PROJECT

Cardiovascular (Heart) Disease Prediction

Framingham Heart Study Dataset

Description

The Framingham Heart Study is a long-running epidemiological study that has played a significant role in advancing our understanding of cardiovascular disease. The primary goal of the Framingham Heart Study is to identify and understand the risk factors associated with cardiovascular diseases, such as heart disease and stroke. Researchers collect data on various aspects of participants' health, including their medical history, lifestyle, diet, physical activity, and more. This data is used to identify patterns and risk factors associated with the development of heart diseases.

Data Dictionary

Sex: The gender of the participant (typically coded as 0 for female and 1 for male).

Age: The age of the participant at the time of data collection.

Education: The "education" column typically records information about the educational attainment of the study participants.

1: Less than high school education

2: High school graduate or equivalent

3: Some college or vocational training

4: College graduate

5: Postgraduate education (e.g., master's degree, doctorate)

Current Smoker: Information on whether the participant is a current smoker or non-smoker.

CigsPerDay: If the participant is Smoker, then it shows number of Cigarettes per day.

BPMeds: Information on the use of medications, such as blood pressure medication.

Prevalent Stroke: The "Prevalent Stroke" column in the dataset records whether or not each participant has had a prevalent stroke at the time of data collection. A prevalent stroke refers to the occurrence of a stroke before or at the beginning of the study period.

Values:

1: Participants with a value of 1 in this column have had a stroke before or at the beginning of the study. This indicates a history of stroke.

0: Participants with a value of 0 in this column have not had a stroke before or at the beginning of the study. This indicates the absence of a stroke history at the time of data collection.

PrevalentHyp: The "PrevalentHyp" column represents the presence or absence of prevalent hypertension (high blood pressure) among study participants at the time of data collection.

Values:

1: Participants with a value of 1 in this column have prevalent hypertension. This indicates that they have high blood pressure.

0: Participants with a value of 0 in this column do not have prevalent hypertension. This indicates that they do not have high blood pressure at the time of data collection.

Diabetes: Information on whether the participant has diabetes

TotChol: Cholesterol Levels: Typically includes measurements of total cholesterol, HDL cholesterol (high-density lipoprotein), and LDL cholesterol (low-density lipoprotein).

SysBP: The "sysBP" column contains numerical values representing the systolic blood pressure of each participant in the study. Systolic blood pressure is the higher of the two blood pressure values and is associated with the force exerted on blood vessel walls when the heart contracts.

Measurement Unit: Systolic blood pressure is typically measured in millimeters of mercury (mm Hg).

DiaBP: The "diaBP" column contains numerical values representing the diastolic blood pressure of each participant in the study. Diastolic blood pressure is the lower of the two blood pressure values and is associated with the force exerted on blood vessel walls when the heart is at rest between beats.

Measurement Unit: Diastolic blood pressure is typically measured in millimeters of mercury (mm Hg).

BMI (Body Mass Index): A measure of body weight relative to height, calculated as weight (kg) divided by height (m^2).

Heart Rate: The "heartrate" column contains numerical values representing the heart rate (number of heartbeats per minute) of each participant in the study. It provides information about the rate at which the heart is pumping blood throughout the body.

Measurement Unit: Heart rate is typically measured in beats per minute (bpm).

Glucose: The "Glucose" column contains numerical values representing blood glucose levels, typically measured in milligrams per deciliter (mg/dL), for each participant in the study. Blood glucose levels indicate the concentration of sugar (glucose) in the blood.

Measurement Unit: Blood glucose is typically measured in milligrams per deciliter (mg/dL).

TenYearCHD: The "TenYearCHD" column records binary values indicating whether each participant is at risk of developing coronary heart disease (CHD) within the next ten years. It serves as an outcome variable in the study, with "1" typically indicating a higher risk and "0" indicating a lower risk.

Values:

1: Participants with a value of 1 in this column are considered at risk of developing CHD within the next ten years.

0: Participants with a value of 0 in this column are considered not at risk of developing CHD within the next ten years.

Business Problem Statement

1. How can we develop an accurate and scalable heart disease prediction model using the Framingham Heart Study data to proactively identify individuals at high risk of developing coronary heart disease (CHD) and implement targeted interventions for prevention and early management?
2. Which features in the dataset have the strongest correlations with the presence or absence of heart disease?
3. What features (biomarkers, clinical data) are the most important in predicting heart disease?
4. Are there significant differences in heart disease risk factors and prediction between males and females?
5. Is it possible to detect heart disease at an early stage, and what are the key indicators?

More Problem Statements will be added as we proceed further with this Project.

Tools that will be used in Project.

Excel: Data Understanding.
SQL: Data Cleaning & Data Preprocessing.
Python: Data Analysis & Machine Learning.
Power BI: Data Visualization.

First, we started with importing the Packages

Cardiovascular (Heart) Disease Prediction

Import Library Packages

In [2]:
```
# Importing the Packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy as sc
import sklearn as sk
import seaborn as sns
import missingno as msno
```

Then We imported the dataset

Importing Dataset

3]:
```
#Importing the Raw Dataset and Displaying the Dataset
data = pd.read_csv(r"C:\Users\DELL\Desktop\HealthCare Project\framingham.csv")
data.head(10)
```

3]:

	Male	Age	Education	CurrentSmoker	CigsPerDay	BPMeds	PrevalentStroke	PrevalentHyp	Diabetes	TotChol	SysBP	DiaBP	BMI	HeartRate	Glucose	T
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	
5	0	43	2.0	0	0.0	0.0	0	1	0	228.0	180.0	110.0	30.30	77.0	99.0	
6	0	63	1.0	0	0.0	0.0	0	0	0	205.0	138.0	71.0	33.11	60.0	85.0	
7	0	45	2.0	1	20.0	0.0	0	0	0	313.0	100.0	71.0	21.68	79.0	78.0	
8	1	52	1.0	0	0.0	0.0	0	1	0	260.0	141.5	89.0	26.36	76.0	79.0	
9	1	43	1.0	1	30.0	0.0	0	1	0	225.0	162.0	107.0	23.61	93.0	88.0	

Then to understand the Dataset we viewed the dimension of the dataset and its variables.

Understanding the Dataset

```
# Dimension Of the Dataset
data.shape
```

```
(4238, 16)
```

```
# The DataTypes Of the Columns of The Dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4238 entries, 0 to 4237
Data columns (total 16 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Male             4238 non-null   int64
 1   Age              4238 non-null   int64
 2   Education        4133 non-null   float64
 3   CurrentSmoker    4238 non-null   int64
 4   CigsPerDay       4209 non-null   float64
 5   BPMeds           4185 non-null   float64
 6   PrevalentStroke  4238 non-null   int64
 7   PrevalentHyp     4238 non-null   int64
 8   Diabetes         4238 non-null   int64
 9   TotChol          4188 non-null   float64
 10  SysBP            4238 non-null   float64
 11  DiaBP            4238 non-null   float64
 12  BMI              4219 non-null   float64
```

Then We started with the data cleaning and data Preprocessing steps. We checked the missing values first.

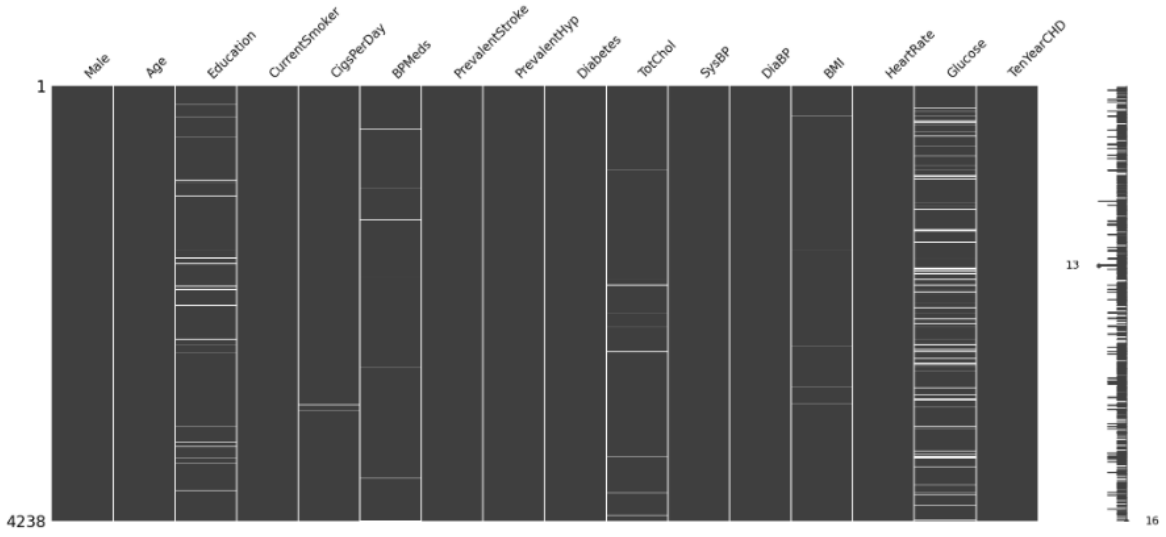
Data Cleaning And Data PreProcessing

```
[6]: ##Finding The Missing Values in the Datset
     data.isnull().sum()
```

```
:[6]: 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
```

Then we visualized the missing values in the dataset.

```
[7]: #Visualizing the Missing Values in the Dataset
     msno.matrix(data)
     plt.show()
```



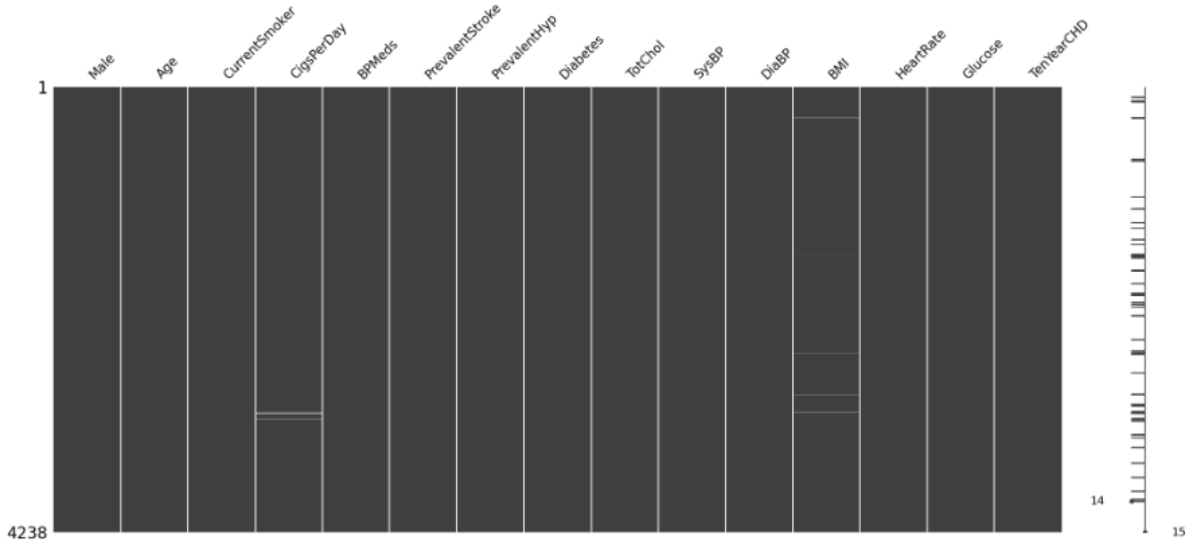
Then We dealt with the missing values in the dataset

In [8]:
```
## Education feature is not required as its not predicting the Ten Year CHD
## Target is Ten Year CHD (0 or 1)
data.drop('Education', axis=1, inplace=True)
```

Dealing with the Missing Records in the Dataset

In [9]:
```
## To fill the Missing values in the Gulcose column,TotChol column and BPMeds column we can use mean value of the Glucose Column
data['Glucose']=data['Glucose'].fillna(data['Glucose'].mean())
data['TotChol']=data['TotChol'].fillna(data['TotChol'].mean())
data['BPMeds']=data['BPMeds'].fillna(data['BPMeds'].mean())
```

10]:
```
#Visualizing the Missing Values in the Dataset
msno.matrix(data)
plt.show()
```



11]:
```
## For the rest of the missing values we drop the rows containing missing values
data = data.dropna()
```

12]:
```
# The Percentage of missing values in the Dataset
(data.isnull().sum())*100/len(data)
```

12]:
```
Male               0.0
Age                0.0
CurrentSmoker      0.0
CigsPerDay         0.0
BPMeds             0.0
PrevalentStroke    0.0
PrevalentHyp       0.0
Diabetes           0.0
TotChol            0.0
SysBP              0.0
DiaBP              0.0
BMI                0.0
HeartRate          0.0
Glucose            0.0
TenYearCHD         0.0
dtype: float64
```

Then We checked the Duplicated values in the dataset

Finding the Duplicates in the dataset

```
## Finding the Duplicated rows in the Dataset
data.duplicated().sum()
```

```
0
```

There are No Duplicated Rows in the Dataset

There was no duplicated values.

Then We started with Exploratory Data Analysis (EDA) of the Dataset.

Exploratory Data Analysis (EDA)

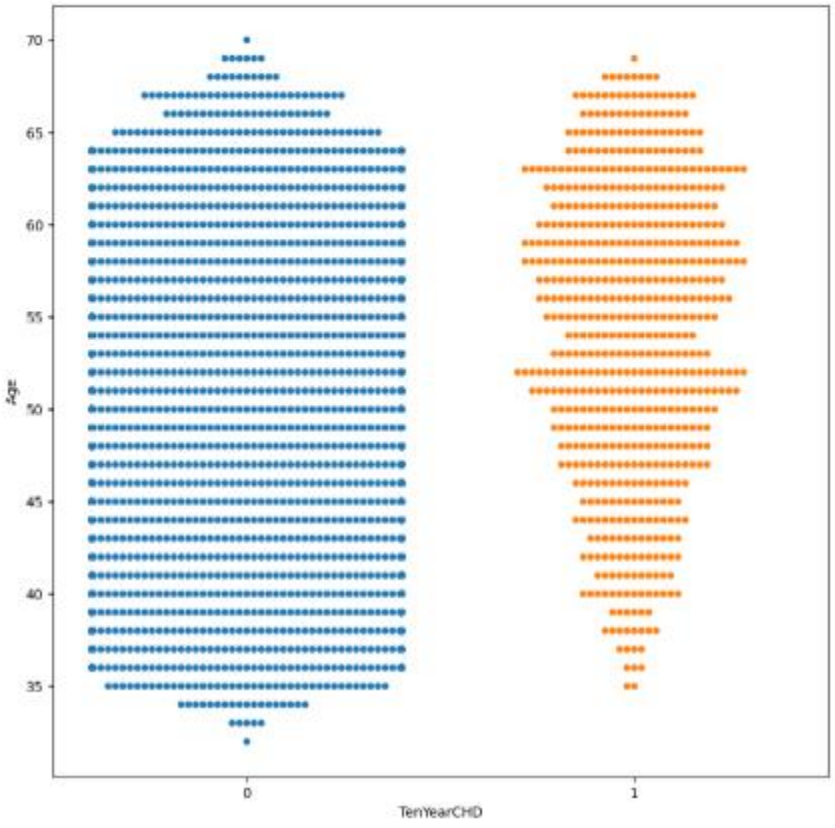
]:
```
# Dimension Of the Dataset
data.shape
```

]: (4189, 15)

16]:
```
# Visualizing the number of heart disease patients with respect to age
# age vs CHD
plt.figure(figsize=(10,10))
sns.swarmplot(x='TenYearCHD', y='Age', data=data)
```

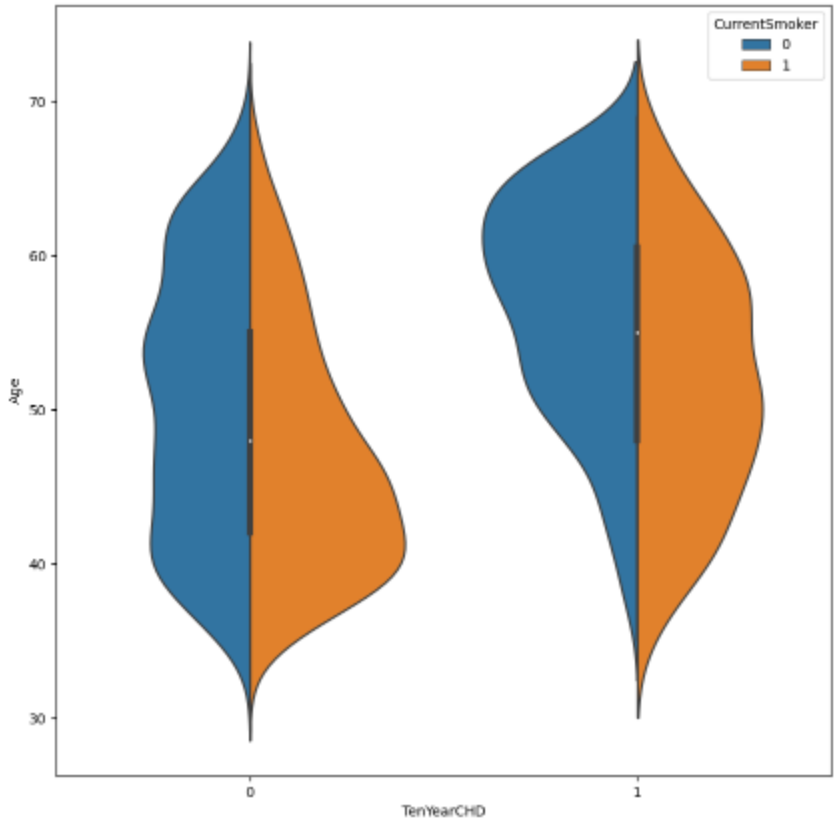
```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 60.2% of the points cannot be pla
ay want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

16]: <AxesSubplot:xlabel='TenYearCHD', ylabel='Age'>



```
# Visualizing the number of heart disease patients with respect to current Smoking habit
plt.figure(figsize=(10,10))
sns.violinplot(x='TenYearCHD', y='Age', data= data, hue='CurrentSmoker', split=True)
```

```
<AxesSubplot:xlabel='TenYearCHD', ylabel='Age'>
```

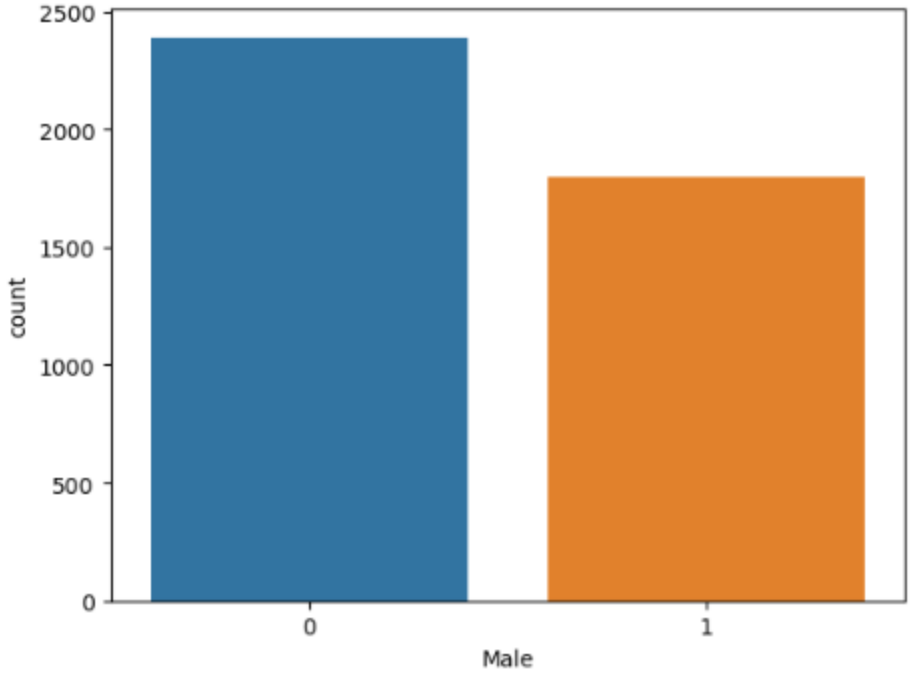


From this violinplot, we see that most of smokers having no risk of CHD are in age around 40 years

But most of non-smokers having risk are in age around 65-70 years Also most smokers having risk are in age around 50 years

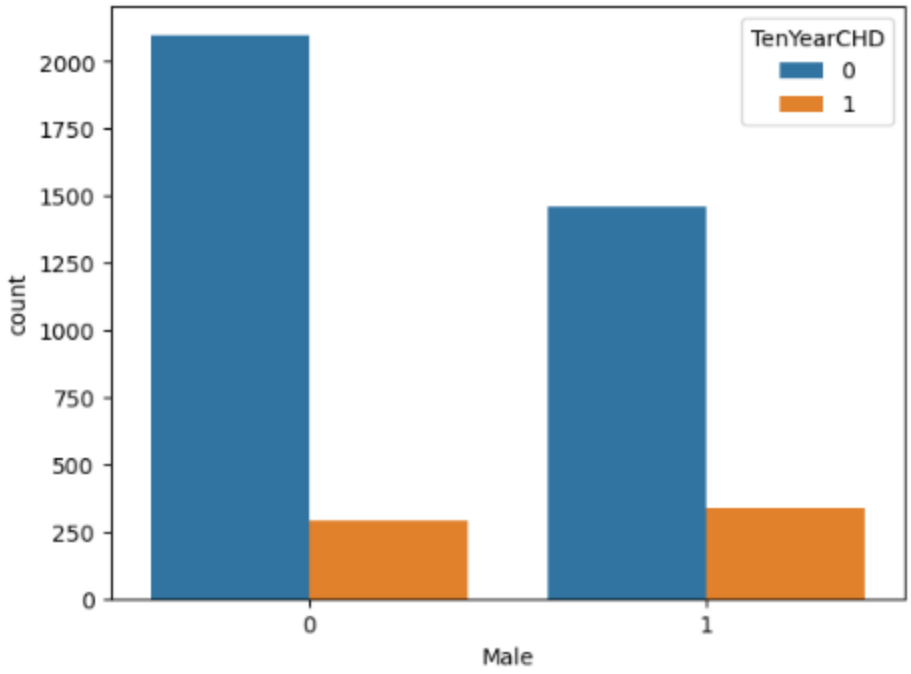
```
# The number of males and females in the dataset
# male and female countplot
sns.countplot(x=data['Male'])
```

```
<AxesSubplot:xlabel='Male', ylabel='count'>
```



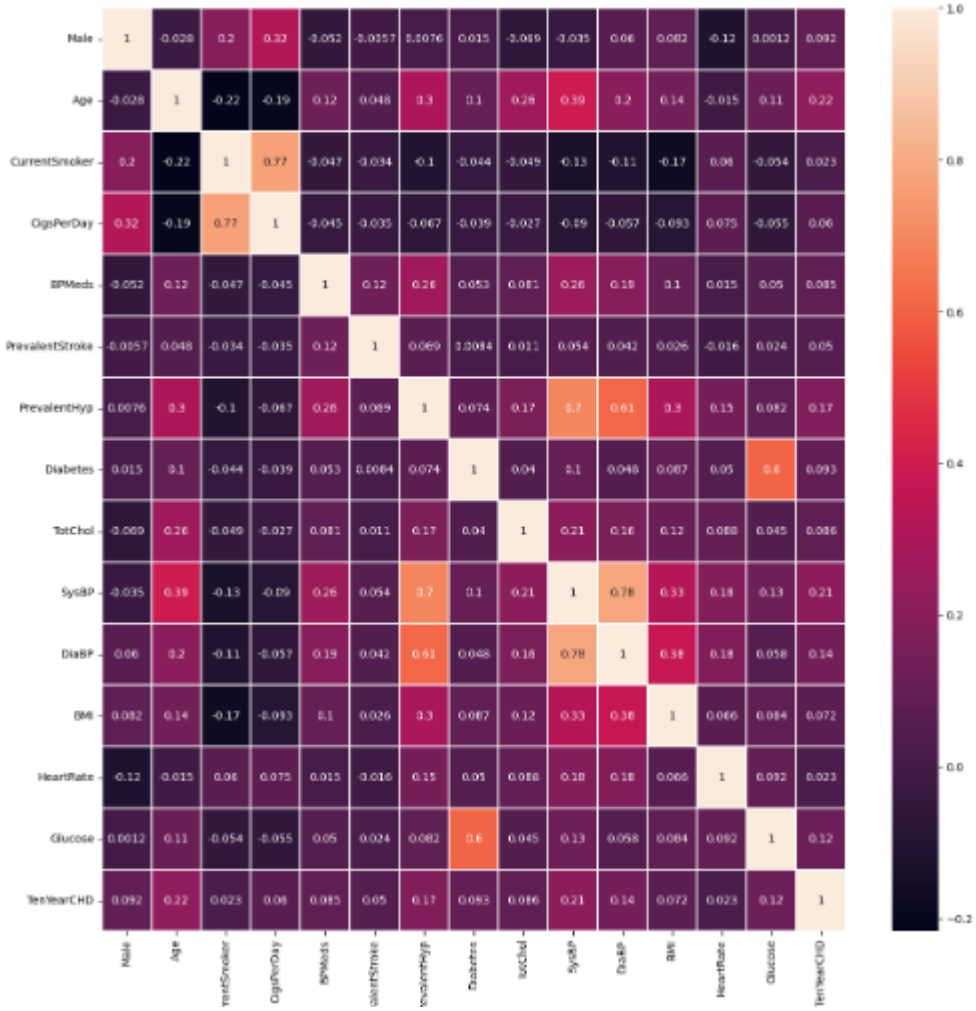
.9]:
```
## Visualizing Whether male and female having Heart disease or not
sns.countplot(x=data['Male'], hue=data['TenYearCHD'])
```

.9]: <AxesSubplot:xlabel='Male', ylabel='count'>



28]:
```
## Visualizing the Corelations between the attributes of the dataset
plt.figure(figsize=(15,15))
sns.heatmap(data.corr(), annot=True, linewidths=0.1)
```

28]: <AxesSubplot:>

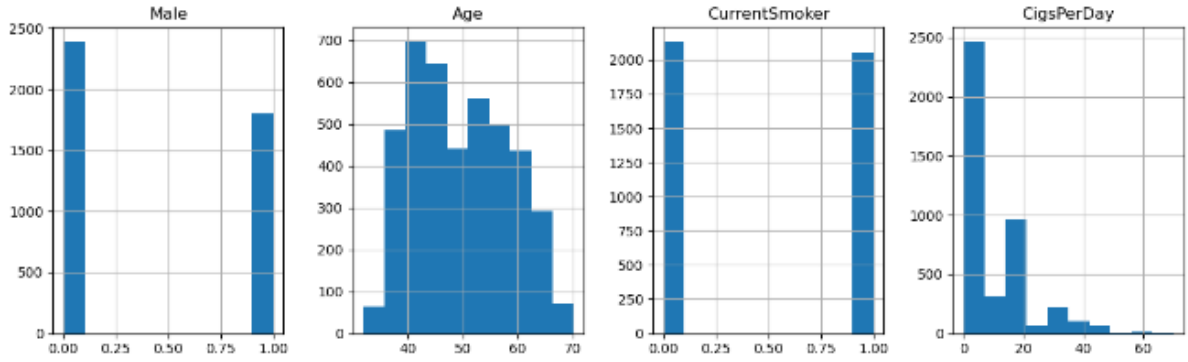


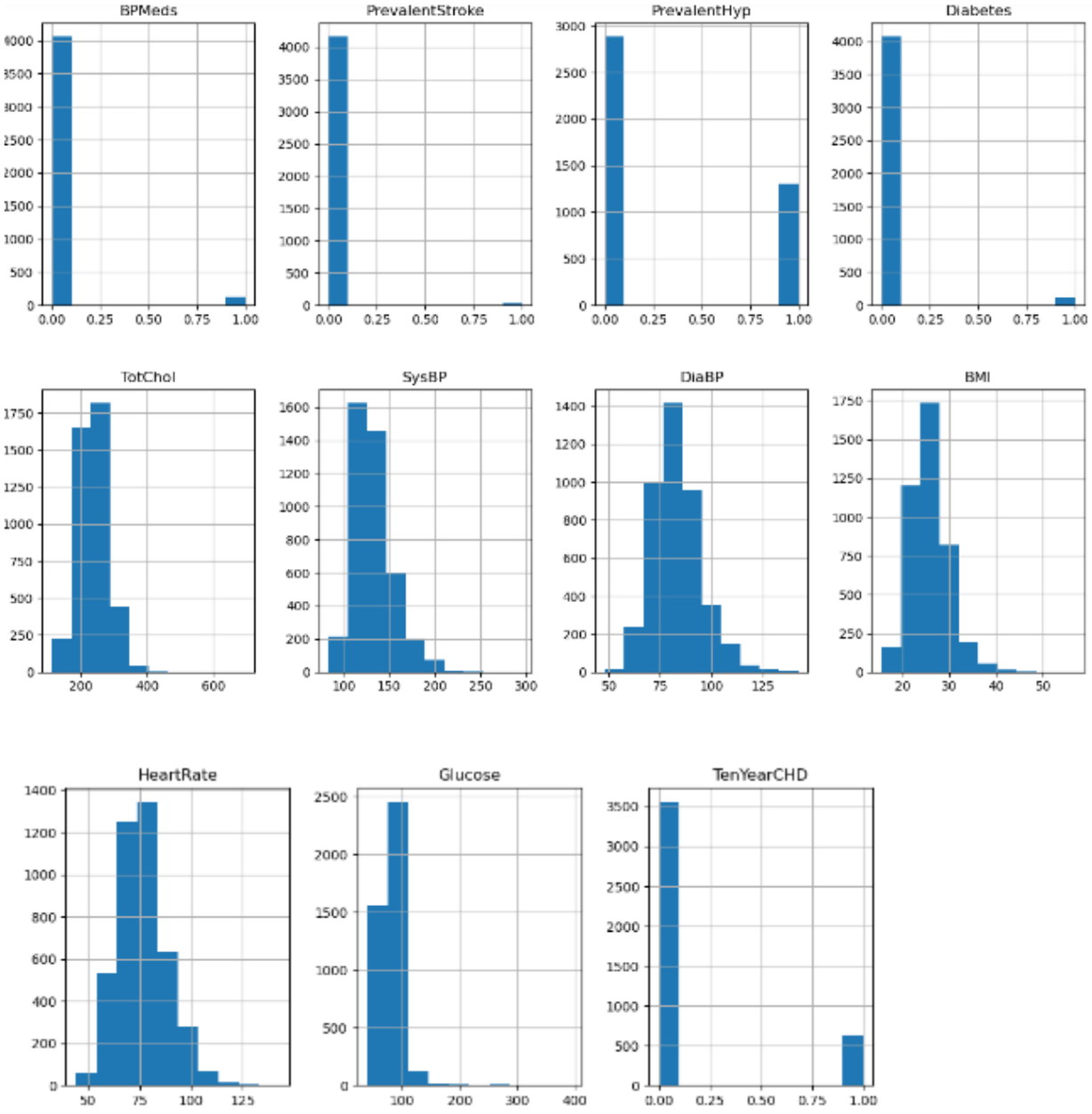
1]:
```
# Visualizing the Distributions of records in the columns using histograms

fig = plt.figure(figsize = (15,20))
ax = fig.gca()
data.hist(ax = ax)
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_9628\3602945128.py:4: UserWarning: To output multiple subplots, the figure containin
g the passed axes is being cleared.
  data.hist(ax = ax)
```

1]:
```
array([[<AxesSubplot:title={'center':'Male'}>,
        <AxesSubplot:title={'center':'Age'}>,
        <AxesSubplot:title={'center':'CurrentSmoker'}>,
        <AxesSubplot:title={'center':'CigsPerDay'}>],
       [<AxesSubplot:title={'center':'BPMeds'}>,
        <AxesSubplot:title={'center':'PrevalentStroke'}>,
        <AxesSubplot:title={'center':'PrevalentHyp'}>,
        <AxesSubplot:title={'center':'Diabetes'}>],
       [<AxesSubplot:title={'center':'TotChol'}>,
        <AxesSubplot:title={'center':'SysBP'}>,
        <AxesSubplot:title={'center':'DiaBP'}>,
        <AxesSubplot:title={'center':'BMI'}>],
       [<AxesSubplot:title={'center':'HeartRate'}>,
        <AxesSubplot:title={'center':'Glucose'}>,
        <AxesSubplot:title={'center':'TenYearCHD'}>, <AxesSubplot:>]],
      dtype=object)
```




BPMeds
PrevalentStroke
PrevalentHyp
Diabetes
TotChol
SysBP
DiaBP
BMI
HeartRate
Glucose
TenYearCHD

Thank You