How to choose domain?

Domain should be chosen on the basis of our interest and understanding. Since all the further procedures would depend on what domain we choose, it is important that we have some prior knowledge of the domain, the methodologies followed, the developments and the need of the our domain.For this we can try working on data from different domains and narrow down to one for our future work. Along with this we need to assess the resources that are available- datasets, tools, softwares, etc and then finalise our domain and project plan.

How to choose the dataset?

Once we have finalised the domain, it is important to choose the right dataset. The source needs to be reliable and the dataset should contain as much information/variables as necessary for the analysis. We need to consider all the factors that make a good dataset- less missing values, reliability of source, more information. We can even collect data on our own based on the requirement, money and time we have. For example, I have taken a Medical Dataset here to predict the possibility of heart failure.

In [1]:

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

In [4]:

```
df=pd.read_csv('D:/Downloads/heart.csv')
df.head()
```

Out[4]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	Exercis
0	40	M	ATA	140	289	0	Normal	172	
1	49	F	NAP	160	180	0	Normal	156	
2	37	M	ATA	130	283	0	ST	98	
3	48	F	ASY	138	214	0	Normal	108	
4	54	M	NAP	150	195	0	Normal	122	

How to understand data?

Data can be understood in various ways. We can start by understanding our variables, their type and their descriptive statistics. We can also try and understand if their are any codes which have meanings/have been encoded. We can also understand the data by checking missing values, outliers, etc and see if they carry a meaning. Then, we can start visualising our data to understand its distribution, the relations between variables, the affect on our target variable,etc. For example:

• Univariate Analysis

o Understanding mean, median, mode and skewness of data

o Plots like boxplot, count plot etc

• Bi Variate Anaysis

o correlations

o Plots like boxplot, line plot etc

• Multivariate Analysis, etc

Here, we have a basic knowledge of the variables regarding their units and meaning. It is as follows:

Attribute Information

Age: age of the patient [years]

Sex: sex of the patient [M: Male, F: Female]

ChestPainType: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]

RestingBP: resting blood pressure [mm Hg]

Cholesterol: serum cholesterol [mm/dl]

FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]

RestingECG: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), LVH: showing probable or definite left ventricular hypertrophy by Estes' criteria]

MaxHR: maximum heart rate achieved [Numeric value between 60 and 202]

ExerciseAngina: exercise-induced angina [Y: Yes, N: No]

Oldpeak: oldpeak = ST [Numeric value measured in depression]

ST_Slope: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]

HeartDisease: output class [1: heart disease, 0: Normal]

In [5]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             918 non-null    int64
 1   Sex             918 non-null    object
 2   ChestPainType   918 non-null    object
 3   RestingBP       918 non-null    int64
 4   Cholesterol     918 non-null    int64
 5   FastingBS       918 non-null    int64
 6   RestingECG      918 non-null    object
 7   MaxHR           918 non-null    int64
 8   ExerciseAngina  918 non-null    object
 9   Oldpeak         918 non-null    float64
 10  ST_Slope        918 non-null    object
 11  HeartDisease    918 non-null    int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.2+ KB
```

We can see that there are no null values. We can also see the various data types of the variables.

In [8]:

```
1 df.nunique()
```

Out[8]:

```
Age               50
Sex                2
ChestPainType      4
RestingBP         67
Cholesterol      222
FastingBS          2
RestingECG         3
MaxHR            119
ExerciseAngina     2
Oldpeak           53
ST_Slope           3
HeartDisease       2
dtype: int64
```

Here, we can see that there are certain categorical values- the ones with unique values such as 2,3,4. So we can have a basic idea about our variables.

In [10]:

```
1 df.describe(include='all')
```

Out[10]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	
count	918.000000	918	918	918.000000	918.000000	918.000000	918	91
unique	NaN	2	4	NaN	NaN	NaN	3	
top	NaN	M	ASY	NaN	NaN	NaN	Normal	
freq	NaN	725	496	NaN	NaN	NaN	552	
mean	53.510893	NaN	NaN	132.396514	198.799564	0.233115	NaN	13
std	9.432617	NaN	NaN	18.514154	109.384145	0.423046	NaN	2
min	28.000000	NaN	NaN	0.000000	0.000000	0.000000	NaN	6
25%	47.000000	NaN	NaN	120.000000	173.250000	0.000000	NaN	12
50%	54.000000	NaN	NaN	130.000000	223.000000	0.000000	NaN	13
75%	60.000000	NaN	NaN	140.000000	267.000000	0.000000	NaN	15
max	77.000000	NaN	NaN	200.000000	603.000000	1.000000	NaN	20

To get a statistical summary of our variables. The categorical variables would not have a mean, median ,etc. This would give us an idea about the average values, quartiles, range, etc.

In [11]:

```
1 df['Sex'].value_counts()
```

Out[11]:

```
M    725
F    193
Name: Sex, dtype: int64
```

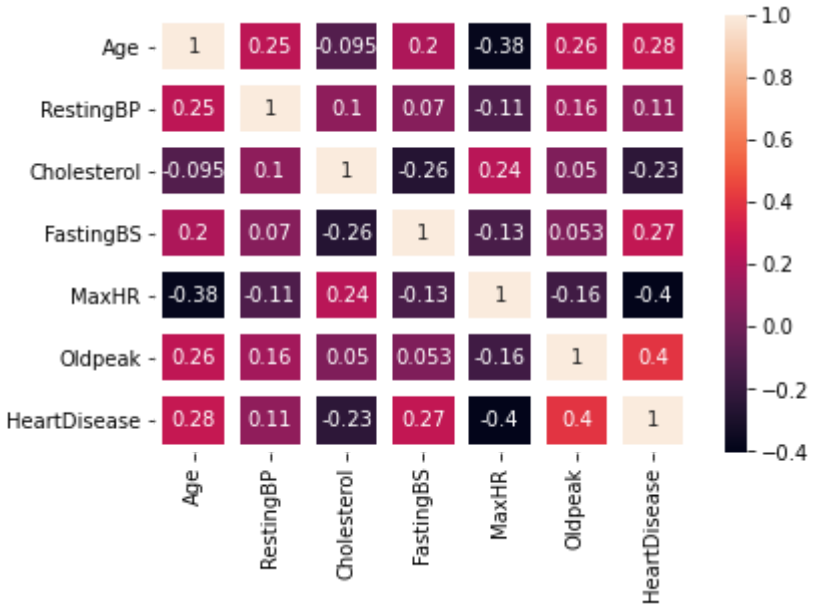
We can see that our data is male dominated and hence might produce biased results.

In [13]:

```
corr = df.corr()
sns.heatmap(corr, annot = True, linewidth=7)
```

Out[13]:

<AxesSubplot:>



We try to understand how variables are correlated with each other and with our target variable as well

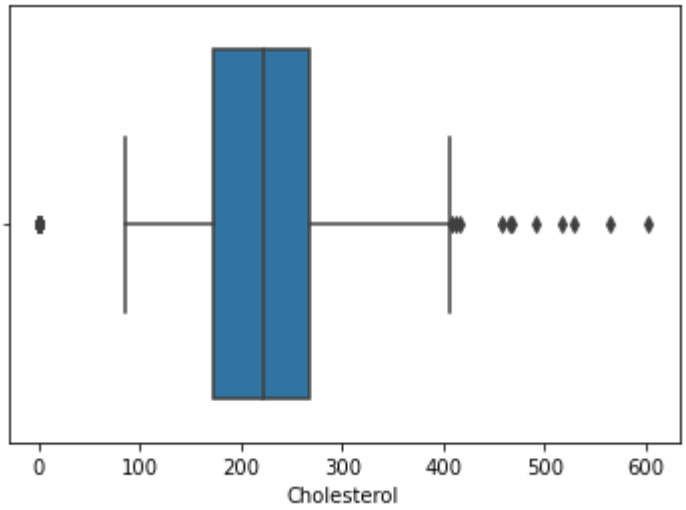
In [14]:

```
1 sns.boxplot(df["Cholesterol"])
```

```
D:\ANACONDA\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[14]:

```
<AxesSubplot:xlabel='Cholesterol'>
```



We can see that their our values which are outliers in the "Cholestrol" variable and need to be dealt with properly.

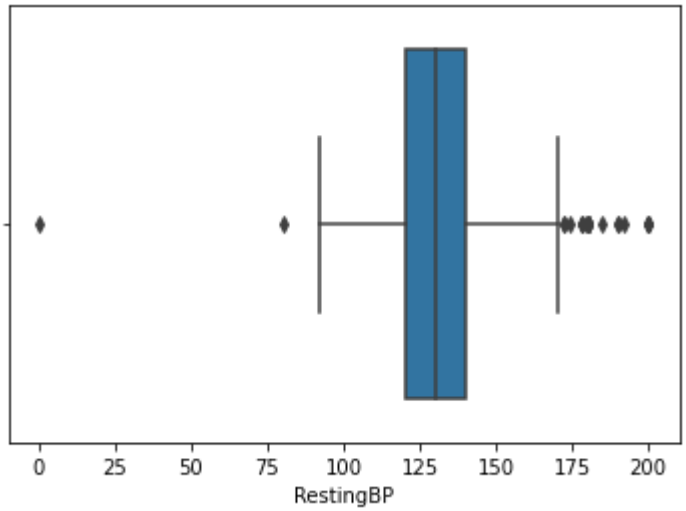
In [15]:

```
1 sns.boxplot(df["RestingBP"])
```

```
D:\ANACONDA\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass
the following variable as a keyword arg: x. From version 0.12, the only vali
d positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[15]:

```
<AxesSubplot:xlabel='RestingBP'>
```



Even "Resting BP" has outliers and we can remove the ones which are not logically or medically possible.

How to identify the proper learning method (Supervised (classification/prediction), Unsupervised learning [Clustering/Association Rule Mining) and Reinforcement Learning (optimization/strategy/ agent based learning]

We can make use following steps to identify right learning method :

Categorize the Problem

a. Categorize by the input:

If it is a labeled data, it’s a supervised learning problem. If it is not labelled, it is unsupervised learning. If it needs to train itself again and again to learn, it is reinforcement learning.

b. Categorize by output:

If the output of the model is a number, it’s a regression problem.

If it is categorical, we can use Logistic Regression.

If we have to put it into certain categories, it is classification problem.

How to identify proper model [SVM/NaiveBayes/ K-means etc.]

Finding the available algorithms

> The accuracy and complexity of the model.
>
> How long does it take to build, train, and test the model?
>
> Does the model meet the business goal?

Following are the consideration for choosing right ML algorithms:

a. Size of training data

b. Interpretability of data

c. Speed and Training time

d. Types of learning

e. Comparing the performance of model

In [ ]:

```
1
```