# Institute of Information Technology (IIT)

Jahangirnagar University

## Lab Report: 05

Submitted by:

Name: Md. Shakil Hossain

Roll No: 2023

Lab Date: 08/08/2023

Submission Date: 18/08/2023

# Lab Report # Day 05

## Example 1:

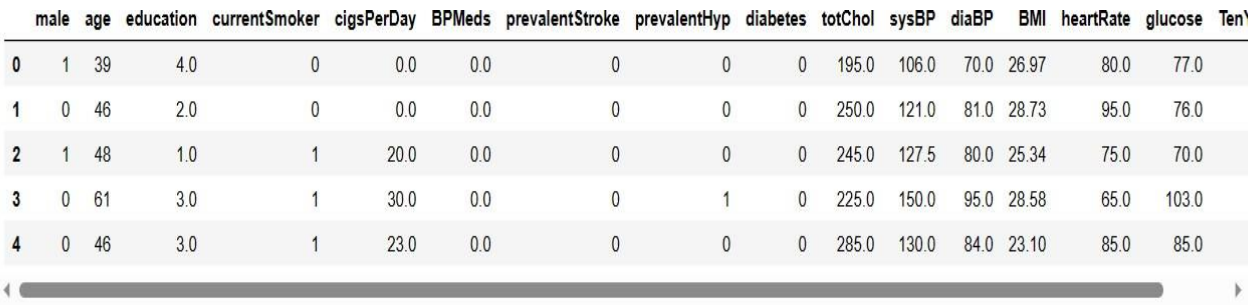
Apply logistic íegíession techniques to píedict CHD based on the píovided featuíes in the attached Dataset. (Given fíamingham.csv)

### Import Libraries & Read CSV File:

### Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import math
%matplotlib inline

data=pd.read_csv("framingham.csv")
data.head()
```

### Output:

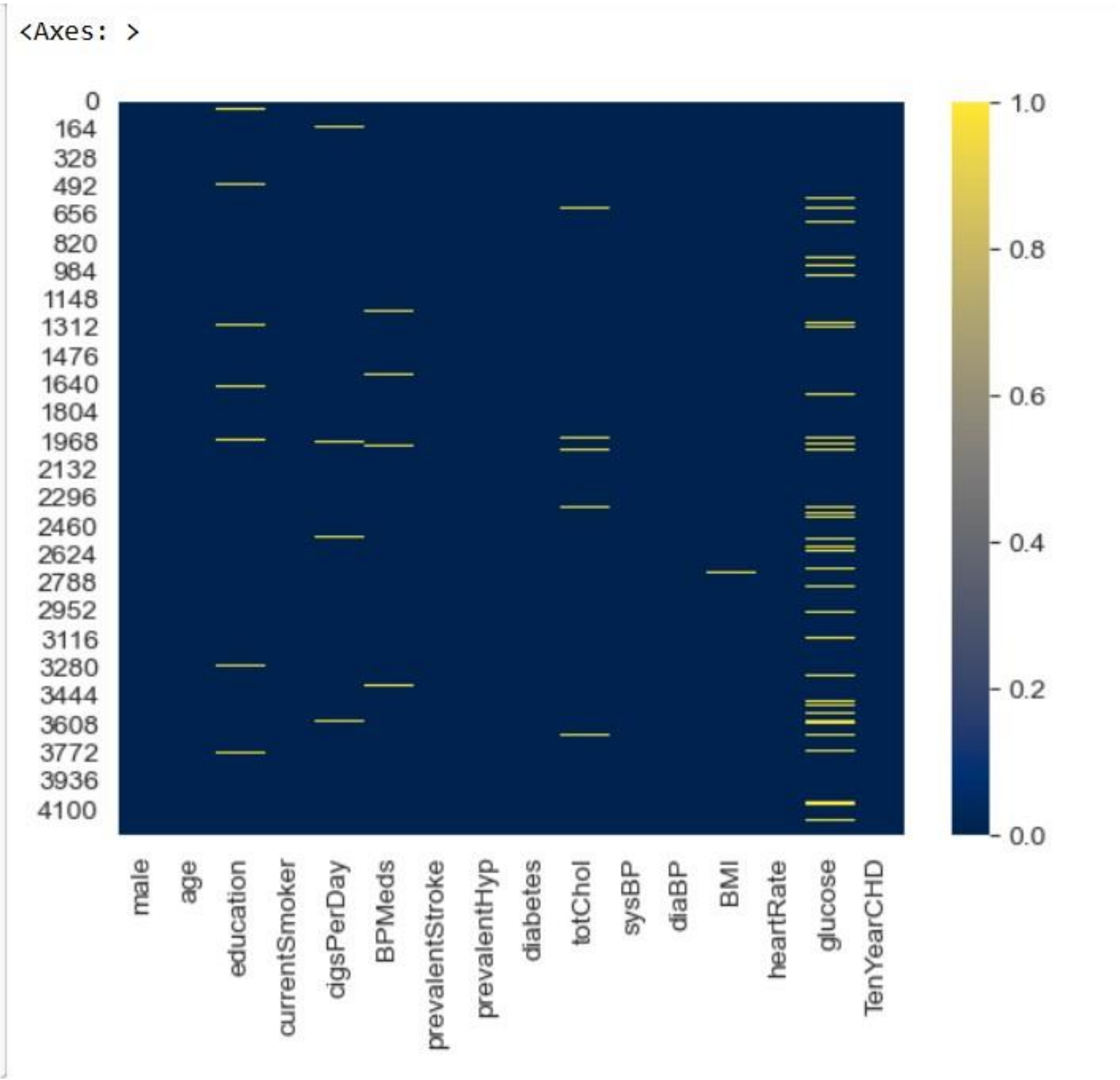


| | male | age | education | currentSmoker | cigsPerDay | BPMeds | prevalentStroke | prevalentHyp | diabetes | totChol | sysBP | diaBP | BMI | heartRate | glucose | TenY |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 39 | 4.0 | 0 | 0.0 | 0.0 | 0 | 0 | 0 | 195.0 | 106.0 | 70.0 | 26.97 | 80.0 | 77.0 | |
| **1** | 0 | 46 | 2.0 | 0 | 0.0 | 0.0 | 0 | 0 | 0 | 250.0 | 121.0 | 81.0 | 28.73 | 95.0 | 76.0 | |
| **2** | 1 | 48 | 1.0 | 1 | 20.0 | 0.0 | 0 | 0 | 0 | 245.0 | 127.5 | 80.0 | 25.34 | 75.0 | 70.0 | |
| **3** | 0 | 61 | 3.0 | 1 | 30.0 | 0.0 | 0 | 1 | 0 | 225.0 | 150.0 | 95.0 | 28.58 | 65.0 | 103.0 | |
| **4** | 0 | 46 | 3.0 | 1 | 23.0 | 0.0 | 0 | 0 | 0 | 285.0 | 130.0 | 84.0 | 23.10 | 85.0 | 85.0 | |

**Code:**

```
sns.heatmap(data.isnull(),cmap='cividis')
```

**Output:**

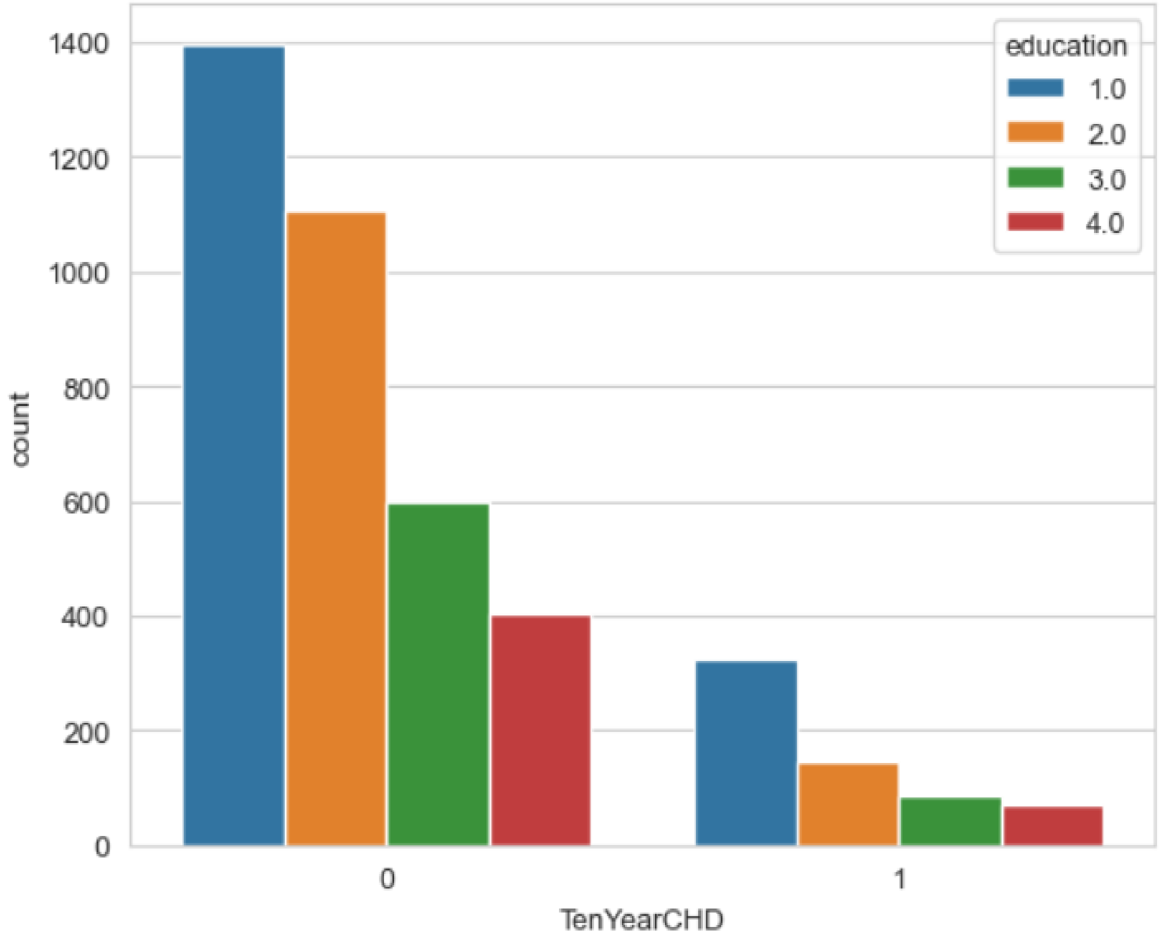


```
<Axes: >
```

```
sns.countplot(x='TenYearCHD',hue='education’,data=data)
```

**Output:**

```
<Axes: xlabel='TenYearCHD', ylabel='count'>
```

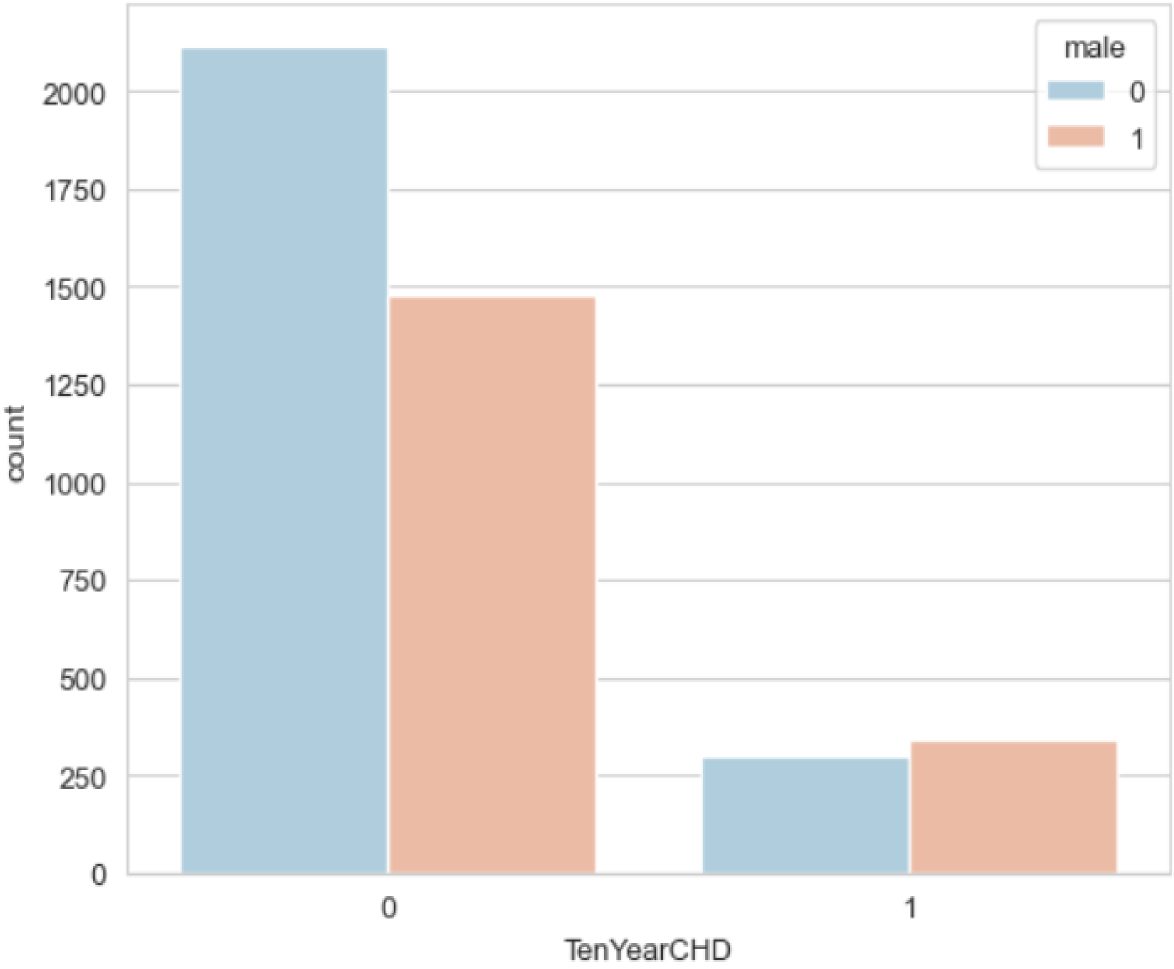


**Code:**

```
sns.countplot(x='TenYearCHD',hue='male',data=data,palette='RdBu_r')
```

**Output:**

```
<Axes: xlabel='TenYearCHD', ylabel='count'>
```

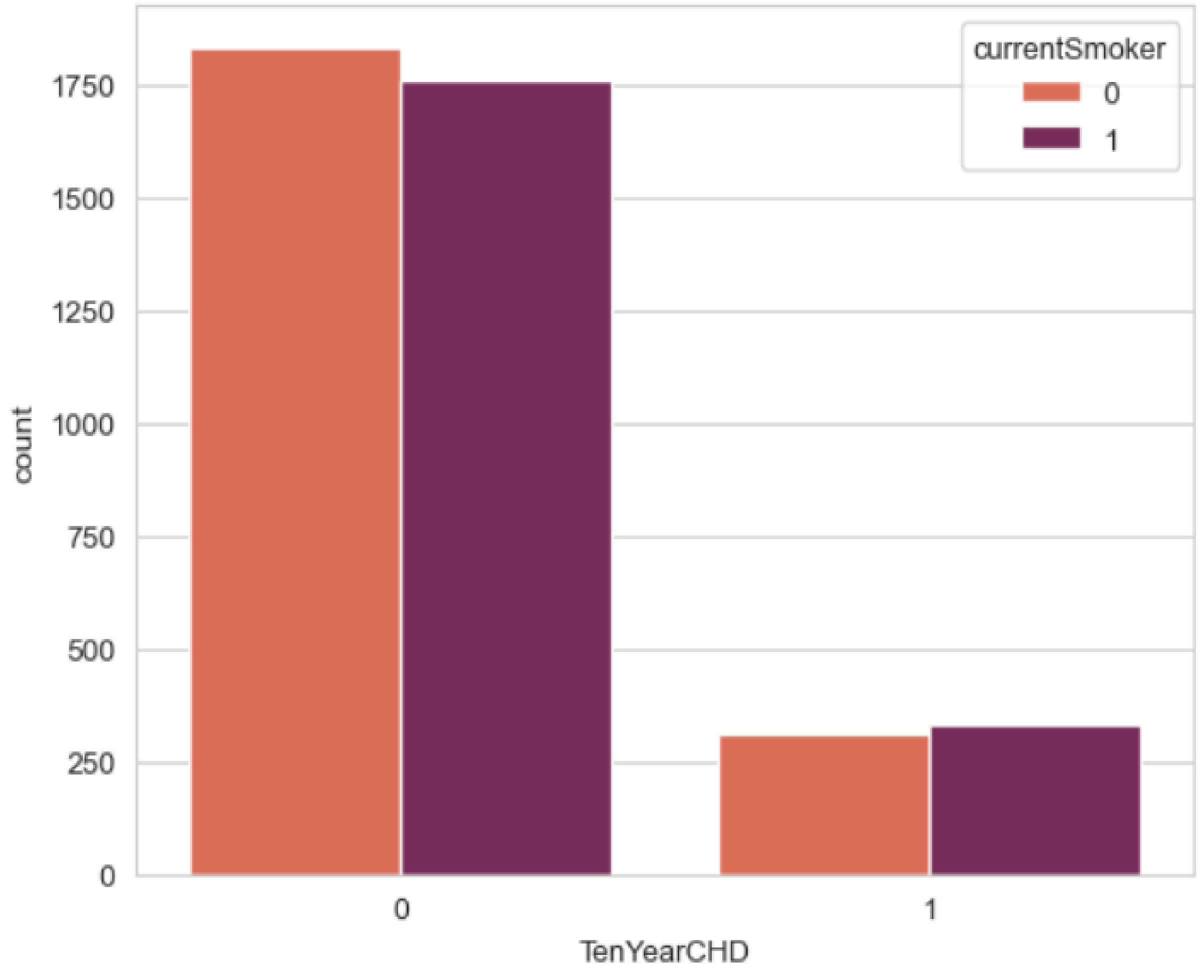


**Code:**

```
sns.countplot(x='TenYearCHD',hue='currentSmoker',data=data,palette='rocket_r')
```

**Output:**

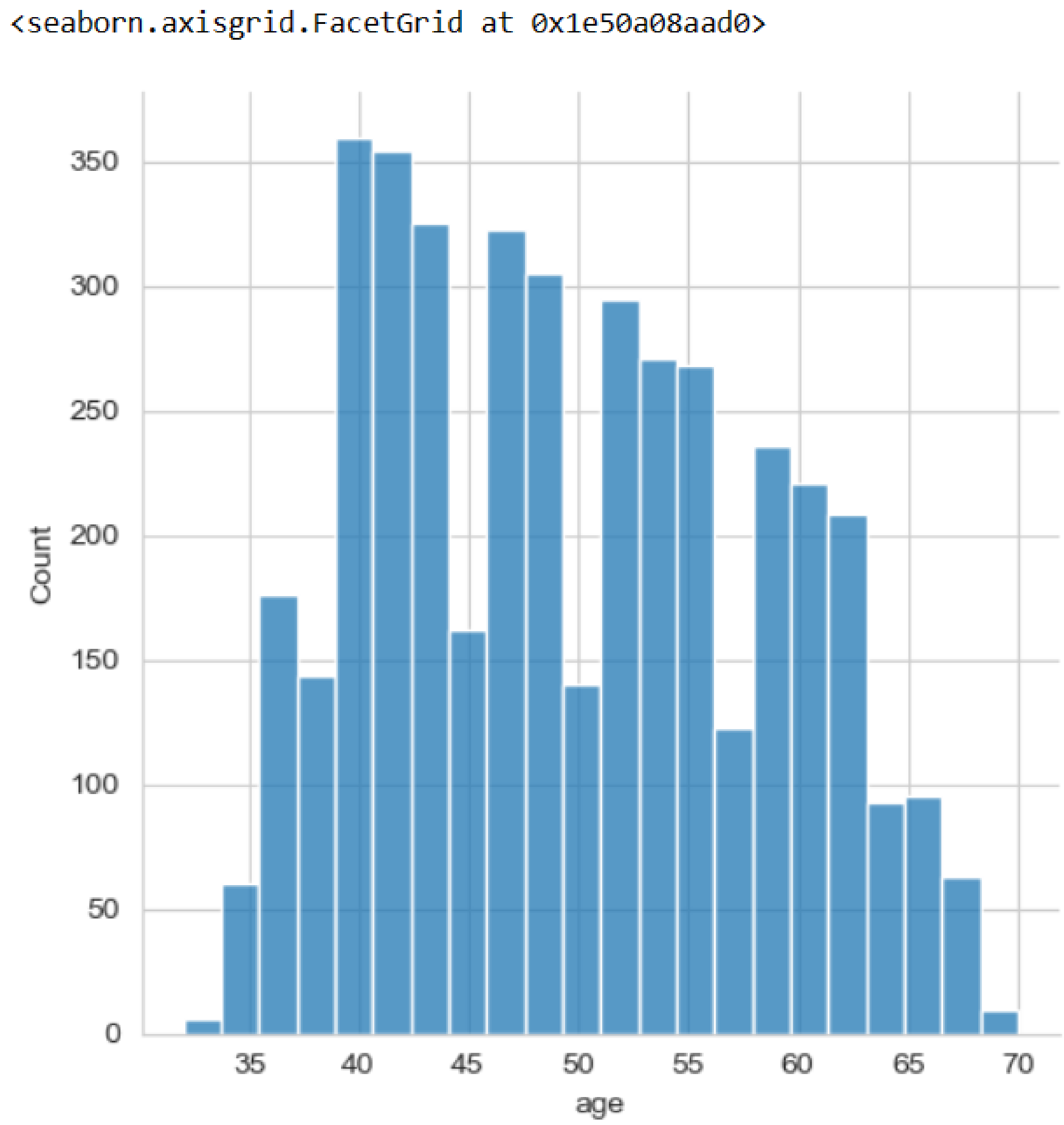
```
<Axes: xlabel='TenYearCHD', ylabel='count'>
```



**Code:**

```
sns.displot(data['age'])
```

**Output:**



```
<seaborn.axisgrid.FacetGrid at 0x1e50a08aad0>
```

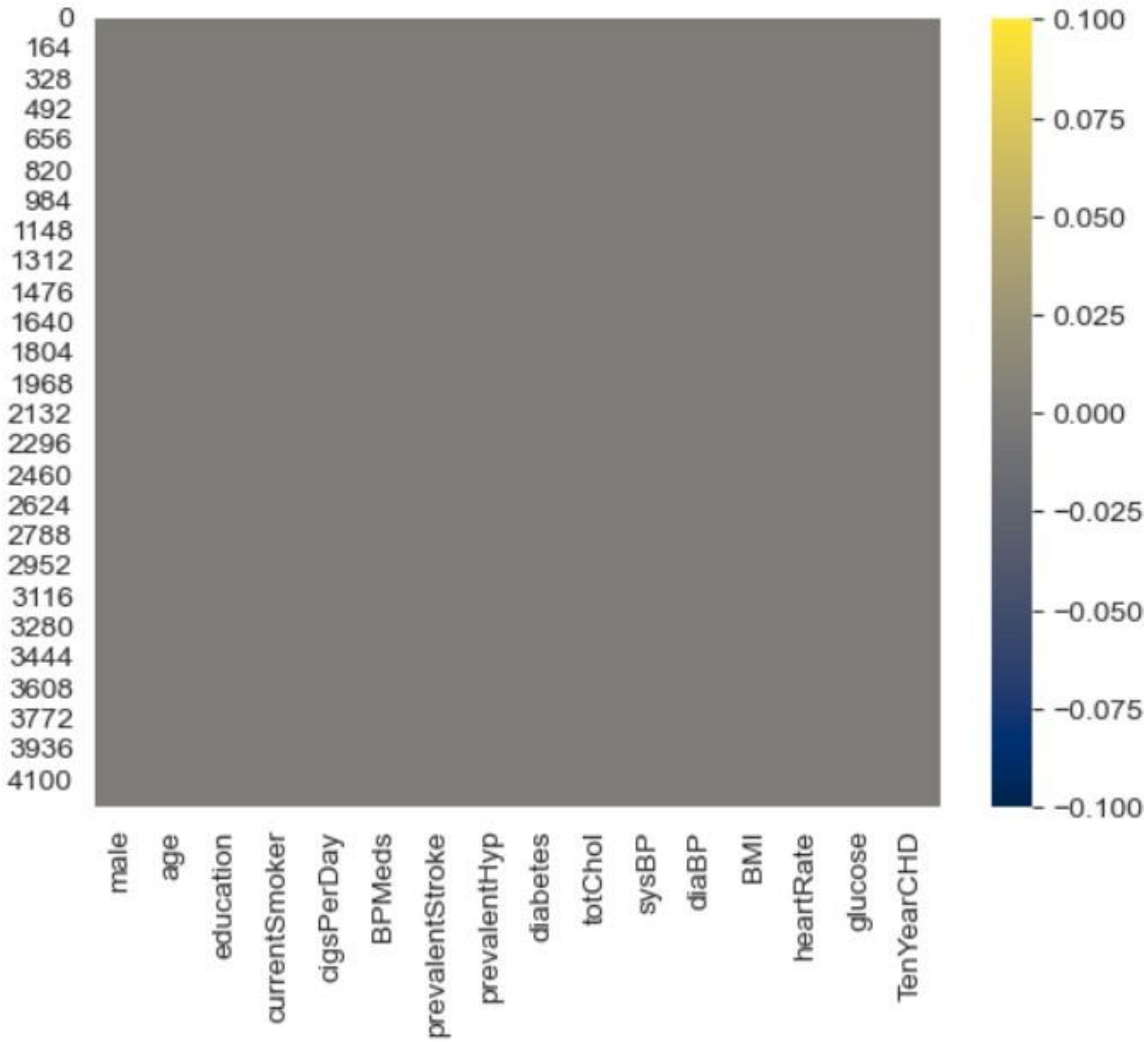
**Code:**

```
#Function
def fill_with_Data(cols):
    coloum=cols.keys()[0]
    female=cols[0]
    male=cols[1]

    if pd.isnull(female):
        if male==1:
            return math.ceil(data[data['male']==1][coloum].mean())
        elif male==0:
            return math.ceil(data[data['male']==0][coloum].mean())

    else:
        return female
```

```
#Clean Null Values
data['education'] = data[['education','male']].apply(fill_with_Data,axis=1)
data['cigsPerDay']=data[['cigsPerDay','male']].apply(fill_with_Data,axis=1)
data['BPMeds']=data[['BPMeds','male']].apply(fill_with_Data,axis=1)
data['totChol']=data[['totChol','male']].apply(fill_with_Data,axis=1)
data['BMI']=data[['BMI','male']].apply(fill_with_Data,axis=1)
data['heartRate']=data[['heartRate','male']].apply(fill_with_Data,axis=1)
data['glucose']=data[['glucose','male']].apply(fill_with_Data,axis=1)


sns.heatmap(data.isnull(),cmap='cividis')
```

**Output:**

```
<Axes: >
```



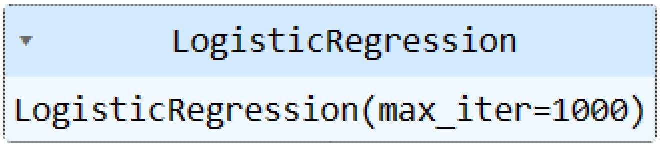
**Building a Logistic Regression Model:**

**Code:**

```
# Train Test Split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(data.drop('TenYearCHD',axis=1),
data['TenYearCHD'], test_size=0.3, random_state=101)
```

```
#Training
from sklearn.linear_model import LogisticRegression
lrmodel=LogisticRegression(solver='lbfgs',max_iter=1000)
lrmodel.fit(x_train,y_train)
```

**Output:**

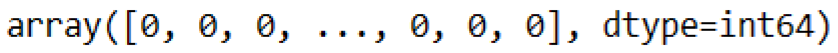


```
▾         LogisticRegression
LogisticRegression(max_iter=1000)
```

**Code:**

```
#Predicting
predictions=lrmodel.predict(x_test)
predictions
```

**Output:**



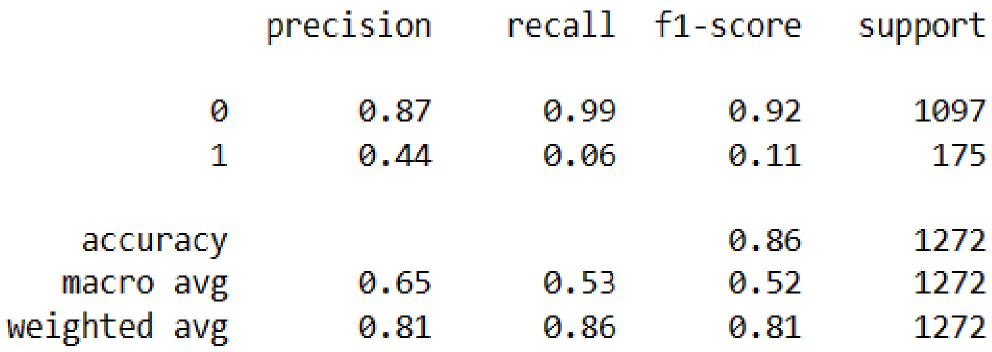
```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

**Evaluation**:

**Code:**

```
from sklearn.metrics import classification_report
print(classification_report(y_test,predictions))
```

**Output:**



```
              precision    recall  f1-score   support

           0       0.87      0.99      0.92      1097
           1       0.44      0.06      0.11       175

    accuracy                           0.86      1272
   macro avg       0.65      0.53      0.52      1272
weighted avg       0.81      0.86      0.81      1272
```