	1.000000	1.00000

In [354…

```
data.isna()
```

Out[354…

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaB
0	False	False	False	False	False	False	False	False	False	False	False	Fals
1	False	False	False	False	False	False	False	False	False	False	False	Fals
2	False	False	False	False	False	False	False	False	False	False	False	Fals
3	False	False	False	False	False	False	False	False	False	False	False	Fals
4	False	False	False	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	
4233	False	False	False	False	False	False	False	False	False	False	False	Fals
4234	False	False	False	False	False	False	False	False	False	False	False	Fals
4235	False	False	False	False	False	True	False	False	False	False	False	Fals
4236	False	False	False	False	False	False	False	False	False	False	False	Fals
4237	False	False	False	False	False	False	False	False	False	False	False	Fals

4238 rows × 16 columns

In [356…

```
data.isna().any()
```

Out[356…

```
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
```

In [358…

```
data.isna().sum()
```

Out[358…

```
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 Treatment

In [361…
```
data['glucose'].fillna(value = data['glucose'].mean(),inplace=True)
data['education'].fillna(value = data['education'].mean(),inplace=True)
data['heartRate'].fillna(value = data['heartRate'].mean(),inplace=True)
```

In [363…
```
data['BMI'].fillna(value = data['BMI'].mean(),inplace=True)
data['totChol'].fillna(value = data['totChol'].mean(),inplace=True)
data['cigsPerDay'].fillna(value = data['cigsPerDay'].mean(),inplace=True)
data['BPMeds'].fillna(value = data['BPMeds'].mean(),inplace=True)
```

In [365…
```
#Splitting the dependent and independent variables.
x= data.drop("TenYearCHD",axis=1)
y = data['TenYearCHD']
```

In [367…
```
x #checking the features
```

Out[367…

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP	diaBP
0	1	39	4.0	0	0.0	0.00000	0	0	0	195.0	106.0	70.0
1	0	46	2.0	0	0.0	0.00000	0	0	0	250.0	121.0	81.0
2	1	48	1.0	1	20.0	0.00000	0	0	0	245.0	127.5	80.0
3	0	61	3.0	1	30.0	0.00000	0	1	0	225.0	150.0	95.0
4	0	46	3.0	1	23.0	0.00000	0	0	0	285.0	130.0	84.0
...	...	...	...	...	...	...	...	...	...	...	...	...
4233	1	50	1.0	1	1.0	0.00000	0	1	0	313.0	179.0	92.0
4234	1	51	3.0	1	43.0	0.00000	0	0	0	207.0	126.5	80.0
4235	0	48	2.0	1	20.0	0.02963	0	0	0	248.0	131.0	72.0
4236	0	44	1.0	1	15.0	0.00000	0	0	0	210.0	126.5	87.0
4237	0	52	2.0	0	0.0	0.00000	0	0	0	269.0	133.5	83.0

4238 rows × 15 columns

In [369…
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

In [371…
```
y_train
```

Out[371…
```
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
```

Logistic Regression Algorithm

In [383…
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression().fit(x_train,y_train)
model.score(x_train,y_train)
```

Out[383…
```
0.848377581120944
```

In [ ]:

In [ ]: