Project: Health Care-Prediction of Risk of a Heart Attack

Problem statement: Cardiovascular diseases are the leading cause of death globally. It is therefore necessary to identify the causes and develop a system to predict heart attacks in an effective manner. The data below has the information about the factors that might have an impact on cardiovascular health.

Dataset description:

Variable	Description
Age	Age in years
Sex	1 = male; 0 = female
cp	Chest pain type
trestbps	Resting blood pressure (in mm Hg on admission to the hospital)
chol	Serum cholesterol in mg/dl
fbs	Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
restecg	Resting electrocardiographic results
thalach	Maximum heart rate achieved
exang	Exercise induced angina (1 = yes; 0 = no)
oldpeak	ST depression induced by exercise relative to rest
slope	Slope of the peak exercise ST segment
ca	Number of major vessels (0-3) colored by fluoroscopy
thal	3 = normal; 6 = fixed defect; 7 = reversible defect
Target	1 or 0

Task to be performed:

1. Preliminary analysis:

    ```
    a. Perform preliminary data inspection and report the findings on the structure
    of the data, missing values, duplicates, etc.
    b. Based on these findings, remove duplicates (if any) and treat missing values
    using an appropriate strategy.
    ```

2. Prepare a report about the data explaining the distribution of the disease and the related factors using the steps listed below:

a. Get a preliminary statistical summary of the data and explore the measures of central tendencies and spread of the data.
b. Identify the data variables which are categorical and describe and explore these variables using the appropriate tools, such as count plot.
c. Study the occurrence of CVD across the Age category.
d. Study the composition of all patients with respect to the Sex category.
e. Study if one can detect heart attacks based on anomalies in the resting blood pressure (trestbps) of a patient.
f. Describe the relationship between cholesterol levels and a target variable.
g. State what relationship exists between peak exercising and the occurrence of a heart attack.
h. Check if thalassemia is a major cause of CVD.
i. List how the other factors determine the occurrence of CVD.
j. Use a pair plot to understand the relationship between all the given variables.

3. Build a baseline model to predict the risk of a heart attack using a logistic regression and random forest and explore the results while using correlation analysis and logistic regression (leveraging standard error and p-values from statsmodels) for feature selection.

In [1]:

```
import numpy as np
import pandas as pd

import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
hcare = pd.read_excel("1645792390_cep1_dataset.xlsx")
```

In [3]:

```
hcare.head()
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

In [4]:

```
hcare.tail()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

In [5]:

```
hcare.shape
```

Out[5]:

```
(303, 14)
```

In [6]:

```
hcare.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  target    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

In [7]:

```
hcare.dtypes
```

Out[7]:

```
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

In [8]:

```
# Checking for missing values
hcare.isnull().sum(axis = 0)
```

Out[8]:

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

In [9]:

```
hcare.describe()
```

Out[9]:

	age	sex	cp	trestbps	chol	fbs	restecg	t
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.(
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.(
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.9
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.(
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.5
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.(
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.(
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.(

We can see that the scale of each feature column is different and varied.

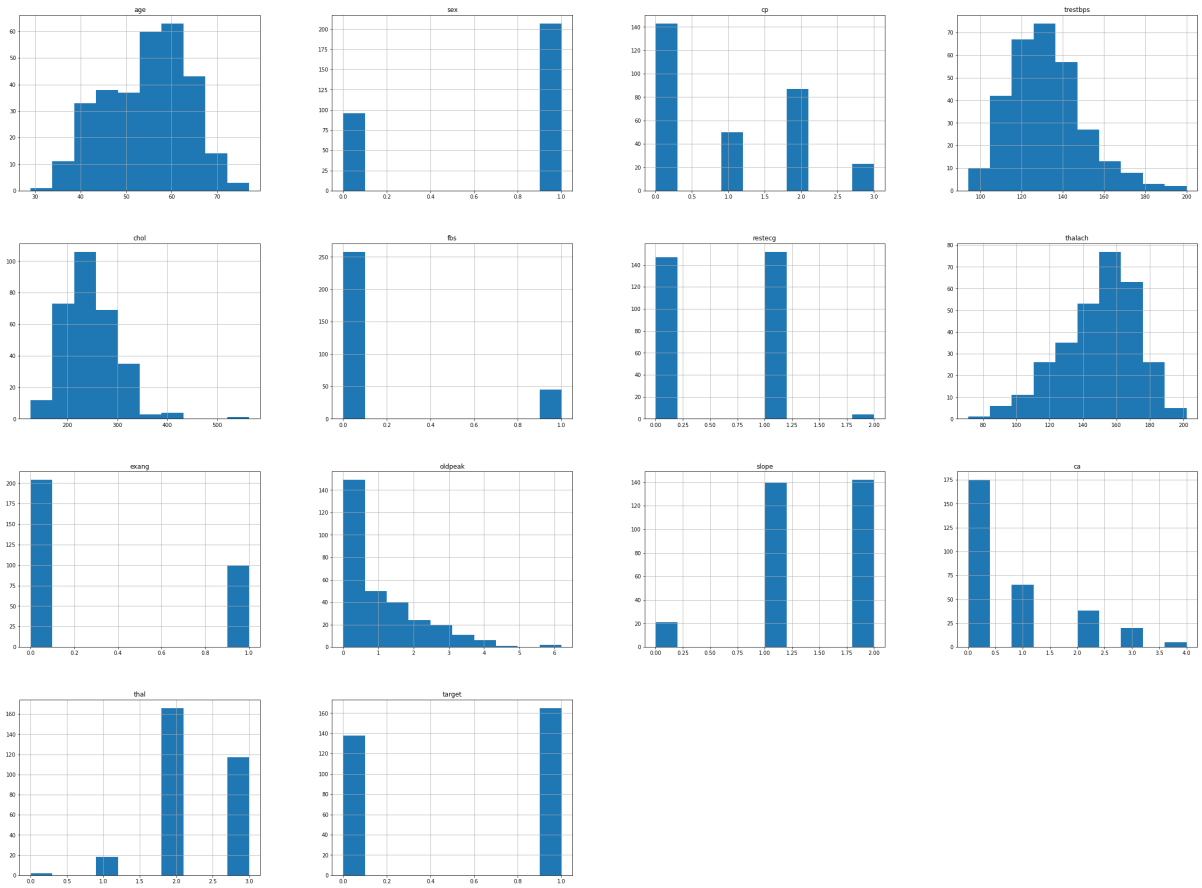
In [10]:

```
# For visualizations
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
import seaborn as sns
```

In [11]:

```
# Histogram of the Heart Dataset

fig = plt.figure(figsize = (40,30))
hcare.hist(ax = fig.gca());
```



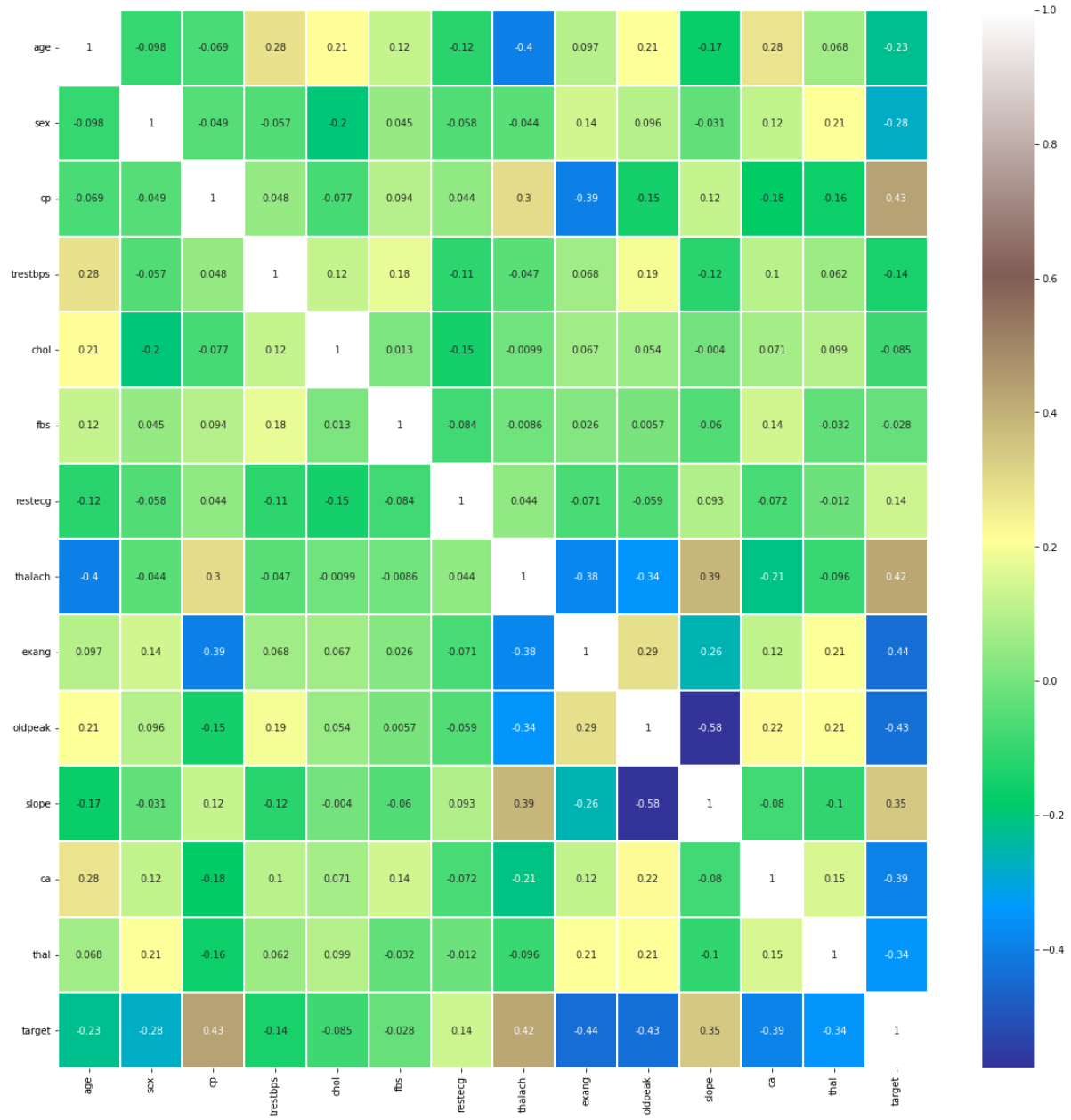
From the above histogram plots, we can see that the features are skewed and not normally distributed. Also, the scales are different between one and another.

Understanding the Data

Let us observe the creelation between different features with help of a heat mat.

In [12]:

```
# Creating a correlation heatmap
sns.heatmap(hcare.corr(),annot=True, cmap='terrain', linewidths=0.1)
fig=plt.gcf()
fig.set_size_inches(20,20)
plt.show()
```



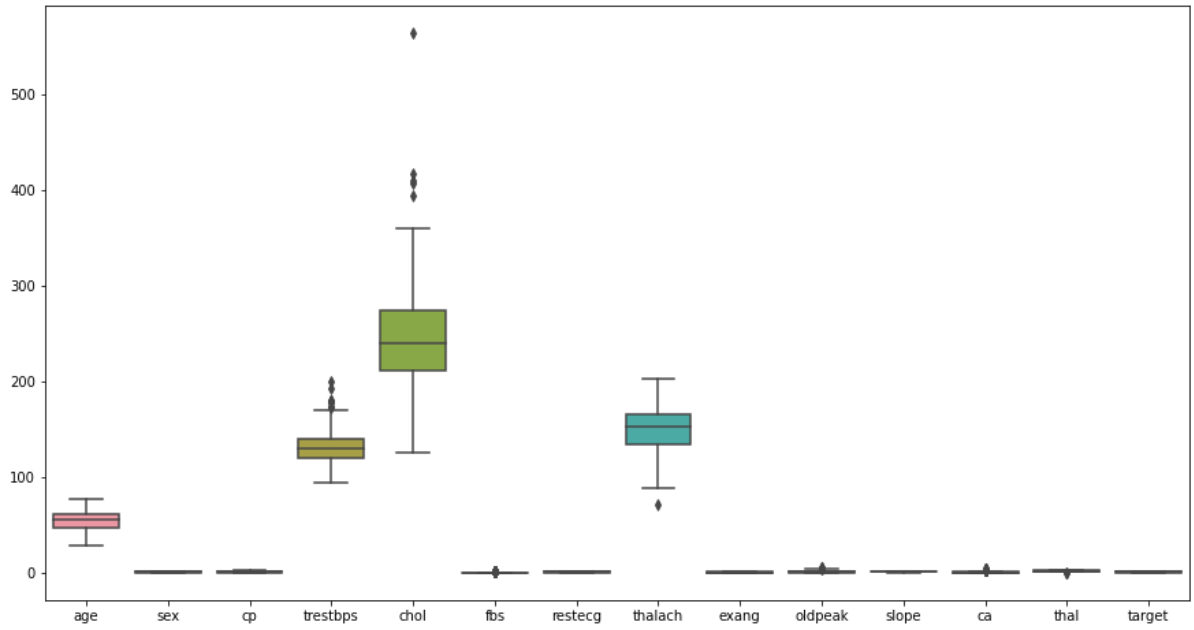
From the above HeatMap, we can see that cp and thalach are the features with highest positive correlation whereas exang, oldpeak and ca are negatively correlated.While other features do not hold much correlation with the response variable "target".

Outlier Detection

Since the dataset is not large, we cannot discard the outliers. We will treat the outliers as potential observations.

In [13]:

```
# Boxplots
fig_dims = (15,8)
fig, ax = plt.subplots(figsize=fig_dims)
sns.boxplot(data=hcare, ax=ax);
```



Handling Imbalance

Imbalance in a dataset leads to inaccuracy and high precision, recall scores. There are certain resampling techniques such as undersampling and oversampling to handle these issues.

Considering our dataset, the response variable target has two outcomes "Patients with Heart Disease" and "Patients without Heart Disease". Let us now observe their distribution in the dataset.

In [14]:

```
hcare["target"].value_counts()
```

Out[14]:

```
1    165
0    138
Name: target, dtype: int64
```

From the above chart, we can conclude even when the distribution is not exactly 50:50, but still the data is good enough to use on machine learning algorithms and to predict standard metrics like Accuracy and AUC scores. So, we do not need to resample this dataset.

Train-Test Split

Let us distribute the data into **training** and **test** datasets using the **train_test_split()** function.

In [15]:

```
X = hcare.drop("target",axis=1)
y = hcare["target"]
```

Logistic Regression

In [16]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,stratify=y,random_state=7)
```

In [17]:

```
from sklearn.linear_model import LogisticRegression
```

In [18]:

```
lr = LogisticRegression()
lr.fit(X_train, y_train)
```

Out[18]:

```
LogisticRegression()
```

In [19]:

```
pred = lr.predict(X_test)
```

In [20]:

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

In [21]:

```
# Accuracy on Test data
accuracy_score(y_test, pred)
```

Out[21]:

```
0.8032786885245902
```

In [22]:

```
# Accuracy on Train data
accuracy_score(y_train, lr.predict(X_train))
```

Out[22]:

```
0.8471074380165289
```

Building a predictive system

In [23]:

```
import warnings
in_data = (57,0,0,140,241,0,1,123,1,0.2,1,0,3)

# Changing the input data into a numpy array
in_data_as_numpy_array = np.array(in_data)

# Reshaping the numpy array as we predict it
in_data_reshape = in_data_as_numpy_array.reshape(1,-1)
pred = lr.predict(in_data_reshape)
print(pred)

if(pred[0] == 0):
    print('The person does not have heart disease.')
else:
    print('The person has heart disease.')
```

```
[0]
The person does not have heart disease.
```

- Rajeev Vhanhuve