stone4

رواد مصر الرقمية

DEPI Graduation Project

HealthCare Predictive analysis

Team Members:

- Amina Zeyad Atta Ibrahim
- Amr Ashour Mohamed Ali
- Elham Mokhtar Ahmed Mohammed
- Enas Samir Abbas Mohamed
- Esraa Mohamed Sayed Mohamed

Training Company: CLS

Group Number: CLS__CAI2_AIS4_G4

Instructor: Mahmoud El-Sayed

HealthCare Predictive analysis

Contents

List of Figures

1. Project Overview

The Healthcare Predictive Analytics specially **Cardiovascular Disease (CVD)** project focuses on developing a classification model to classify the patient if he/she suffers from cardiovascular disease or not by providing data-driven insights. The model will be designed to help healthcare professionals with tasks such as patient risk detection, making informed decisions based on predictive analytics. The project will utilize a classification model.

2. Milestone1

2.1 Data Collection:

Our Dataset was collected from Kaggle
Link:https://www.kaggle.com/datasets/scientificstephen/medical-examination-dataset-analysis

Link of Dataset on our drive:
https://drive.google.com/file/d/1uD5d16AkU_fdwTs3Fq6xqlGCN09A7o_7/view?usp=drive_link

2.2 Data Exploration:

Healthcare cardiovascular disease dataset consists of 70000 records of people and 13 factors taken into consideration in the dataset. The below figure shows a sample of the dataset

id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	18393	2	168	62	110	80	1	1	0	0	1	0
1	20228	1	156	85	140	90	3	1	0	0	1	1
2	18857	1	165	64	130	70	3	1	0	0	0	1
3	17623	2	169	82	150	100	1	1	0	0	1	1
4	17474	1	156	56	100	60	1	1	0	0	0	0
8	21914	1	151	67	120	80	2	2	0	0	0	0
9	22113	1	157	93	130	80	3	1	0	0	1	0
12	22584	2	178	95	130	90	3	3	0	0	1	1
13	17668	1	158	71	110	70	1	1	0	0	1	0
14	19834	1	164	68	110	60	1	1	0	0	0	0
15	22530	1	169	80	120	80	1	1	0	0	1	0
16	18815	2	173	60	120	80	1	1	0	0	1	0
18	14791	2	165	60	120	80	1	1	0	0	0	0
21	19809	1	158	78	110	70	1	1	0	0	1	0

Figure 1 Original Dataset sample

Description of each column (Key Features):

- id: this is just a number used to identify the patient.
- age: contains the age of each patient in days.
- gender: The column identify the sex of each patient (1: Female,2: Male).
- height: contains the height of each patient in meters (m).
- weight: contains the weight of each patient in kilograms (kg).
- ap_hi: Systolic of the patient in mmHg.
- ap_lo: Diastolic of the patient in mmHg.
- cholesterol: categorize the cholesterol level of each patient (1:Low, 2:Medium, 3:High).
- gluc: categorize the glucose level of each patient (1:Low, 2:Medium, 3:High).

- Smoke: Categorize the patient if he/she is a smoker or not.(1: Smoker, 0: Not a smoker)
- alco: Categorize the patient if he/she drinks alcohol or not. (1: drinker, 0: not a drinker)
- active: Categorize if the patient practices any sport or not (1: practice any activity, 0: not practice any activity).
- cardio: this is the target column in which classifies the patient if he/ she suffers from cardiovascular disease or not.

Data Summary:

1- Size: (13*70000) Thousands of individual records.

2- Type: Mixed numeric and categorical data.

3- Challenges: Includes outliers and categorical data requiring cleaning and preprocessing.

2.2.1 Check Data Statistics; First 5 records, Last 5 records, NULLs , Duplicates in Dataset and check if there is any categorical column needs encoding:

```
First 5 rows of the dataset:
   id    age  gender  height  weight  ap_hi  ap_lo  cholesterol  gluc  smoke  \
0   0  18393       2     168    62.0    110     80            1     1      0
1   1  20228       1     156    85.0    140     90            3     1      0
2   2  18857       1     165    64.0    130     70            3     1      0
3   3  17623       2     169    82.0    150    100            1     1      0
4   4  17474       1     156    56.0    100     60            1     1      0

   alco  active  cardio
0     0       1       0
1     0       1       1
2     0       0       1
3     0       1       1
4     0       0       0
```

Figure 2 First 5 records in the dataset

```
Last 5 rows of the dataset:
          id    age  gender  height  weight  ap_hi  ap_lo  cholesterol  gluc  \
69995  99993  19240       2     168    76.0    120     80            1     1
69996  99995  22601       1     158   126.0    140     90            2     2
69997  99996  19066       2     183   105.0    180     90            3     1
69998  99998  22431       1     163    72.0    135     80            1     2
69999  99999  20540       1     170    72.0    120     80            2     1

       smoke  alco  active  cardio
69995      1     0       1       0
69996      0     0       1       1
69997      0     1       0       1
69998      0     0       0       1
69999      0     0       1       0
```

Figure 3 Last 5 records in the dataset

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70000 entries, 0 to 69999
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   id           70000 non-null  int64
 1   age          70000 non-null  int64
 2   gender       70000 non-null  int64
 3   height       70000 non-null  int64
 4   weight       70000 non-null  float64
 5   ap_hi        70000 non-null  int64
 6   ap_lo        70000 non-null  int64
 7   cholesterol  70000 non-null  int64
 8   gluc         70000 non-null  int64
 9   smoke        70000 non-null  int64
 10  alco         70000 non-null  int64
 11  active       70000 non-null  int64
 12  cardio       70000 non-null  int64
dtypes: float64(1), int64(12)
memory usage: 6.9 MB
```

Figure 4 Checks on Data(columns type,NULLS)

```
Shape of Healthcare dataset ---> (70000, 13)
Check Duplication in the dataset ---> 0
Check Nulls in the dataset --->
 id             0
age            0
gender         0
height         0
weight         0
ap_hi          0
ap_lo          0
cholesterol    0
gluc           0
smoke          0
alco           0
active         0
cardio         0
dtype: int64
```

Figure 5 Checks on Duplicates and Nulls

From the above figure it is found that:

- No Duplicates.
- No NULLs.
- No need for encoding.

2.2.2 Data Distribution and Handling:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke
count	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000
mean	49972.419900	19468.865814	1.349843	164.359229	74.205690	128.817286	96.630414	1.366871	1.226457	0.088129
std	28851.302323	2467.251667	0.477253	8.210126	14.395757	154.011419	188.472530	0.680250	0.572270	0.283484
min	0.000000	10798.000000	1.000000	55.000000	10.000000	-150.000000	-70.000000	1.000000	1.000000	0.000000
25%	25006.750000	17664.000000	1.000000	159.000000	65.000000	120.000000	80.000000	1.000000	1.000000	0.000000
50%	50001.500000	19703.000000	1.000000	165.000000	72.000000	120.000000	80.000000	1.000000	1.000000	0.000000
75%	74889.250000	21327.000000	2.000000	170.000000	82.000000	140.000000	90.000000	2.000000	1.000000	0.000000
max	99999.000000	23713.000000	3.000000	250.000000	200.000000	16020.000000	11000.000000	3.000000	3.000000	1.000000

Figure 6 Data Description Part1

alco	active	cardio
70000.000000	70000.000000	70000.000000
0.053771	0.803729	0.499700
0.225568	0.397179	0.500003
0.000000	0.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	0.000000
0.000000	1.000000	1.000000
1.000000	1.000000	1.000000

Figure 7 Data Description part2

From the above figures, it is found that the age has very large numbers as it is calculated in days.

So we have converted days into years then we have shown a sample of the data after conversion as follow

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	50	2	168	62.0	110	80	1	1	0	0	1	0
1	1	55	1	156	85.0	140	90	3	1	0	0	1	1
2	2	51	1	165	64.0	130	70	3	1	0	0	0	1
3	3	48	2	169	82.0	150	100	1	1	0	0	1	1
4	4	47	1	156	56.0	100	60	1	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
69995	99993	52	2	168	76.0	120	80	1	1	1	0	1	0
69996	99995	61	1	158	126.0	140	90	2	2	0	0	1	1
69997	99996	52	2	183	105.0	180	90	3	1	0	1	0	1
69998	99998	61	1	163	72.0	135	80	1	2	0	0	0	1
69999	99999	56	1	170	72.0	120	80	2	1	0	0	1	0

70000 rows × 13 columns

Figure 8 Sample of Data after age conversion

```
[ ] Healthcare_Cleaned.describe()
```

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc
count	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000	70000.000000
mean	49972.419900	52.840671	1.349843	164.359229	74.205690	128.817286	96.630414	1.366871	1.226457
std	28851.302323	6.766774	0.477253	8.210126	14.395757	154.011419	188.472530	0.680250	0.572270
min	0.000000	29.000000	1.000000	55.000000	10.000000	-150.000000	-70.000000	1.000000	1.000000
25%	25006.750000	48.000000	1.000000	159.000000	65.000000	120.000000	80.000000	1.000000	1.000000
50%	50001.500000	53.000000	1.000000	165.000000	72.000000	120.000000	80.000000	1.000000	1.000000
75%	74889.250000	58.000000	2.000000	170.000000	82.000000	140.000000	90.000000	2.000000	1.000000
max	99999.000000	64.000000	3.000000	250.000000	200.000000	16020.000000	11000.000000	3.000000	3.000000

Figure 9 Data Description after age conversion. Changes are in age

Then we have drown distribution of each key feature as follow:

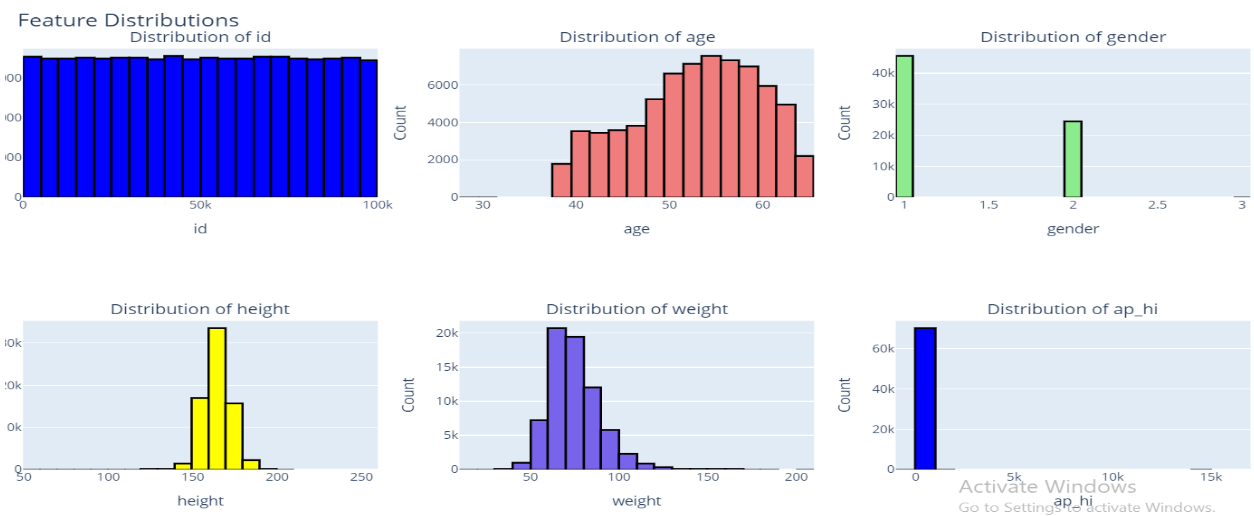


Figure 10 Features Distribution part1

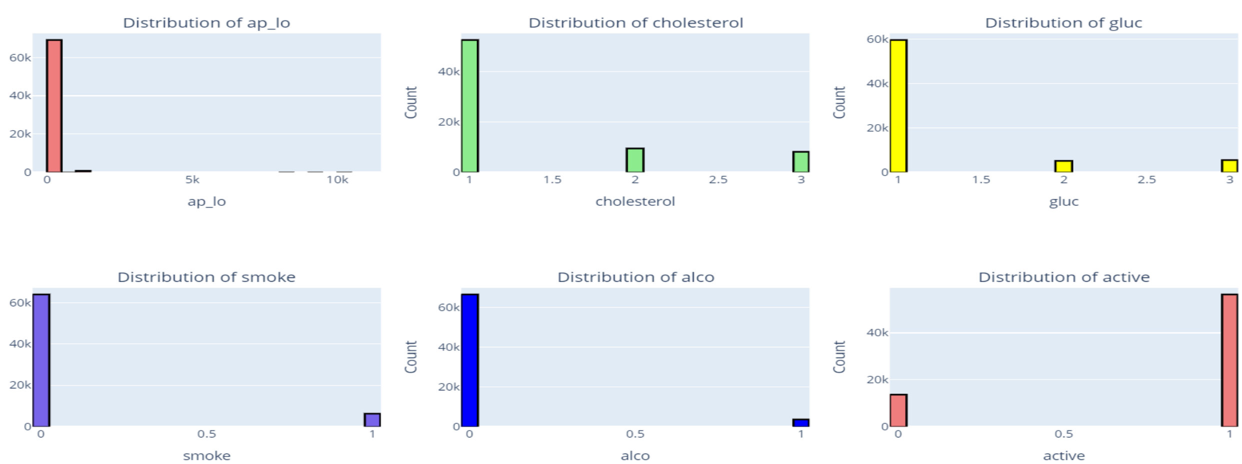


Figure 11 Features Distribution Part2

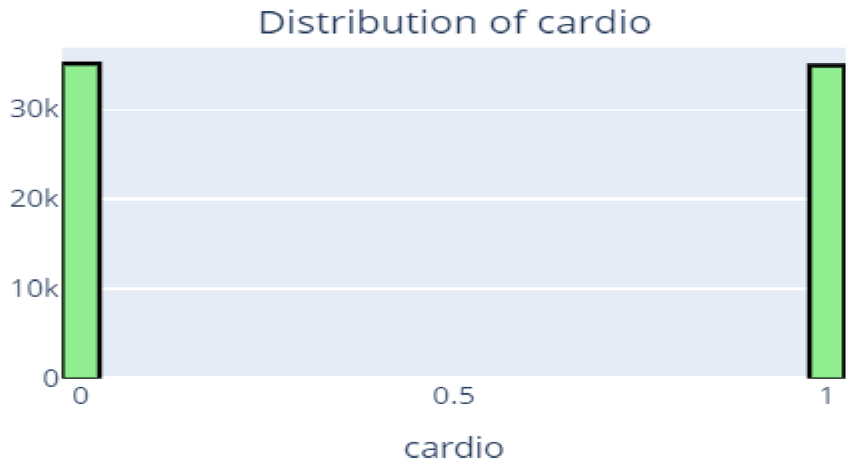


Figure 12 Features Distribution Part3

From the above figure we found that:

- Age distribution is left skewed data so it needs normalization to convert the left skewed to Gaussian distribution.
- Gender contains 3 categories which is not logic as 0 for Males, 1 for Females and 2 for what?! so it depends on the number of samples of this category. the number of samples is 11 records as shown below, it is recommended to eliminate it.

```
gender
1    45522
2    24467
3       11
Name: count, dtype: int64
gender
1    45522
2    24467
Name: count, dtype: int64
```

Figure 13 Number of records in which gender=3

- Height and weight are almost Gaussian distribution.
- ap_hi and ap_lo seem to have outliers as the maximum values from data description are 16020 and 11000 respectively which are not logic values and number of samples at these values are not large so we can handle them by elimination as follow.
- In features the most dominant samples are the normal samples of people but cardiovascular column is balanced which means that there are outliers in the dataset.

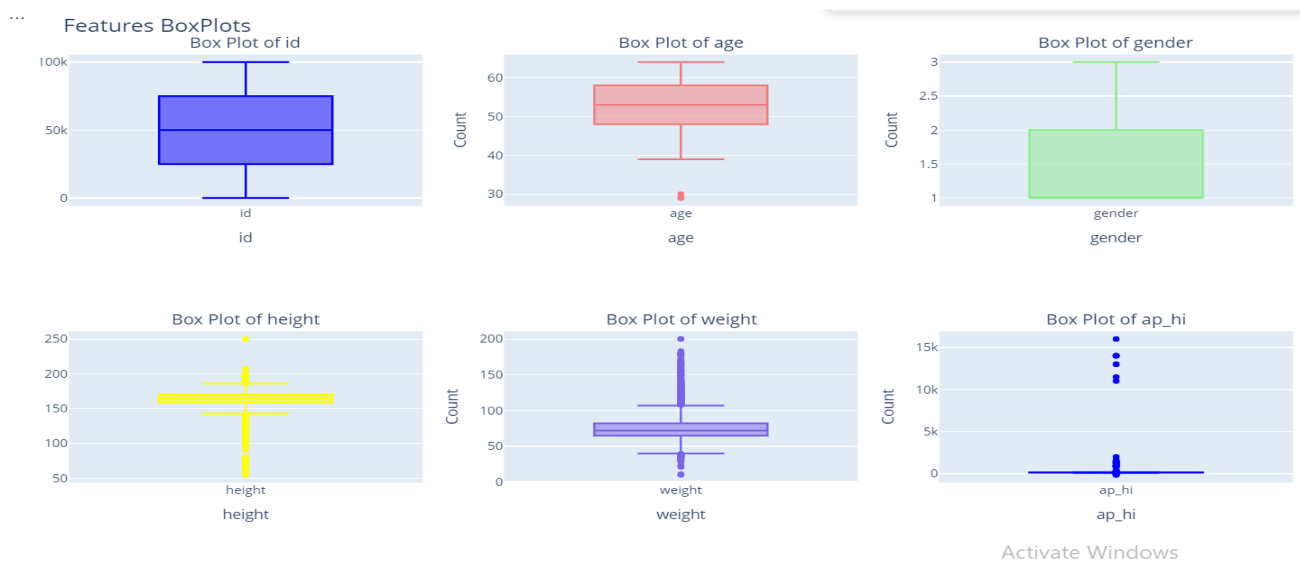


Figure 14 Features Box plots part1

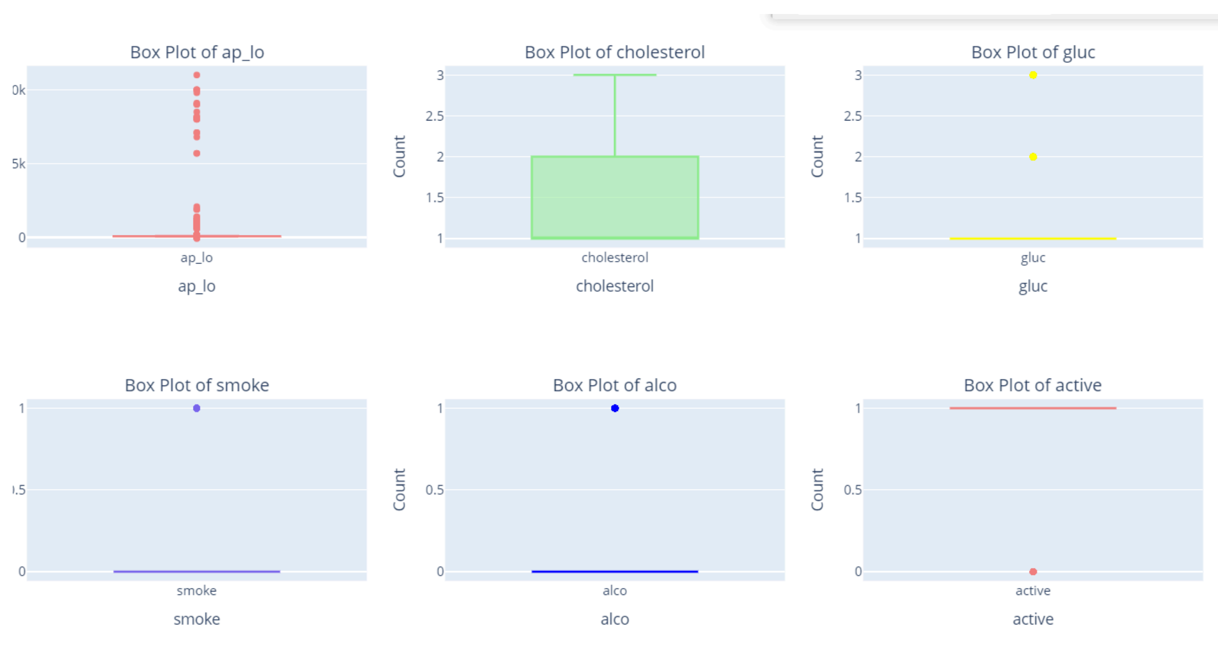


Figure 15 Features Box Plots part2

From the above figures we found that:

- The minimum and maximum values of age are 29 and 64 respectively which are normal values so we decided to keep it as it is.

- In height column minimum and maximum values are 55 and 250 cm respectively. But some of these values are not logic and not correlated with age. Thresholds we have chosen are 100 cm for the lower threshold and 200 cm for the upper threshold. Based on these thresholds outlier samples count was 31 sample so we decided to eliminate them.
- In weight column minimum and maximum values are 10 and 200 kg respectively. Value 10 does not match with the minimum value of age 29 so our thresholds for weight were 45 kg as a lower threshold and 190 kg as an upper threshold. We found that 304 samples were out of this range (outlier) so we decided to eliminate them.
- For ap_hi and ap_lo thresholds chosen are dependent on medical domain as follow:

Condition	Possible Lowest Systolic (`ap_hi`)	Description
Healthy (Normal Range)	**90 mmHg**	Generally considered the safe lower limit for normal individuals.
Hypotension	**60 - 89 mmHg**	May cause dizziness, fainting, risk of shock if too low.
Critical Hypotension	**Below 60 mmHg**	Associated with severe conditions like shock, organ failure, or trauma.

Figure 16 Possible Lowest Systolic (ap_hi)

From the above figure we found that the possible lowest systolic value is 60 mmHg below that the human will not be alive. But if there is a noise in the device used to measure the pressure this may affect the measurement to we have chosen the lower threshold to be 50 mmHg.

Category	Systolic (ap_hi)	Description
Normal	90 - 119 mmHg	Ideal blood pressure.
Elevated	120 - 129 mmHg	Increased risk if not managed.
Hypertension Stage 1	130 - 139 mmHg	Requires lifestyle changes or medication.
Hypertension Stage 2	140 - 179 mmHg	High risk of cardiovascular disease; treatment needed.
Hypertensive Crisis	180 mmHg and above	Immediate medical attention required.

Figure 17 Possible Highest Systolic(ap_hi)

From the above figure we found that the maximum value for ap_hi is 180 mmHg but during search we found that the maximum value ap_hi can be taken and the human is a live is around 250 mmHg.

So thresholds chosen for ap_hi → from 50 mmHg to 250 mmHg.

Based on the above thresholds it is found that 224 samples are considered as outliers so we decided to eliminate these samples.

- For ap_lo from search we found that:

Condition	Possible Diastolic (ap_lo) Range	Description
Normal	60 - 79 mmHg	Healthy blood pressure.
Elevated	80 - 89 mmHg	Potential risk of hypertension.
Hypertension Stage 1	90 - 99 mmHg	Requires monitoring and treatment.
Hypertension Stage 2	100 - 119 mmHg	High risk; medical treatment often required.
Hypertensive Crisis	120 - 150 mmHg	Emergency; very rare to exceed this range in valid data.
Hypotension (Low BP)	40 - 59 mmHg	May cause dizziness and fainting; emergency if too low.
Critical Low (Possible Error)	Below 40 mmHg	Unlikely to be physiologically valid, potential data error.

Figure 18 Thresholds for ap_lo

From The above figure we found that the minimum value for ap_lo is 40 mmHg and the upper threshold 150 mmHg.

Based on the previous thresholds we found that 1013 samples as outlier we decided not to eliminate these samples as they represent 1.4%.

The logic used to impute the outliers is as follow:

- By logic, people whose ap_lo is lower than 40 mmHg and higher than 150 mmHg definitely suffer from cardiovascular disease which means cardio flag=1. This will give us subset of data. Number of samples in this subset is 837 records. we decide to calculate the mean of ap_lo of these people and impute the outlier values with this mean.

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke
count	33912.000000	33912.000000	33912.000000	33912.000000	33912.000000	33912.000000	33912.000000	33912.000000	33912.000000	33912.000000
mean	50080.200047	54.462373	1.353149	164.323602	76.795081	133.829323	84.620370	1.516749	1.278043	0.083510
std	28825.283297	6.353931	0.478202	8.057732	14.745338	17.342840	9.630336	0.776686	0.625256	0.276656
min	1.000000	39.000000	1.000000	100.000000	45.000000	70.000000	45.000000	1.000000	1.000000	0.000000
25%	25257.500000	50.000000	1.000000	159.000000	66.000000	120.000000	80.000000	1.000000	1.000000	0.000000
50%	50134.500000	55.000000	1.000000	165.000000	75.000000	130.000000	80.000000	1.000000	1.000000	0.000000
75%	74996.500000	60.000000	2.000000	170.000000	85.000000	140.000000	90.000000	2.000000	1.000000	0.000000
max	99998.000000	64.000000	3.000000	198.000000	183.000000	240.000000	140.000000	3.000000	3.000000	1.000000

Figure 19 Mean of ap_hi of subset (ap_hi < 40mmHg & ap_hi > 150)

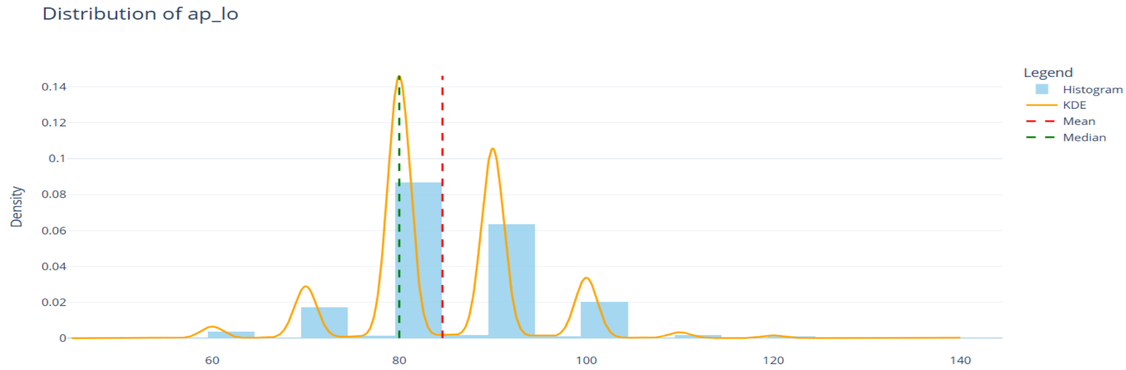


Figure 20 Distribution of ap_lo filtered on (<40 & >150 & cardio=1)

From the above figure we found that mean=84 and median=80 so we substituted by mean.

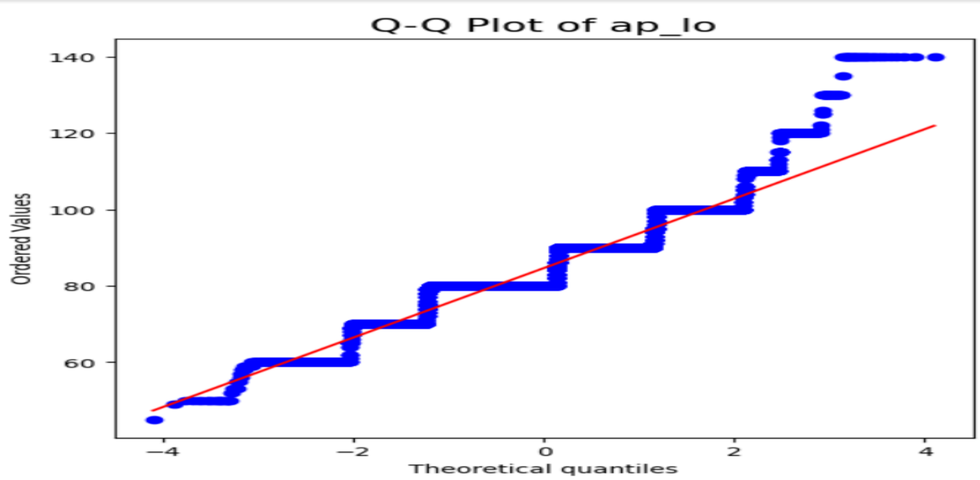


Figure 21 Q-Q Plot of ap_lo after removing outliers

This is a Q-Q plot of ap_lo after substitution which means that the data distribution almost became normal distribution.

```
Healthcare_Cleaned
```

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	50	2	168	62.0	110	80	1	1	0	0	1	0
1	1	55	1	156	85.0	140	90	3	1	0	0	1	1
2	2	51	1	165	64.0	130	70	3	1	0	0	0	1
3	3	48	2	169	82.0	150	100	1	1	0	0	1	1
4	4	47	1	156	56.0	100	60	1	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
69995	99993	52	2	168	76.0	120	80	1	1	1	0	1	0
69996	99995	61	1	158	126.0	140	90	2	2	0	0	1	1
69997	99996	52	2	183	105.0	180	90	3	1	0	1	0	1
69998	99998	61	1	163	72.0	135	80	1	2	0	0	0	1
69999	99999	56	1	170	72.0	120	80	2	1	0	0	1	0

69441 rows × 13 columns

Figure 22 Sample of Dataset after cleaning

```
Healthcare_Cleaned.describe()
```

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc
count	69416.000000	69416.000000	69416.000000	69416.000000	69416.000000	69416.000000	69416.000000	69416.000000	69416.000000
mean	49957.639492	52.847903	1.351086	164.442780	74.349992	127.060303	81.424124	1.367711	1.227008
std	28854.763496	6.761097	0.477645	7.953601	14.227458	17.068139	9.477888	0.681076	0.572872
min	0.000000	29.000000	1.000000	100.000000	45.000000	60.000000	40.000000	1.000000	1.000000
25%	24982.750000	48.000000	1.000000	159.000000	65.000000	120.000000	80.000000	1.000000	1.000000
50%	49980.500000	53.000000	1.000000	165.000000	72.000000	120.000000	80.000000	1.000000	1.000000
75%	74882.000000	58.000000	2.000000	170.000000	82.000000	140.000000	90.000000	2.000000	1.000000
max	99999.000000	64.000000	3.000000	200.000000	183.000000	240.000000	150.000000	3.000000	3.000000

Figure 23 Data Description after cleaning

There is a bug found during data investigation:

It is found that there are some records in which ap_lo > ap_hi which does not make sense.

2.2.2.1 Features Distribution and Box plots after data handling

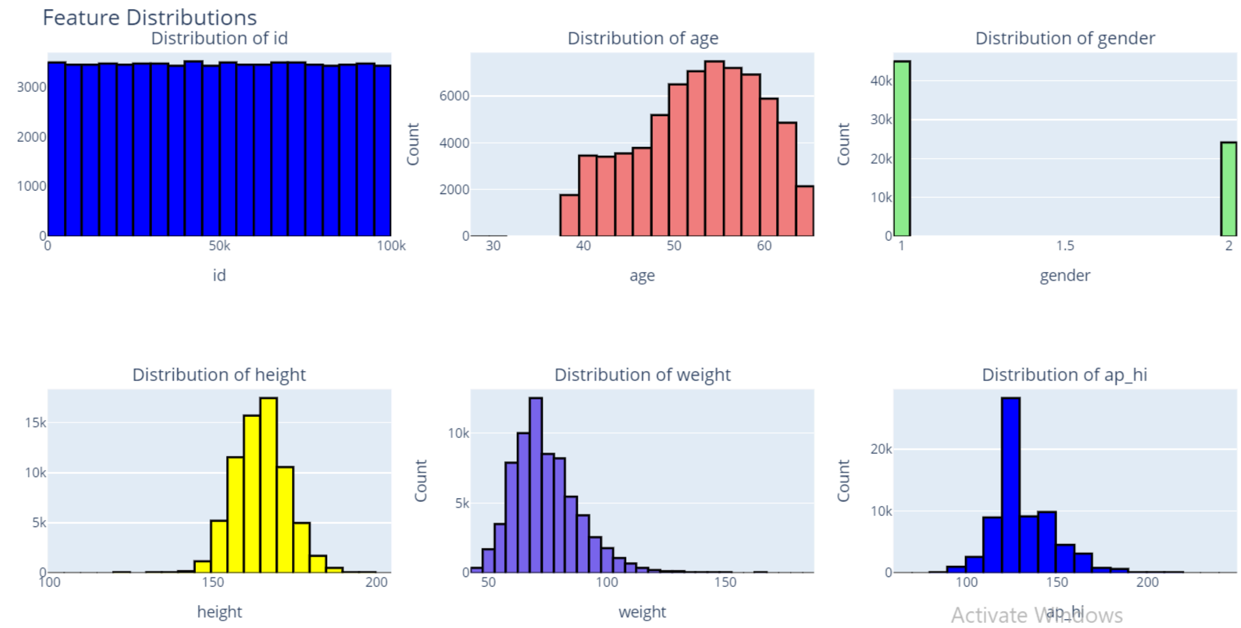


Figure 24 Features Distribution Part1

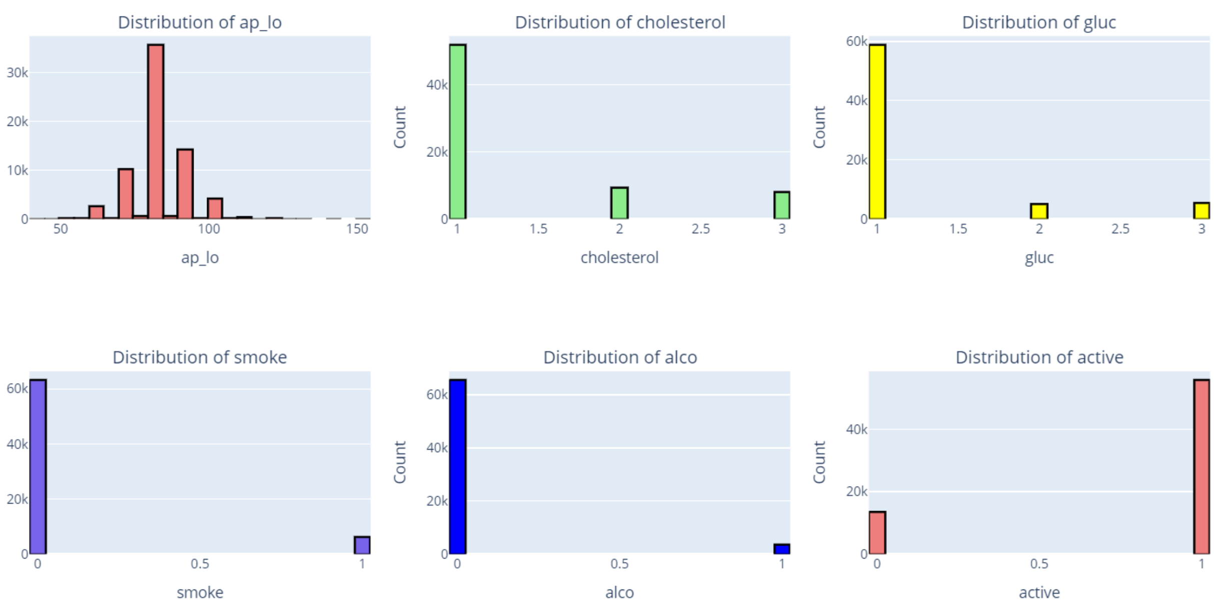


Figure 25 Features Distribution Part2

From the above figures it is found that:

- *Gender feature now contains only two categories.*
- *Distribution of ap_hi & ap_lo are enhanced*

3. Milestone2

3.1 Exploratory Data Analysis (EDA)

In this part it is required to extract some features from the dataset which called Feature Engineering.

3.1.1 Body Mass Indicator (BMI)

It is the measure which relates between the human's wright and his height to know if he suffers from obesity or not.

Its formula as shown bemow we have used the first formula.

The formula is:

$$\text{BMI} = \frac{\text{Weight (kg)}}{\text{Height (m)}^2}$$

Or in pounds and inches:

$$\text{BMI} = \frac{\text{Weight (lb)} \times 703}{\text{Height (in)}^2}$$

Figure 26 BMI formula

Based on search the BMI category is:

📊 **BMI Categories:**

BMI Range	Category
< 18.5	Underweight
18.5 - 24.9	Normal weight
25 - 29.9	Overweight
30 and above	Obesity

Figure 27 BMI Categories

We have divided our data into 4 categories

1⟶ Underweight

2⟶ Normal weight

3⟶ Overweight

4⟶ Obesity

So we did not encode this column as the categories are mapped to numbers.

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio	bmi	bmi_category
0	0	50	2	168	62.0	110	80	1	1	0	0	1	0	21.967120	2
1	1	55	1	156	85.0	140	90	3	1	0	0	1	1	34.927679	4
2	2	51	1	165	64.0	130	70	3	1	0	0	0	1	23.507805	2
3	3	48	2	169	82.0	150	100	1	1	0	0	1	1	28.710479	3
4	4	47	1	156	56.0	100	60	1	1	0	0	0	0	23.011177	2

Figure 28 Sample of dataset after adding bmi & bmi_category columns

3.1.2 Hypertension Feature

We have thought that we have ap_hi and ap_lo from that we can know that if the client suffers from hypertension or not based on given thresholds attached above we found that if ap_hi> 130 mmHg, it is stage1 of hypertension and if ap_lo >90, it is stage 1 of hypertension. so if(ap_hi>130 or ap_lo>90) hypertension flag will be mapped to 1 as shown below.

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio	bmi	bmi_category	hypertension
0	0	50	2	168	62.0	110	80	1	1	0	0	1	0	21.967120	2	0
1	1	55	1	156	85.0	140	90	3	1	0	0	1	1	34.927679	4	1
2	2	51	1	165	64.0	130	70	3	1	0	0	0	1	23.507805	2	0
3	3	48	2	169	82.0	150	100	1	1	0	0	1	1	28.710479	3	1
4	4	47	1	156	56.0	100	60	1	1	0	0	0	0	23.011177	2	0

Figure 29 Sample of Dataset after adding hypertension column

3.1.3 Pulse Pressure Feature (PP) & Mean Arterial Pressure (MAP)

These features give us relation between ap_hi and ap_lo and from these features we can detect if the user suffers from cardiovascular or not. the formula of PP and MAP as follow:

$$\mathrm{PP} = \text{Systolic Blood Pressure} - \text{Diastolic Blood Pressure}$$

Figure 30 Pulse Pressure PP formula

$$\mathrm{MAP} = \text{Diastolic BP} + \frac{1}{3}(\text{Systolic BP} - \text{Diastolic BP})$$

Figure 31 Mean Arterial Pressure (MAP) formula

id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	bmi	bmi_category	hypertension	MAP	PP	cardio
0	50	2	168	62.0	110	80	1	1	0	0	1	21.967120	2	0	90.000000	30	0
1	55	1	156	85.0	140	90	3	1	0	0	1	34.927679	4	1	106.666667	50	1
2	51	1	165	64.0	130	70	3	1	0	0	0	23.507805	2	0	90.000000	60	1
3	48	2	169	82.0	150	100	1	1	0	0	1	28.710479	3	1	116.666667	50	1
4	47	1	156	56.0	100	60	1	1	0	0	0	23.011177	2	0	73.333333	40	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
99993	52	2	168	76.0	120	80	1	1	1	0	1	26.927438	3	0	93.333333	40	0
99995	61	1	158	126.0	140	90	2	2	0	0	1	50.472681	4	1	106.666667	50	1
99996	52	2	183	105.0	180	90	3	1	0	1	0	31.353579	4	1	120.000000	90	1
99998	61	1	163	72.0	135	80	1	2	0	0	0	27.099251	3	1	98.333333	55	1
99999	56	1	170	72.0	120	80	2	1	0	0	1	24.913495	3	0	93.333333	40	0

Figure 32 Dataset sample after adding MAP and PP features

3.1.4 Age group and PP_Category Features:

In this part we wanted to categorize the clients into age groups to know which group suffers from cardiovascular as by logic when age group increases the probability that the user suffers from cardio disease is higher. and want to categorize the PP into features.

PP_Category:

from 0-->30 category 1 --> very low pressure (0-30 mmHg)

from 30-->60 category 2--> Normal/Low-Normal Pulse Pressure (30-60 mmHg)

from 60-->90 category 3--> Moderately Elevated Pulse Pressure (60-90 mmHg)

from 90-->120 category 4--> High Pulse Pressure (90-120 mmHg)

from 120-->140 category 5--> Very High Pulse Pressure (120-140 mmHg)

Age_group:

from 20-29 ⟶ category 1

from 30-39 ⟶ category 2

from 40-49 ⟶ category 3

from 50-59 ⟶ categroy 4

from 60-69 ⟶ category 5

from 70-79 ⟶ category 6

weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	bmi	bmi_category	hypertension	MAP	PP	cardio	pp_category	age_group
62.0	110	80	1	1	0	0	1	21.967120	2	0	90.000000	30	0	1	50-59
85.0	140	90	3	1	0	0	1	34.927679	4	1	106.666667	50	1	2	50-59
64.0	130	70	3	1	0	0	0	23.507805	2	0	90.000000	60	1	2	50-59
82.0	150	100	1	1	0	0	1	28.710479	3	1	116.666667	50	1	2	40-49
56.0	100	60	1	1	0	0	0	23.011177	2	0	73.333333	40	0	2	40-49
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
76.0	120	80	1	1	1	0	1	26.927438	3	0	93.333333	40	0	2	50-59
126.0	140	90	2	2	0	0	1	50.472681	4	1	106.666667	50	1	2	60-69
105.0	180	90	3	1	0	1	0	31.353579	4	1	120.000000	90	1	3	50-59
72.0	135	80	1	2	0	0	0	27.099251	3	1	98.333333	55	1	2	60-69
72.0	120	80	2	1	0	0	1	24.913495	3	0	93.333333	40	0	2	50-59

Figure 33 Dataset sample after adding PP_category and age_group

3.2 Data Visualization

Figure 34 BMI Category & Age group vs cardio

From the above figure it is found that:

- Cardiovascular is directly proportional with both BMI_Category & Age_group categories for example let us take the extreme case

  when BMI_Category =4 and ae_group is the maximum category (60-69) it has the maximum percentage of people who suffer from cardiovascular disease.

  another example which validates the assumption

  when BMI_category=1 (lowest BMI_Category) and age_group from 20-29 it has the lowest percentage of clients who suffered from cardiovascular disease.

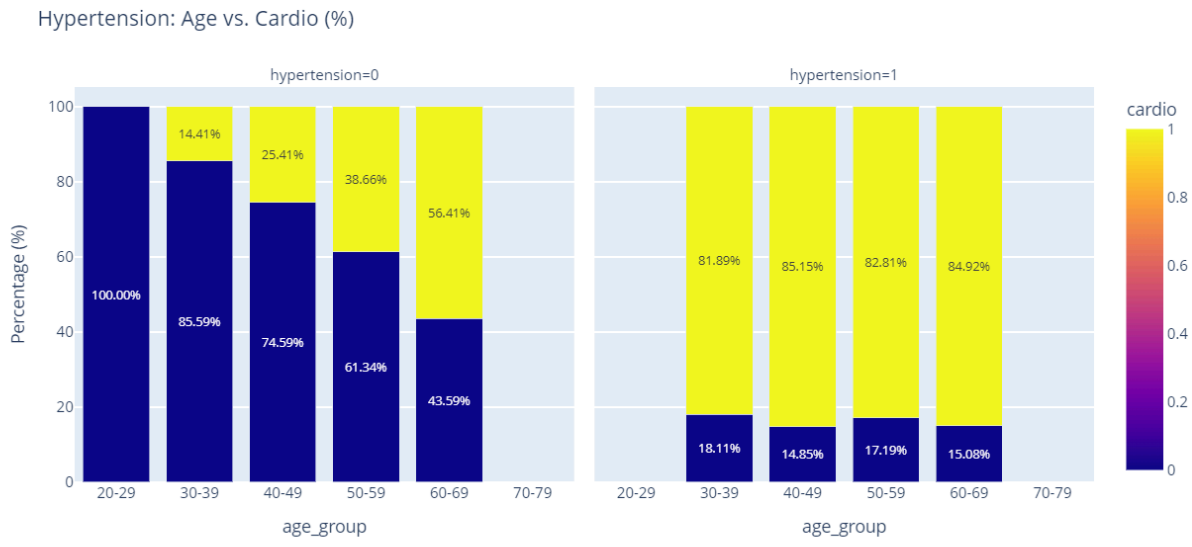


Figure 35 Hypertension_category & age_group vs Cardio

From the above photo it is found that

- Age group and hypertension both of them affect the client if he suffers from cardiovascular problem or not.

3.3 Check Dependency among features

3.3.1 Check dependency between categorical columns

As the target column is categorical (binary classification problem) and there are some categorical features ; `['gender', 'cholesterol', 'gluc', 'smoke', 'alco','active', 'bmi_category', 'hypertension','pp_category','age_group']`

We have used Chi2_Contengency . It assumes Null hypothesis in which assumes the features are independent of each other and calculate p if p<=0.05 this means that the columns are dependent.

	Feature	p-value
0	cholesterol	0.000000e+00
1	bmi_category	0.000000e+00
2	hypertension	0.000000e+00
3	pp_category	0.000000e+00
4	age_group	0.000000e+00
5	gluc	2.601954e-127
6	active	3.069004e-21
7	smoke	3.865160e-05
8	alco	4.744033e-02
9	gender	6.143869e-02

Figure 36 Chi2_Contengency between target column and features.

From the above figure we found that the target column depends on all features except gender and low relationship with alco_column.

3.3.2 Check Dependency between Target column and numerical features

	Feature	Correlation (r_pb)	P-value
1	height	-0.015441	0.000049
2	weight	0.180741	0.000000
5	bmi	0.187504	0.000000
0	age	0.238013	0.000000
4	ap_lo	0.340721	0.000000
7	PP	0.345835	0.000000
6	MAP	0.415388	0.000000
3	ap_hi	0.434577	0.000000

Figure 37 Pi-serial correlation

We find that the most effective features are ap_hi, MAP ,PP & ap_lo. Moderate effective features age, BMI, Weight.

3.3.3 Check Dependency among Features

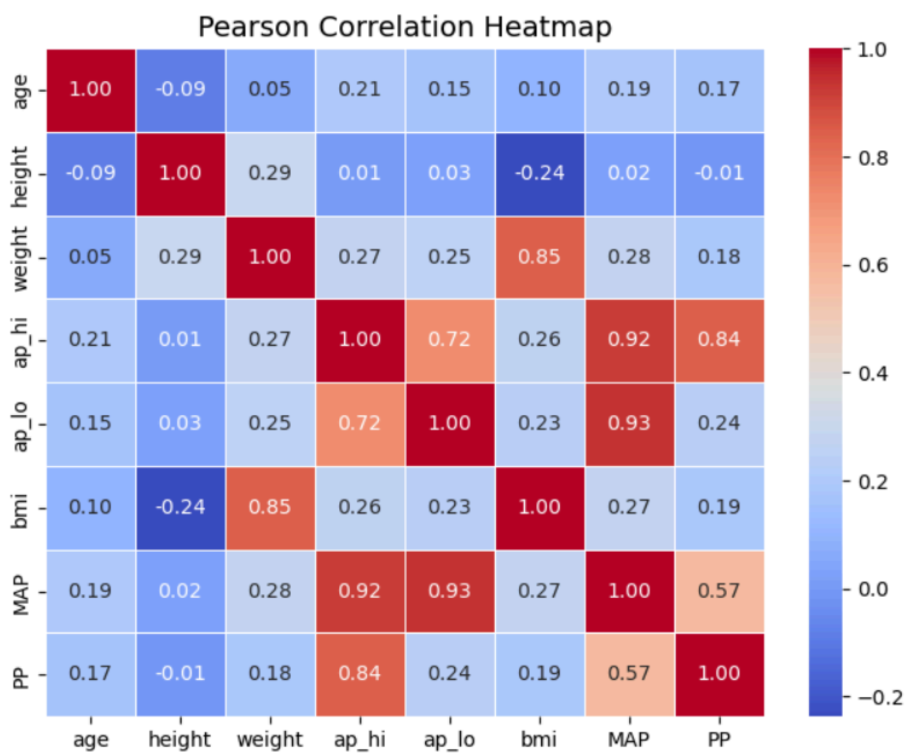


Figure 38 Pearson Correlation among features

3.3 Check Target column balance

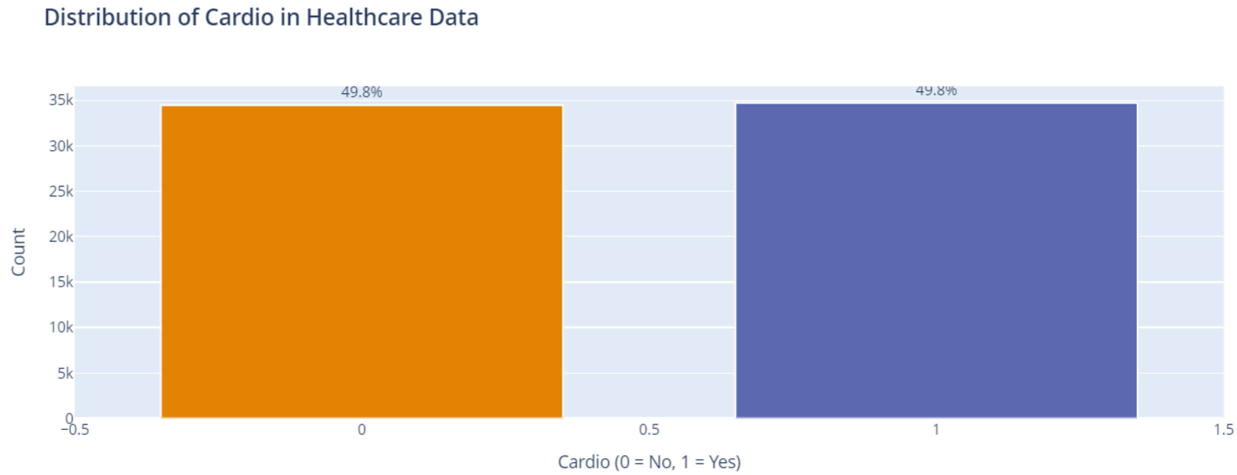


Figure 39 Cardio column

The target column is balanced.

3.4 Data Scaling

Applying Standard Scaler to make sure that all values within the same range

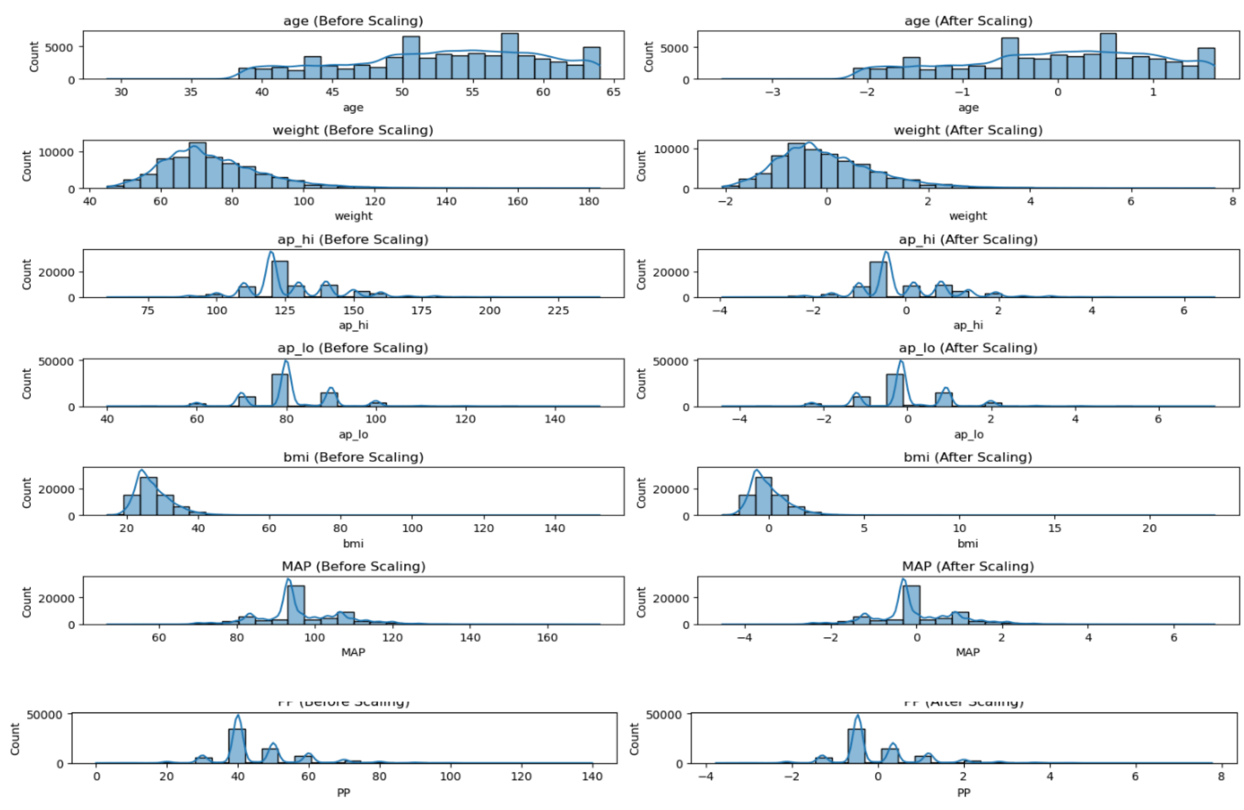


Figure 40 Data Before scaling

Applying PowerTransform to make sure that the features are gaussian distribution as there are some models require the input data has gaussian distribution.

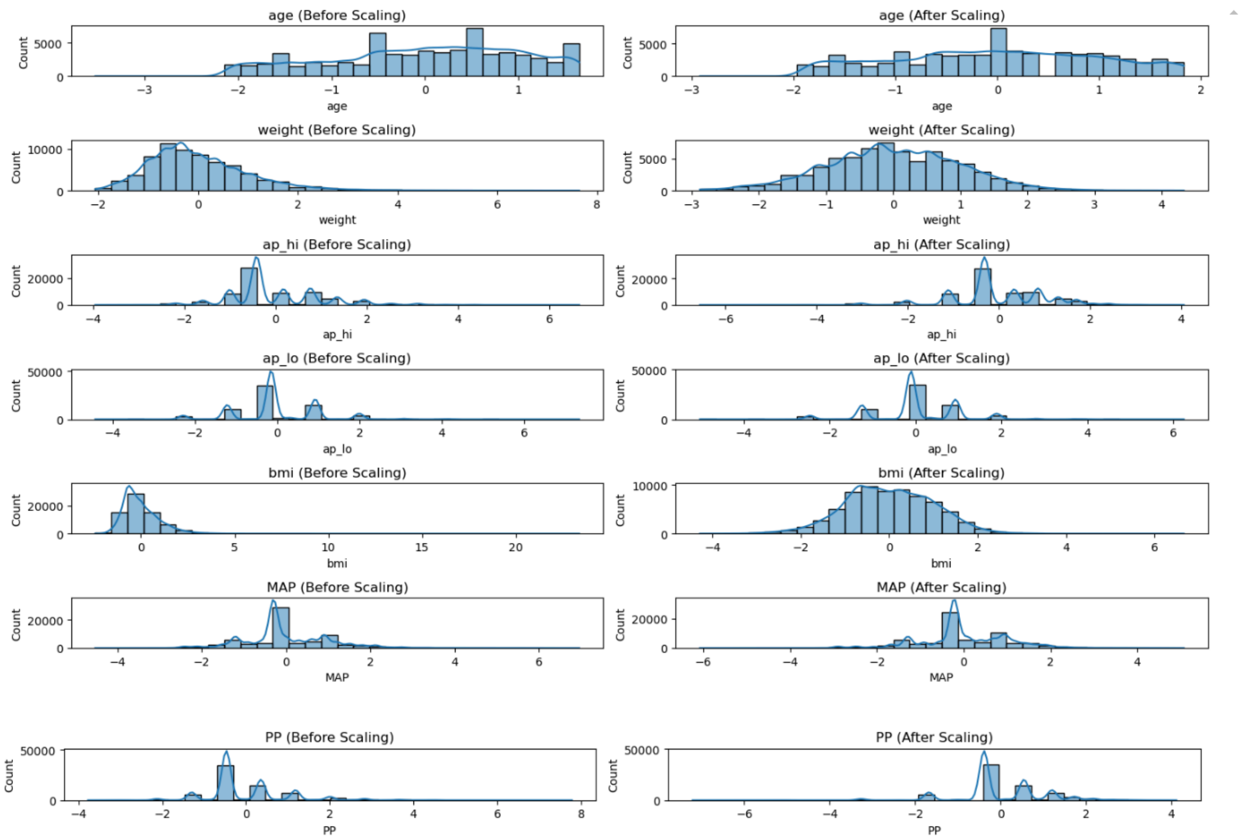


Figure 41 PowerTransform of features

age	weight	ap_hi	ap_lo	bmi	MAP	PP	cholesterol	gluc	smoke	alco	active	bmi_category	hypertension
-0.496434	-0.919960	-1.110842	-0.106249	-1.293991	-0.560369	-1.679330	1	1	0	0	1	2	0
0.242977	0.854561	0.854685	0.927890	1.347881	0.935158	0.546811	3	1	0	0	1	4	1
-0.357816	-0.721934	0.332997	-1.247431	-0.806127	-0.560369	1.204126	3	1	0	0	0	2	0
-0.763359	0.679557	1.300350	1.889918	0.452763	1.671363	0.546811	1	1	0	0	1	3	1
-0.892422	-1.557559	-2.018598	-2.461003	-0.958051	-2.477711	-0.387254	1	1	0	0	0	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
-0.215092	0.289992	-0.318956	-0.106249	0.093094	-0.221929	-0.387254	1	1	1	0	1	3	0
1.273647	2.630318	0.854685	0.927890	2.719509	0.935158	0.546811	2	2	0	0	1	4	1
-0.215092	1.823483	2.392772	0.927890	0.882066	1.902061	2.574213	3	1	0	1	0	4	1
1.273647	-0.010868	0.605976	-0.106249	0.131080	0.246717	0.897602	1	2	0	0	0	3	1
0.405222	-0.010868	-0.318956	-0.106249	-0.404616	-0.221929	-0.387254	2	1	0	0	1	3	0

ws × 16 columns

Figure 42 Sample of data after scaling

Data Visualization using PowerBI:

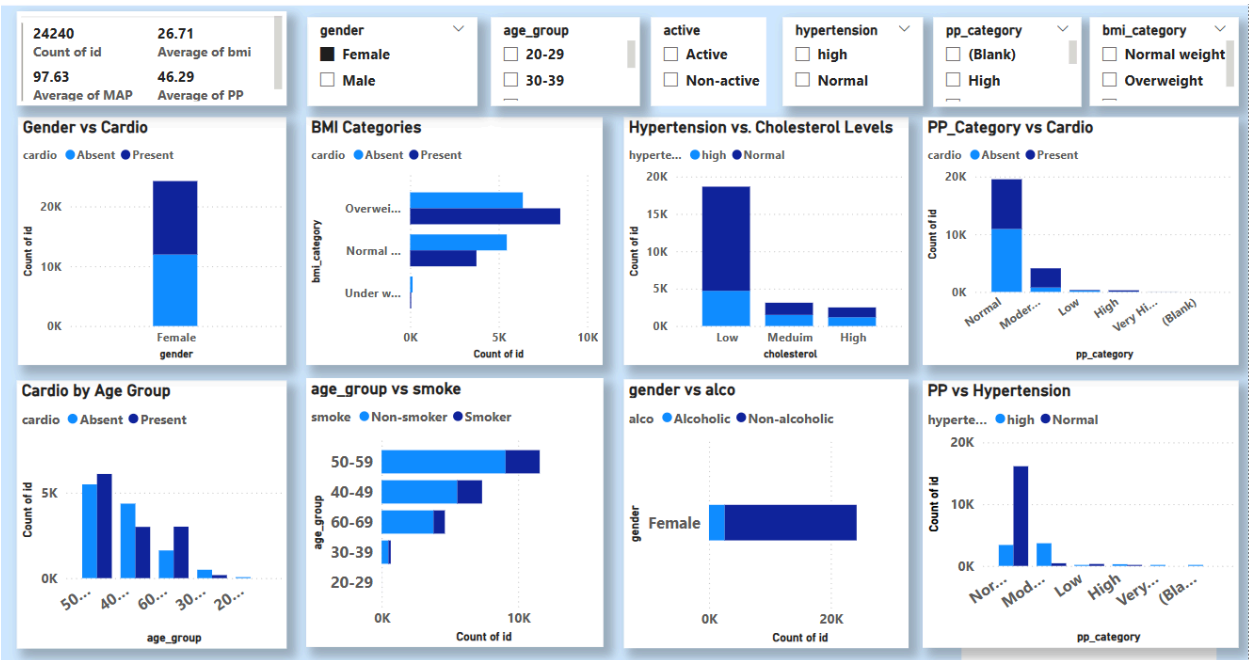


Figure 43 SnapShot of PowerBI

4. Milestone3

4.1 Model Selection

Our problem is a supervised machine learning (binary classification) so we have detected all possible models can be used in binary classification problem.

- Logistic Regression
- Random Forest
- Gradient Boosting
- XGBoost
- LightGBM
- Naive Bayes
- SVM (Support vector machine)
- Decision Tree

So our target is to filter among these models which will fit our data.

4.1.1 Trial 1

Our first trial we try all these models and trained them on all features without excluding any feature

the comparison between these models.

Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	Train F1	Test F1
Logistic Regression	0.7311	0.7296	0.7770	0.7729	0.6510	0.6530	0.7084	0.7079
Random Forest	0.9734	0.7007	0.9816	0.7013	0.9650	0.7030	0.9732	0.7021
Gradient Boosting	0.7407	0.7351	0.7661	0.7582	0.6957	0.6930	0.7292	0.7242
XGBoost	0.7655	0.7369	0.7913	0.7560	0.7236	0.7022	0.7559	0.7282
LightGBM	0.7498	0.7380	0.7729	0.7587	0.7101	0.7007	0.7402	0.7285
Naive Bayes	0.7252	0.7250	0.7850	0.7824	0.6231	0.6260	0.6948	0.6955
SVM	0.7357	0.7340	0.7826	0.7771	0.6553	0.6589	0.7134	0.7131
Decision Tree	0.9734	0.6391	0.9929	0.6425	0.9538	0.6328	0.9730	0.6376

Based on the above analysis we are interested on both precision and recall we found that

The top 3 models in Test F1:

- Gradient Boosting
- XGBoost
- LightGBM

The top 3 models in Test Recall:

- Random Forest
- XGBoost
- LightGBM

The top 3 models in Test Precision:

- Light GBM
- Navie Bayes
- SVM

The intersection among all these matrics is ⟶ LightGBM

Initially we will take LightGBM into our consideration.

4.1.2 Trial 2

We can use random forest to detect the importance of features on the target column we decided to filter on top7 out of 13 features

Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	Train F1	Test F1
Logistic Regression	0.7298	0.7296	0.7776	0.7738	0.6464	0.6515	0.7060	0.7074
Random Forest	0.9571	0.6989	0.9695	0.7006	0.9443	0.6982	0.9567	0.6994
Gradient Boosting	0.7380	0.7359	0.7594	0.7538	0.6995	0.7034	0.7283	0.7277
XGBoost	0.7605	0.7300	0.7846	0.7500	0.7205	0.6929	0.7512	0.7203
LightGBM	0.7458	0.7354	0.7670	0.7534	0.7088	0.7028	0.7368	0.7272
Naive Bayes	0.7246	0.7242	0.7965	0.7924	0.6062	0.6103	0.6884	0.6896
SVM	0.7348	0.7362	0.7654	0.7636	0.6799	0.6868	0.7201	0.7232
Decision Tree	0.9572	0.6389	0.9875	0.6460	0.9264	0.6206	0.9560	0.6330

But No gain from this trial.

4.1.3 Trial 3

From the above two trials there are models overfitting like Random Forest and Decision Tree and there are models under fit such as Navie Bayes so it we take voting among these models this may affect the accuracy. But unfortunately it does not happen as shown below.

Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	Train F1	Test F1
voting_clf	0.8268	0.7337	0.8637	0.7647	0.7777	0.6779	0.8184	0.7187

The accuracy did not enhanced according the model complexity so we will neglect this model.

4.1.4 Trial 4

I want to check if there is a noisy feature or redundant feature the result was so hug so I stored it in csv file to choose the best model.

After this trial we found that the best models which have the maximum F1 score are:

- LightGBM which matches the first notice without hypertension or alco column.
- Gradient Boosting when active col is dropped.

Dropped Feature	Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	Train F1	Test F1
hypertension	LightGBM	0.7505	0.7384	0.7725	0.7572	0.7128	0.7046	0.7414	0.7299

Dropped Feature	Model	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall	Train F1	Test F1
active	Gradient Boosting	0.739	0.7372	0.7604	0.755	0.7008	0.7053	0.7294	0.7293
alco	LightGBM	0.7501	0.7385	0.7728	0.7592	0.711	0.7012	0.7406	0.7291

4.1.5 Trial 5

Choosing the best parameters for LightGBM the results where as follow: (all feature included)

Best Parameters of the model:

📈 Model Performance:

- **Best CV Score:** 0.7377
- **Precision:** 0.7567
- **Recall:** 0.7027
- **F1 Score:** 0.7287

📋 Classification Report:

Class	Precision	Recall	F1-Score	Support
0	0.72	0.77	0.75	6892
1	0.76	0.70	0.73	6942

Accuracy			**0.74**	13834
Macro Avg	0.74	0.74	0.74	13834
Weighted Avg	0.74	0.74	0.74	13834

4.1.6 Trial 6

It is required to try model Gradient boosting classifier (with active and without active feature)

They have the same result.

📈 Model Performance:

Metric	Value
Precision	0.7549
Recall	0.7040
F1 Score	0.7285

📋 Classification Report:

Class	Precision	Recall	F1-Score	Support
0	0.72	0.77	0.74	6892
1	0.75	0.70	0.73	6942
Accuracy			**0.74**	13834
Macro Avg	0.74	0.74	0.74	13834
Weighted Avg	0.74	0.74	0.74	13834

4.1.7 Trial 7

Trained LightGBM model without alco column. the results as attached.

📈 Model Performance:

Metric	Value
Best CV Score	0.7376
Precision	0.7568
Recall	0.7028
F1 Score	0.7288

📋 Classification Report:

Class	Precision	Recall	F1-Score	Support
0	0.72	0.77	0.75	6892
1	0.76	0.70	0.73	6942
Accuracy			**0.74**	13834
Macro Avg	0.74	0.74	0.74	13834
Weighted Avg	0.74	0.74	0.74	13834

We will choose this model as by logic the most of data is for normal people so by logic recall for normal people (cardio=0) is more than recall for patient people (cardio=1).

it is used for optimization **Randomsearch.**

5. MileStone 4

5.1 mlflow for tracking models

mlflow was applied on the code and the code attached in Milestone4.

this is snapshot of tracing which we have done for models.

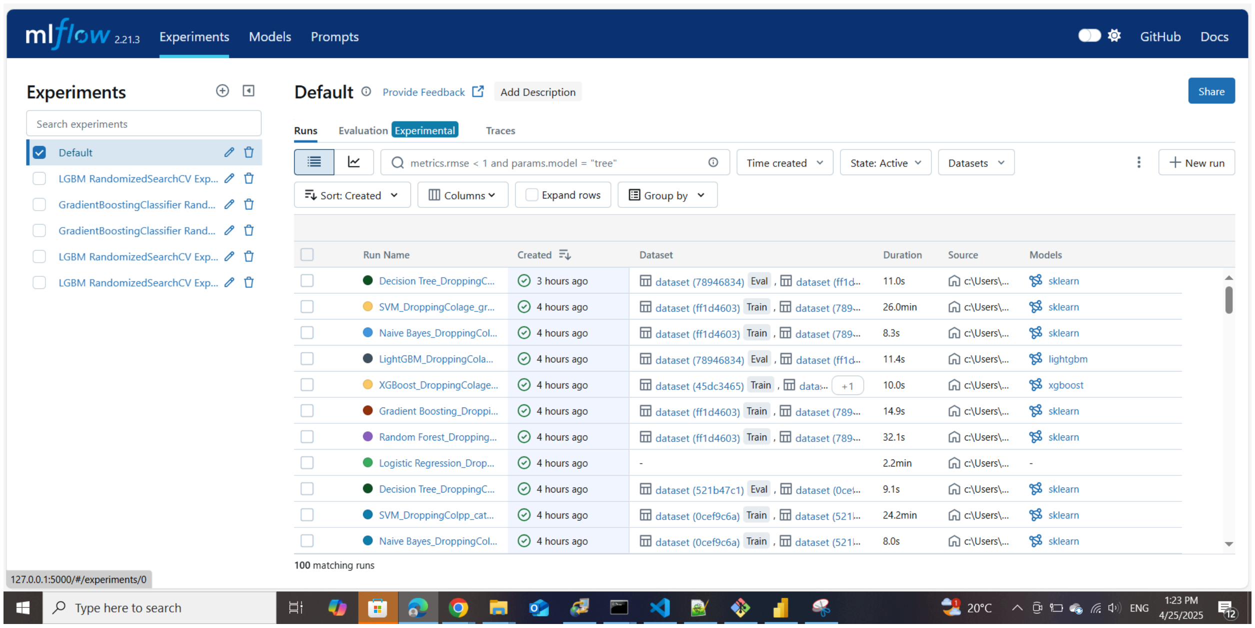
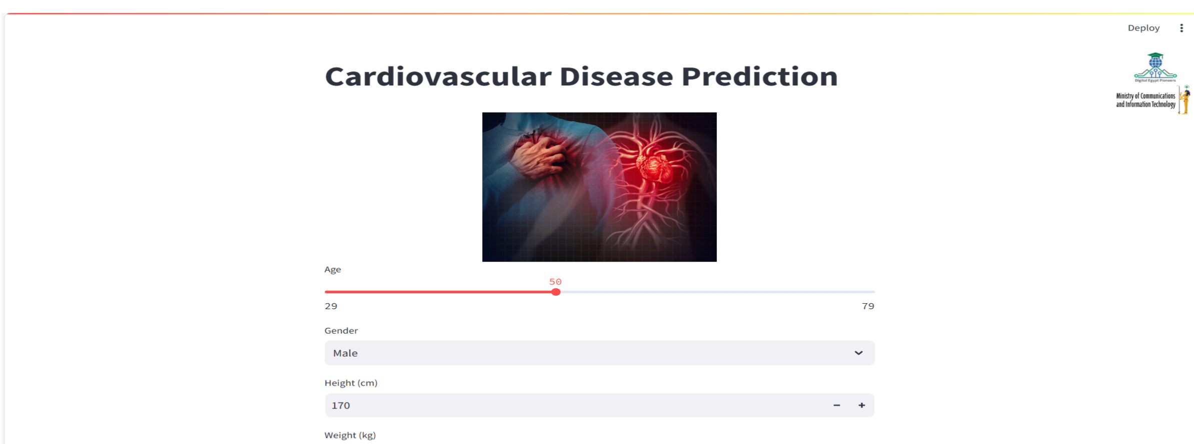


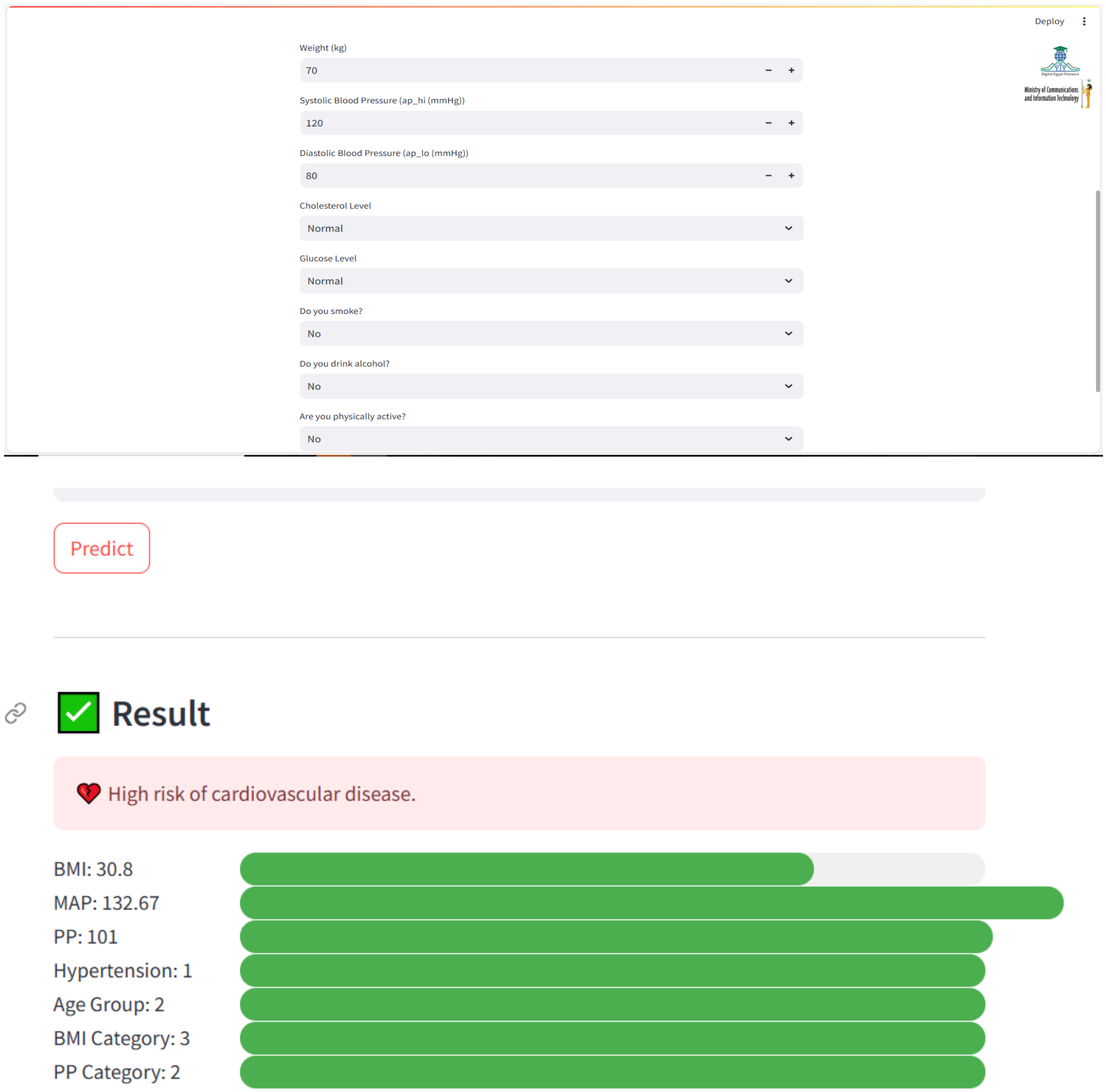
Figure 44 mlflow to track the code.

5.2 Demplyment

When I run streamlit run main_func.py in command prompt this window will appear and takes inputs from the user then predict as follow.

Noting that the model was deployed as pkl file, standardization as pkl file.





Deploy
Weight (kg)
70
Systolic Blood Pressure (ap_hi (mmHg))
120
Diastolic Blood Pressure (ap_lo (mmHg))
80
Cholesterol Level
Normal
Glucose Level
Normal
Do you smoke?
No
Do you drink alcohol?
No
Are you physically active?
No
Predict
✅ Result
💔 High risk of cardiovascular disease.
BMI: 30.8
MAP: 132.67
PP: 101
Hypertension: 1
Age Group: 2
BMI Category: 3
PP Category: 2