

```
In [1]: ## Importing Libraries
import pandas as pd
from sklearn.linear_model import LogisticRegression
```

```
In [2]: df = pd.read_csv(r"C:\Users\hp\Documents\Datasets\framingham.csv")
df.head()
```

```
Out[2]:
```

	male	age	education	currentSmoker	cigsPerDay	BPMeds	prevalentStroke	prevalentHyp	diabetes	totChol	sysBP
0	1	39	4.0	0	0.0	0.0	0	0	0	195.0	1
1	0	46	2.0	0	0.0	0.0	0	0	0	250.0	1
2	1	48	1.0	1	20.0	0.0	0	0	0	245.0	1
3	0	61	3.0	1	30.0	0.0	0	1	0	225.0	1
4	0	46	3.0	1	23.0	0.0	0	0	0	285.0	1

## Checking Null Values in Dataframe

```
In [5]: df.isnull().sum()
```

```
Out[5]: 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 [8]: mn = df["totChol"].mean()
mn
```

```
Out[8]: 236.72158548233045
```

```
In [11]: gm = df["glucose"].mean()
gm
```

```
Out[11]: 81.96675324675324
```

```
In [9]: df["totChol"].fillna(value=mn,inplace=True)
```

```
In [12]: df["glucose"].fillna(value=gm,inplace=True)
```

```
In [13]: df.isnull().sum()
```

```
Out[13]: male                0
age                0
education          105
currentSmoker      0
cigsPerDay         29
BPMeds             53
prevalentStroke    0
prevalentHyp       0
diabetes           0
totChol            0
sysBP              0
diaBP              0
BMI                19
heartRate          1
glucose            0
TenYearCHD         0
dtype: int64
```

## Removing Missing Values

```
In [15]: df.dropna(axis=0,how="any",inplace=True)
```

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

```
Out[16]: 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 [17]: lg = LogisticRegression()
```

## Splitting Data For Training

```
In [20]: X = df.drop(["TenYearCHD"],axis=1)
Y = df["TenYearCHD"]
```

```
In [18]: from sklearn.model_selection import train_test_split
```

```
In [21]: X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=1)
```

## Fitting Model

```
In [22]: model = lg.fit(X_train, Y_train)
```

```
C:\Users\hp\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: Convergence
Warning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

Increase the number of iterations (max\_iter) or scale the data as shown in:  
<https://scikit-learn.org/stable/modules/preprocessing.html>  
Please also refer to the documentation for alternative solver options:  
[https://scikit-learn.org/stable/modules/linear\\_model.html#logistic-regression](https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)

```
n_iter i = check_optimize_result(
```

```
In [26]: Pred = model.predict(X_test)
          Pred
```

[illegible]

## Finding Accuracy

```
In [24]: from sklearn.metrics import accuracy_score
```

```
In [27]: accuracy_score(Y_test, Pred)
```

Out[27]: 0.8451053283767038

## Correlation between Variables

```
In [29]: df[df.columns[1:]].corr()['TenYearCHD'][:].sort_values(ascending=False).to_frame()
```

Out[29]:

	TenYearCHD
TenYearCHD	1.000000
age	0.223880
sysBP	0.215551
prevalentHyp	0.175803
diaBP	0.147938
glucose	0.115993
diabetes	0.091731
BPMeds	0.090103
totChol	0.081770
BMI	0.072172
cigsPerDay	0.057096
prevalentStroke	0.044395
heartRate	0.021626
currentSmoker	0.021320
education	-0.053281

```
In [ ]:
```