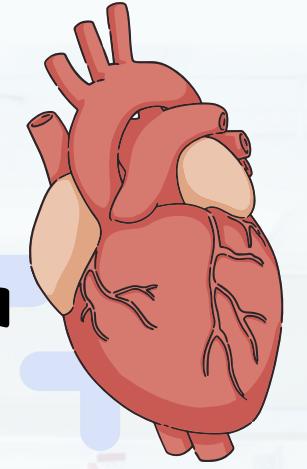


# Cardiovascular Disease Prediction

การทำนายการเกิดโรคหัวใจและหลอดเลือด



# MEMBER

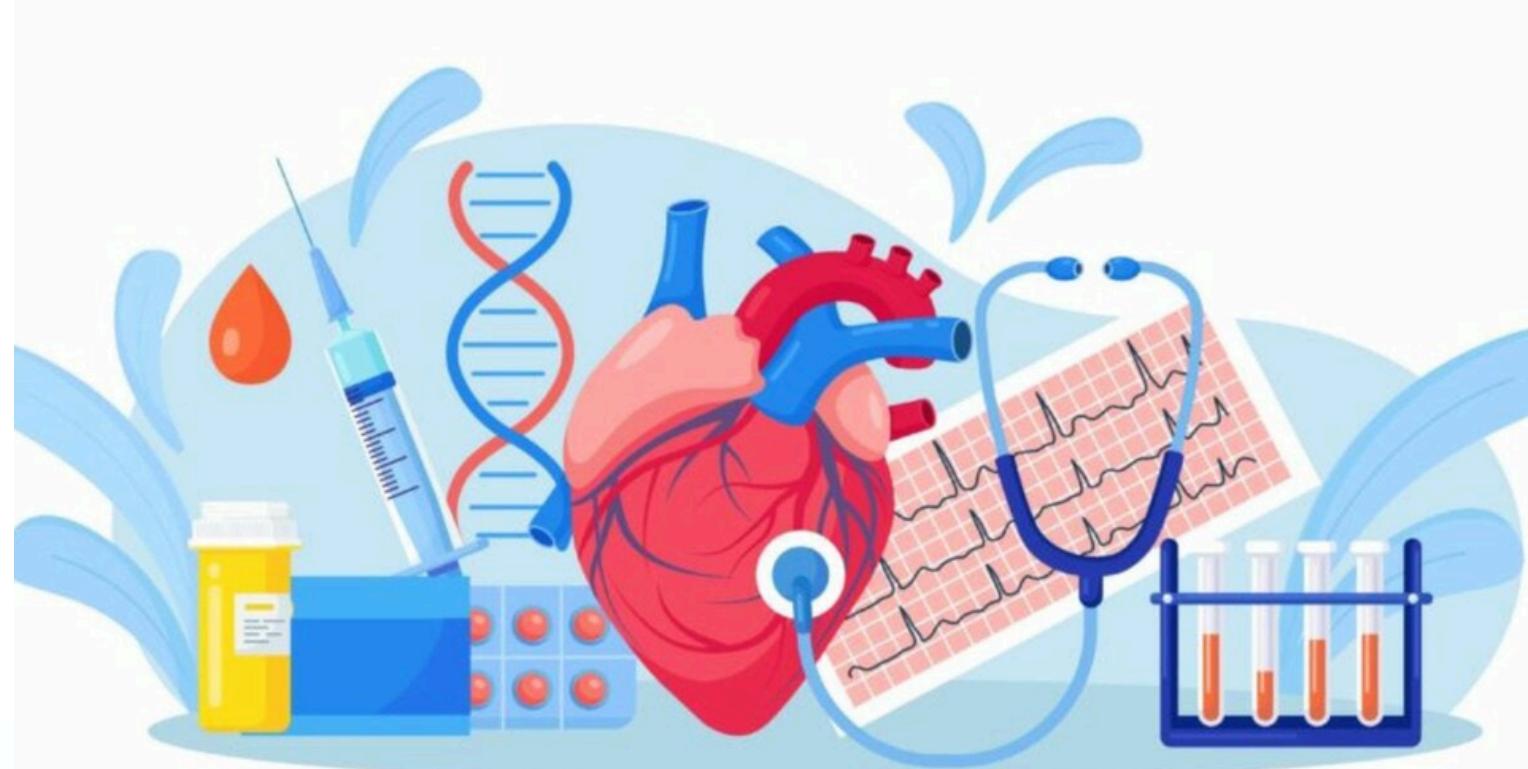
- 01 Aekkarat chaisong
- 02 Pipatsorn Midjaroenrad
- 03 Sroithongthae Auinok

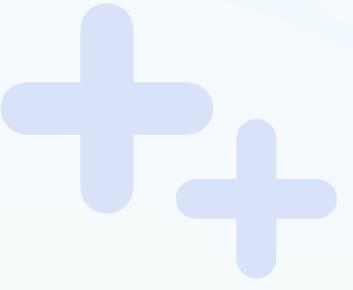


# Cardiovascular disease คือ ??

## โรคหัวใจและหลอดเลือด

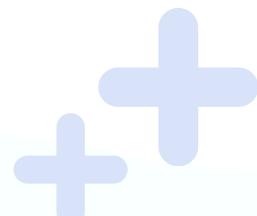
โรคที่เกี่ยวกับระบบหัวใจและหลอดเลือดในร่างกายเรียกว่า โรคหัวใจและหลอดเลือด (Cardiovascular disease) หรือที่มักจะเรียกโดยสั้นๆ ว่า "โรคหัวใจ" (Heart disease) นับเป็นหนึ่งในโรคที่เป็นสาเหตุการเสียชีวิตมากที่สุดในโลก โรคหัวใจและหลอดเลือดมักจะเกิดจากการสะสมของไขมันและสารต่างๆ ในผนังของหลอดเลือด (เรียกว่า เส้นเลือด) ซึ่งทำให้เส้นเลือดตีบและลดการไหลเวียนของเลือดไปยังหัวใจหรือส่วนต่างๆ ของร่างกาย





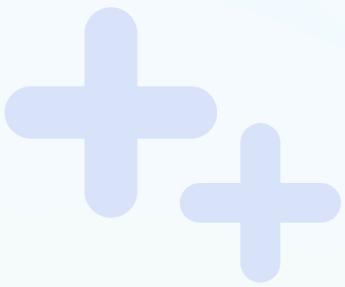
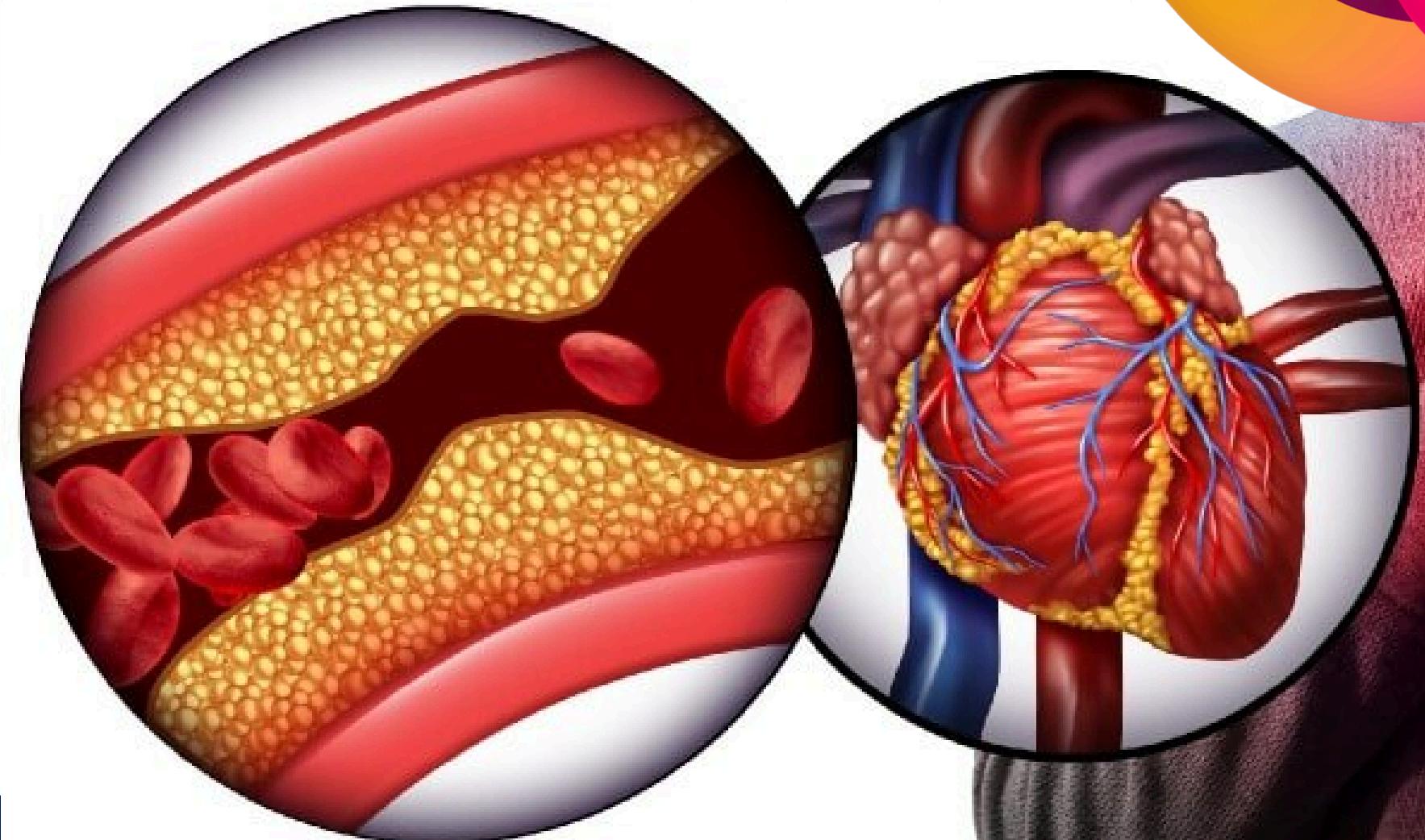
# ที่มาของโรคหัวใจ

ในปัจจุบันโรคหัวใจและหลอดเลือดเป็นสาเหตุหลักของการเสียชีวิตมีคนหลายคนที่เสียชีวิตจากโรคหัวใจและหลอดเลือดมากกว่าโรคอื่น ๆ กลุ่มของเราริบอยาศาสิกษาเรื่องโรคหัวใจและหลอดเลือดเพื่อช่วยให้ทราบถึงวิธีป้องกันและ หลีกเลี่ยงพฤติกรรม ที่ทำให้เสี่ยงเกิดโรคนี้ได้



# วัตถุประสงค์

- เพื่อศึกษาว่าปัจจัยหรือโรคต่างๆ มีผลที่ทำให้เกิดโรคหัวใจหรือไม่
- เพื่อคำนวณความเสี่ยงโรคหัวใจและหลอดเลือด
- เพื่อหาว่าปัจจัยใดที่จะส่งผลให้เป็นโรคหัวใจมากที่สุด
- เพื่อศึกษาว่าพฤติกรรมการใช้ชีวิตประจำวันนี้ ทำให้เป็นโรคหัวใจหรือไม่



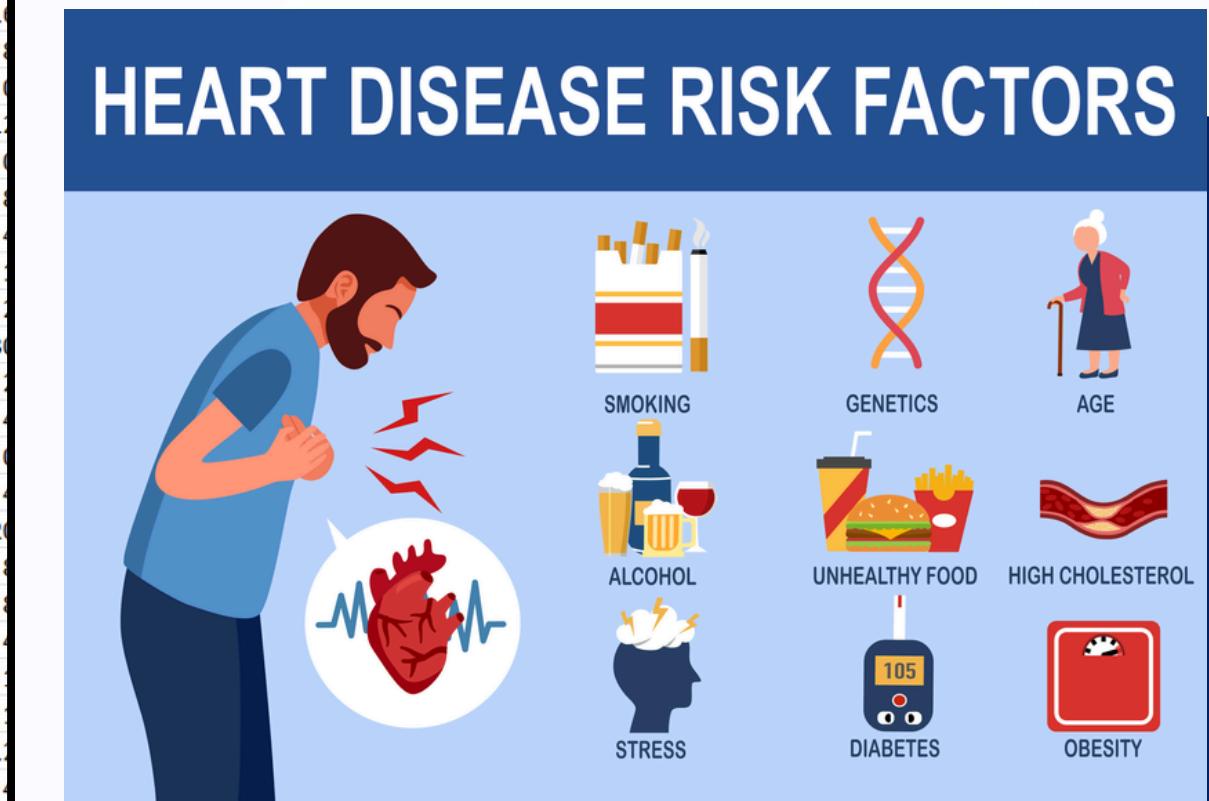
# ข้อมูลที่นำมาใช้

ข้อมูลที่ใช้ในการวิจัยในครั้งนี้คือข้อมูลจากเว็บไซต์ Kaggle.com

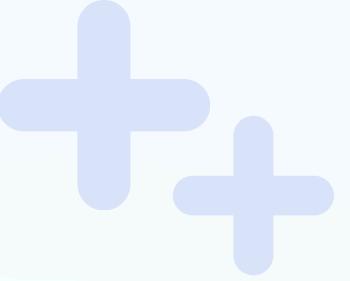
ชุดข้อมูลชื่อ Cardiovascular Diseases Risk Prediction Dataset โดยชุดข้อมูลนี้จะรวบรวมข้อมูลและปัจจัยที่มีผลต่อการเกิดโรคหัวใจและ หลอดเลือด มีจำนวนประมาณ 300000 row และ 20 Col

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	General_H	Checkup	Exercise	Heart_Dis	Skin_Canc	Other_Car	Depressio	Diabetes	Arthritis	Sex	Age_Categ	Height_(cm)	Weight_(kg)	BMI	Smoking_F	Alcohol_C	Fruit_Con	Green_Veg	FriedPotat
2	Poor	Within the No	No	No	No	No	No	Yes	Female	70-74	150	32.66	14.54	Yes	0	30	16	12	
3	Very Good	Within the No	Yes	No	No	No	Yes	No	Female	70-74	165	77.11	28.29	No	0	30	0	12	
4	Very Good	Within the Yes	No	No	No	No	Yes	No	Female	60-64	163	88.45	33.47	No	4	12	3	12	
5	Poor	Within the Yes	Yes	No	No	No	Yes	No	Male	75-79	180	93.44	28.73	No	0	30	30	12	
6	Good	Within the No	No	No	No	No	No	No	Male	80+	191	88.45	24.37	Yes	0	8	4	12	
7	Good	Within the No	No	No	No	Yes	No	Yes	Male	60-64	183	154.22	46.11	No	0	12	12	12	
8	Fair	Within the Yes	Yes	No	No	No	No	Yes	Male	60-64	175	69.85	22.74	Yes	0	16	8	12	
9	Good	Within the Yes	No	No	No	No	Yes	Female	65-69	165	108.86	39.94	Yes	3	30	8	12	12	
10	Fair	Within the No	No	No	No	Yes	No	No	Female	65-69	163	72.57	27.46	Yes	0	12	12	12	
11	Fair	Within the No	No	No	No	No	Yes	Yes	Female	70-74	163	91.63	34.67	No	0	12	12	12	
12	Fair	Within the Yes	Yes	No	No	No	No	Yes	Female	75-79	160	74.84	29.23	No	0	30	20	12	
13	Fair	Within the No	Yes	Yes	No	No	Yes	No	Male	75-79	175	73.48	23.92	No	0	2	8	30	
14	Very Good	Within the No	No	No	No	Yes	No	No	Female	50-54	168	83.91	29.86	No	8	8	0	12	
15	Fair	Within the No	No	Yes	No	No	No	No	Male	65-69	178	113.4	35.87	Yes	4	2	3	12	
16	Excellent	Within the Yes	No	No	No	No	No	No	Female	70-74	152	52.16	22.46	No	0	30	4	12	
17	Fair	Within the No	No	No	No	No	Yes	Yes	Female	70-74	163	116.12	43.94	No	0	8	8	12	
18	Good	Within the No	No	No	No	No	No	No	Male	80+	183	99.79	29.84	No	0	1	4	20	
19	Very Good	Within the Yes	No	No	No	No	No	Yes	Male	80+	168	81.65	29.05	No	30	30	12	12	
20	Fair	Within the Yes	No	No	No	No	Yes	No	Male	45-49	178	104.33	33	Yes	2	16	12	12	
21	Good	Within the No	No	Yes	Yes	No	Yes	Yes	Female	70-74	163	79.38	30.04	No	0	12	8	12	
22	Very Good	Within the Yes	No	No	No	Yes	No	No	Female	18-24	157	55.79	22.5	No	0	60	30	12	
23	Fair	Within the No	No	No	Yes	No	Yes	Yes	Male	80+	175	81.65	26.58	No	0	0	1	12	
24	Very Good	5 or more	Yes	No	No	No	No	No	Female	30-34	180	124.74	38.35	No	0	2	4	12	
25	Good	Within the No	No	No	No	No	Yes	No	Female	55-59	163	81.19	30.72	Yes	0	30	20	12	
26	Good	Within the No	No	Yes	Yes	No	No	Yes	Female	80+	160	70.31	27.46	Yes	12	8	16	12	
27	Good	Within the Yes	No	No	Yes	No	No	Yes	Female	70-74	160	88.45	34.54	Yes	0	8	4	12	

ตัวอย่าง  
ข้อมูล



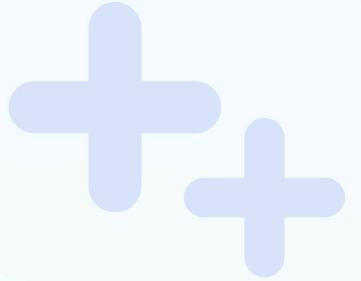
# ขั้นตอนการทำโครงการ



- 01 Import Data**
- 02 Explore & Replace Data**
- 03 EDA**
- 04 Model**

# Import Data

## import data cardiovascular (data form Kaggle)



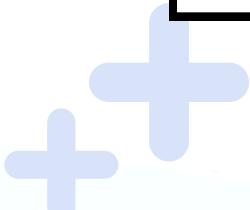
```
[2] !pip install gdown
Requirement already satisfied: gdown in /usr/local/lib/python3.10/dist-packages (4.6.6)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from gdown) (3.12.4)
Requirement already satisfied: requests[socks] in /usr/local/lib/python3.10/dist-packages (from gdown) (2.31.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from gdown) (1.16.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from gdown) (4.66.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from gdown) (4.11.2)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->gdown) (2.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (2.0.6)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (2023.7.22)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (1.7.1)

[3] !gdown 1Z8kb0FrjqOUwqjI-hHk_kJmuMYnqJp0e
Downloading...
From: https://drive.google.com/uc?id=1Z8kb0FrjqOUwqjI-hHk\_kJmuMYnqJp0e
To: /content/CVD_cleaned.csv
100% 32.5M/32.5M [00:00<00:00, 63.2MB/s]

[4] #Read the dataset CVD
df=pd.read_csv('/content/CVD_cleaned.csv')
df.head()
```

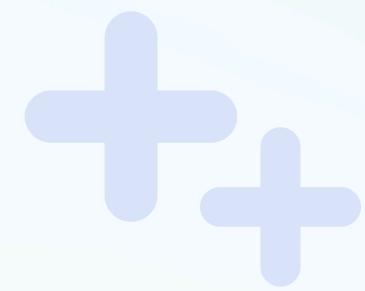
	General_Health	Checkup	Exercise	Heart_Disease	Skin_Cancer	Other_Cancer	Depression	Diabetes	Arthritis	Sex	Age_Category	Height_(cm)	Weight_(kg)	BMI	Smoking_History	Alcohol_Consumption	Fruit_Consumption	Green_Vegetables_Consumption	FriedPota
0	Poor	Within the past 2 years	No	No	No	No	No	Yes	Female	70-74	150.0	32.66	14.54	Yes	0.0	30.0		16.0	
1	Very Good	Within the past year	No	Yes	No	No	No	Yes	No	Female	70-74	165.0	77.11	28.29	No	0.0	30.0		0.0
2	Very Good	Within the past year	Yes	No	No	No	No	Yes	No	Female	60-64	163.0	88.45	33.47	No	4.0	12.0		3.0
3	Poor	Within the past year	Yes	Yes	No	No	No	Yes	No	Male	75-79	180.0	93.44	28.73	No	0.0	30.0		30.0
4	Good	Within the past year	No	No	No	No	No	No	Male	80+	191.0	88.45	24.37	Yes	0.0	8.0		4.0	

ทำการ import data ด้วย !gdown และอ่านไฟล์แบบ Csv



# Explore data & Replace Data

check type data & count of data



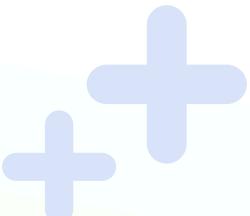
```
[6] df.describe()
```

	Height_(cm)	Weight_(kg)	BMI	Alcohol_Consumption	Fruit_Consumption	Green_Vegetables_Consumption	FriedPotato_Consumption
count	308854.000000	308854.000000	308854.000000	308854.000000	308854.000000	308854.000000	308854.000000
mean	170.615249	83.588655	28.626211	5.096366	29.835200	15.110441	6.296616
std	10.658026	21.343210	6.522323	8.199763	24.875735	14.926238	8.582954
min	91.000000	24.950000	12.020000	0.000000	0.000000	0.000000	0.000000
25%	163.000000	68.040000	24.210000	0.000000	12.000000	4.000000	2.000000
50%	170.000000	81.650000	27.440000	1.000000	30.000000	12.000000	4.000000
75%	178.000000	95.250000	31.850000	6.000000	30.000000	20.000000	8.000000
max	241.000000	293.020000	99.330000	30.000000	120.000000	128.000000	128.000000

```
[5] df.info()
print ('='*40)
df.shape
```

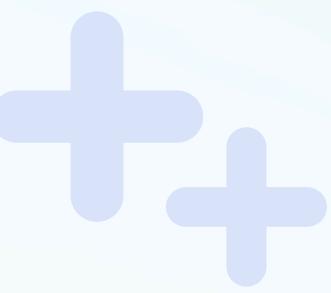
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 308854 entries, 0 to 308853
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   General_Health    308854 non-null   object  
 1   Checkup          308854 non-null   object  
 2   Exercise         308854 non-null   object  
 3   Heart_Disease    308854 non-null   object  
 4   Skin_Cancer       308854 non-null   object  
 5   Other_Cancer      308854 non-null   object  
 6   Depression        308854 non-null   object  
 7   Diabetes          308854 non-null   object  
 8   Arthritis         308854 non-null   object  
 9   Sex               308854 non-null   object  
 10  Age_Category      308854 non-null   object  
 11  Height_(cm)       308854 non-null   float64 
 12  Weight_(kg)       308854 non-null   float64 
 13  BMI               308854 non-null   float64 
 14  Smoking_History   308854 non-null   object  
 15  Alcohol_Consumption 308854 non-null   float64 
 16  Fruit_Consumption 308854 non-null   float64 
 17  Green_Vegetables_Consumption 308854 non-null   float64 
 18  FriedPotato_Consumption 308854 non-null   float64 
dtypes: float64(7), object(12)
memory usage: 44.8+ MB
=====
(308854, 19)
```

เช็ค type ของข้อมูลด้วยคำสั่ง .info  
และดูข้อมูลเบื้องต้นด้วยคำสั่ง .describe



# Explore data & Replace Data

replace data in col01



```
[0] col01 = ['Exercise','Heart_Disease','Skin_Cancer','Other_Cancer','Depression','Arthritis','Smoking_History']
```

```
[1] [9] df[col01] = df[col01].replace({  
    'Yes':1,  
    'No':0  
})
```

```
df[col01]
```

	Exercise	Heart_Disease	Skin_Cancer	Other_Cancer	Depression	Arthritis	Smoking_History
0	0	0	0	0	0	1	1
1	0	1	0	0	0	0	0
2	1	0	0	0	0	0	0
3	1	1	0	0	0	0	0
4	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...
308849	1	0	0	0	0	0	0
308850	1	0	0	0	0	0	0
308851	1	0	0	0	1	0	1
308852	1	0	0	0	0	0	0
308853	1	0	0	0	0	0	0

308854 rows × 7 columns

ทำการสร้าง col01 เพื่อเก็บข้อมูลตัวแปรต่างๆ เพื่อนำมานวนลูปและแทนค่าข้อมูลด้วย 0,1 โดยที่จะกำหนดให้ Yes = 1  
NO = 0

# Explore data & Replace Data

## replace General\_Heath and Sex Data



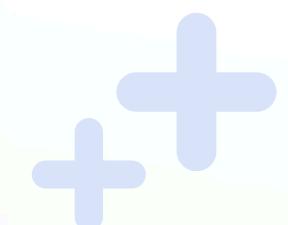
```
#replace data 'General_Health' : Very Good = 5 , Good = 4 , Excellent = 3 , Fair = 2 , Poor = 1  
df['General_Health'] = df['General_Health'].replace({  
    'Very Good':5,  
    'Good':4,  
    'Excellent':3,  
    'Fair':2,  
    'Poor':1  
})  
df['General_Health'].value_counts()
```

```
5    110395  
4    95364  
3    55954  
2    35810  
1    11331  
Name: General_Health, dtype: int64
```

```
[ ] #replace data 'Sex' : Male = 0 , Female = 1  
df['Sex'] = df['Sex'].replace({  
    'Male':0,  
    'Female':1  
})  
df['Sex'].value_counts()
```

```
1    160196  
0    148658  
Name: Sex, dtype: int64
```

ทำการแทนที่ข้อมูล General\_Health ด้วย 1-5  
ตามลำดับของสุขภาพดังภาพ  
และการแทนที่ข้อมูลเพศด้วย 0,1 โดยที่  
Male = 0 และ Female = 1



# Explore data & Replace Data



ทำการแทนที่ข้อมูลอายุ จาก80+ เป็น80-100 และทำการสร้าง Col'ใหม่และทำการแบ่งข้อมูล  
อายุเป็น Age\_Min และ Age\_max

```
[13] #replace data 'Age_Category' : 80+ = 80-100
df['Age_Category'].value_counts()
df['Age_Category'] = df['Age_Category'].replace({
    '80+':'80-100'
})
df['Age_Category'].value_counts()

65-69      33434
60-64      32418
70-74      31103
55-59      28054
50-54      25097
80-100     22271
40-44      21595
45-49      20968
75-79      20705
35-39      20606
18-24      18681
30-34      18428
25-29      15494
Name: Age_Category, dtype: int64

[14] new_columns = df['Age_Category'].str.split('-', expand=True)
new_columns.columns = ['Age_min','Age_max']
new_columns = new_columns.astype(int)
data = pd.concat([df, new_columns], axis=1)
data = data.drop(['Age_Category'], axis=1)
data

   General_Health Checkup Exercise Heart_Disease Skin_Cancer Other_Cancer Depression Diabetes Arthritis Sex Height_(cm) Weight_(kg) BMI Smoking_History Alcohol_Consumption Fruit_Consumption Green_Vegetables_Consumption FriedPotato_Consumpti
0           1       2       0       0       0       0       0       No      1     1     150.0     32.66    14.54      1         0.0      30.0          16.0        1
1           5       1       0       1       0       0       0      Yes      0     1     165.0     77.11    28.29      0         0.0      30.0          0.0        0
2           5       1       1       0       0       0       0      Yes      0     1     163.0     88.45    33.47      0         4.0      12.0          3.0        1
3           1       1       1       1       0       0       0      Yes      0     0     180.0     93.44    28.73      0         0.0      30.0          30.0       30.0
4           4       1       0       0       0       0       0      No      0     0     191.0     88.45    24.37      1         0.0      8.0           4.0        4.0
...         ...
308849      5       1       1       0       0       0       0      No      0     0     168.0     81.65    29.05      0         4.0      30.0          8.0        8.0
308850      2       3       1       0       0       0       0      Yes      0     0     180.0     69.85    21.48      0         8.0      15.0          60.0       60.0
308851      5       4       1       0       0       0       1  Yes, but female told only during pregnancy      0     1     157.0     61.23    24.69      1         4.0      40.0          8.0        8.0
```

# Explore data & Replace Data

ทำการลบข้อมูลผิดปกติออกจาก col diabetes และแทนค่าด้วย 0,1--> (diabetes\_Nab)

ทำการแทนที่ข้อมูลทั้งหมดด้วย 0,1-->(diabetes\_Aab)

```
[15] #Delete the abnormal row in diabetes
diabetes_Nab = data[~data['Diabetes'].isin(['No, pre-diabetes or borderline diabetes','Yes, but female told only during pregnancy'])]

print('Before')
display(diabetes_Nab['Diabetes'].value_counts())

# Encoding
diabetes_Nab['Diabetes'] = diabetes_Nab['Diabetes'].replace({
    'Yes':1,
    'No':0
})
diabetes_Nab['Diabetes'].value_counts()

Before
No      259141
Yes     40171
Name: Diabetes, dtype: int64
<ipython-input-15-8e636edb18e5>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
diabetes_Nab['Diabetes'] = diabetes_Nab['Diabetes'].replace({
0      259141
1      40171
Name: Diabetes, dtype: int64

[16] diabetes_Aab = data.copy()
diabetes_Aab['Diabetes'] = diabetes_Aab['Diabetes'].replace({
    'Yes, but female told only during pregnancy':1,
    'No, pre-diabetes or borderline diabetes':0,
    'Yes':1,
    'No':0
})

diabetes_Aab['Diabetes'].value_counts()

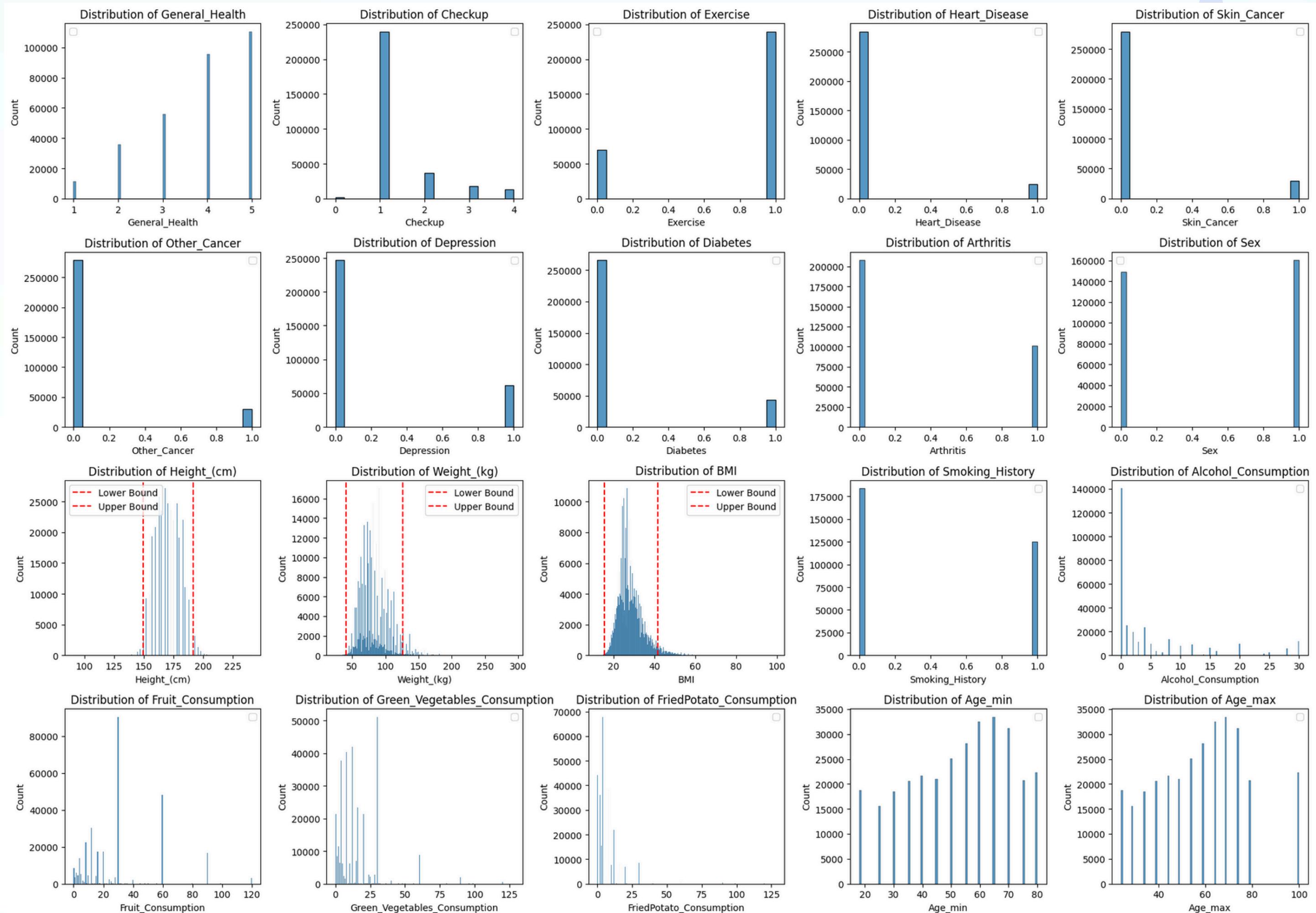
0      266037
1      42817
Name: Diabetes, dtype: int64
```

[18] diabetes\_Nab

	General_Health	Checkup	Exercise
0	1	2	
1	5	1	
2	5	1	
3	1	1	
4	4	1	
...	...	...	
308848	4	3	
308849	5	1	
308850	2	3	
308852	5	1	
308853	3	1	
299312 rows × 20 columns			
[19] diabetes_Aab			
	General_Health	Checkup	Exercise
0	1	2	0
1	5	1	0
2	5	1	1
3	1	1	1
4	4	1	0
...	...	...	...
308849	5	1	1
308850	2	3	1
308851	5	4	1
308852	5	1	1
308853	3	1	1
308854 rows × 20 columns			

# EDA

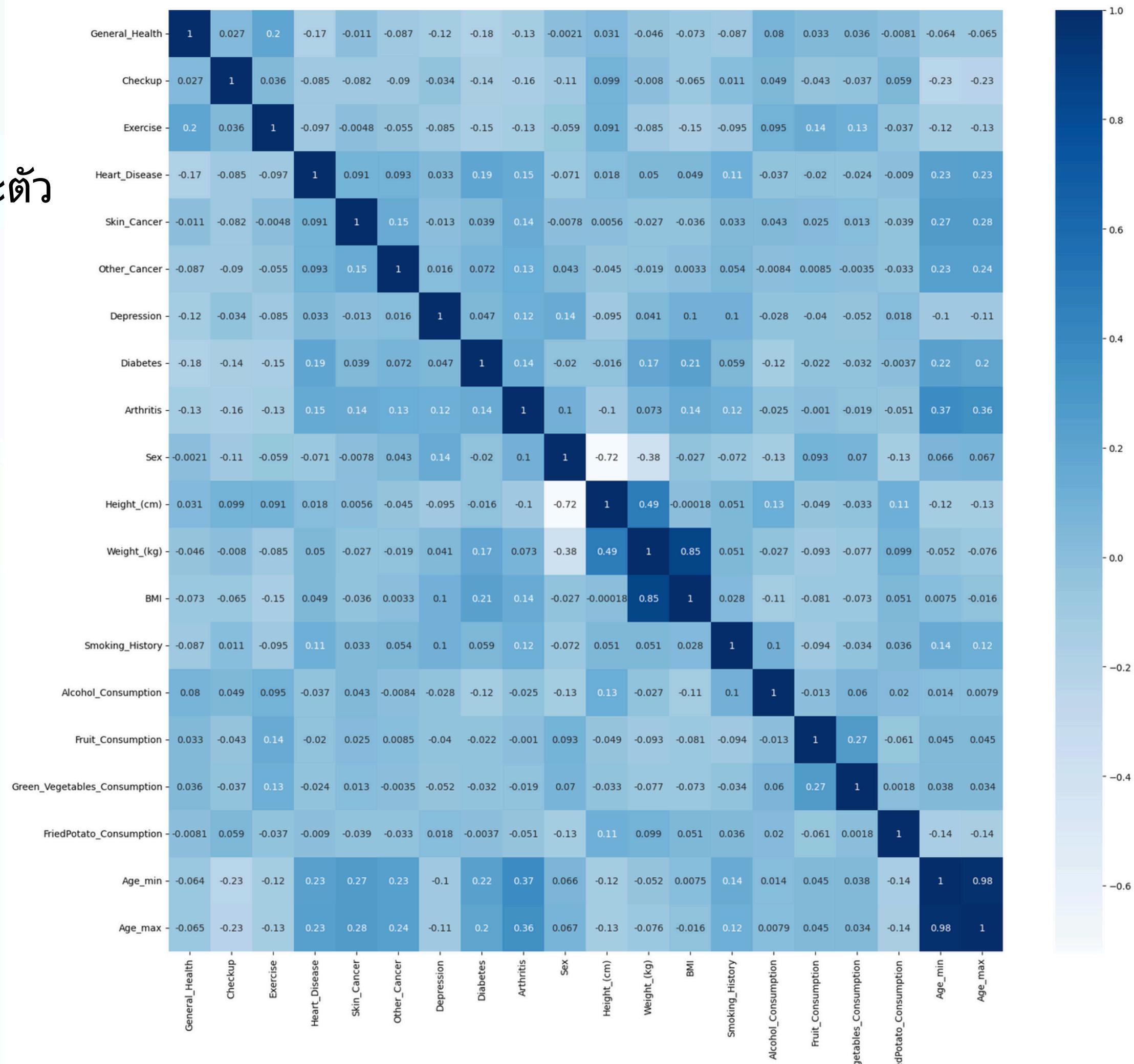
## กราฟแสดงการกระจายของข้อมูลทั้งหมด



# EDA

## Heat Maps (No Abnormal Data)

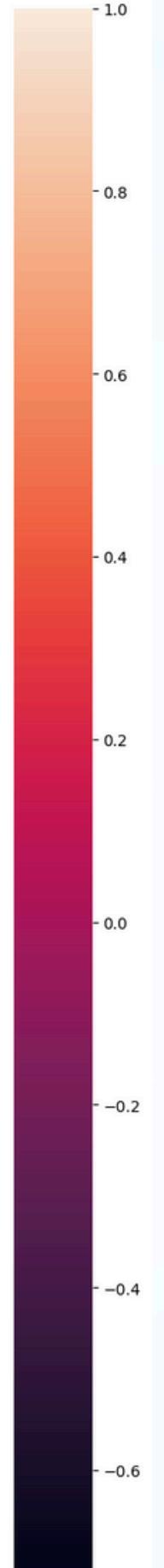
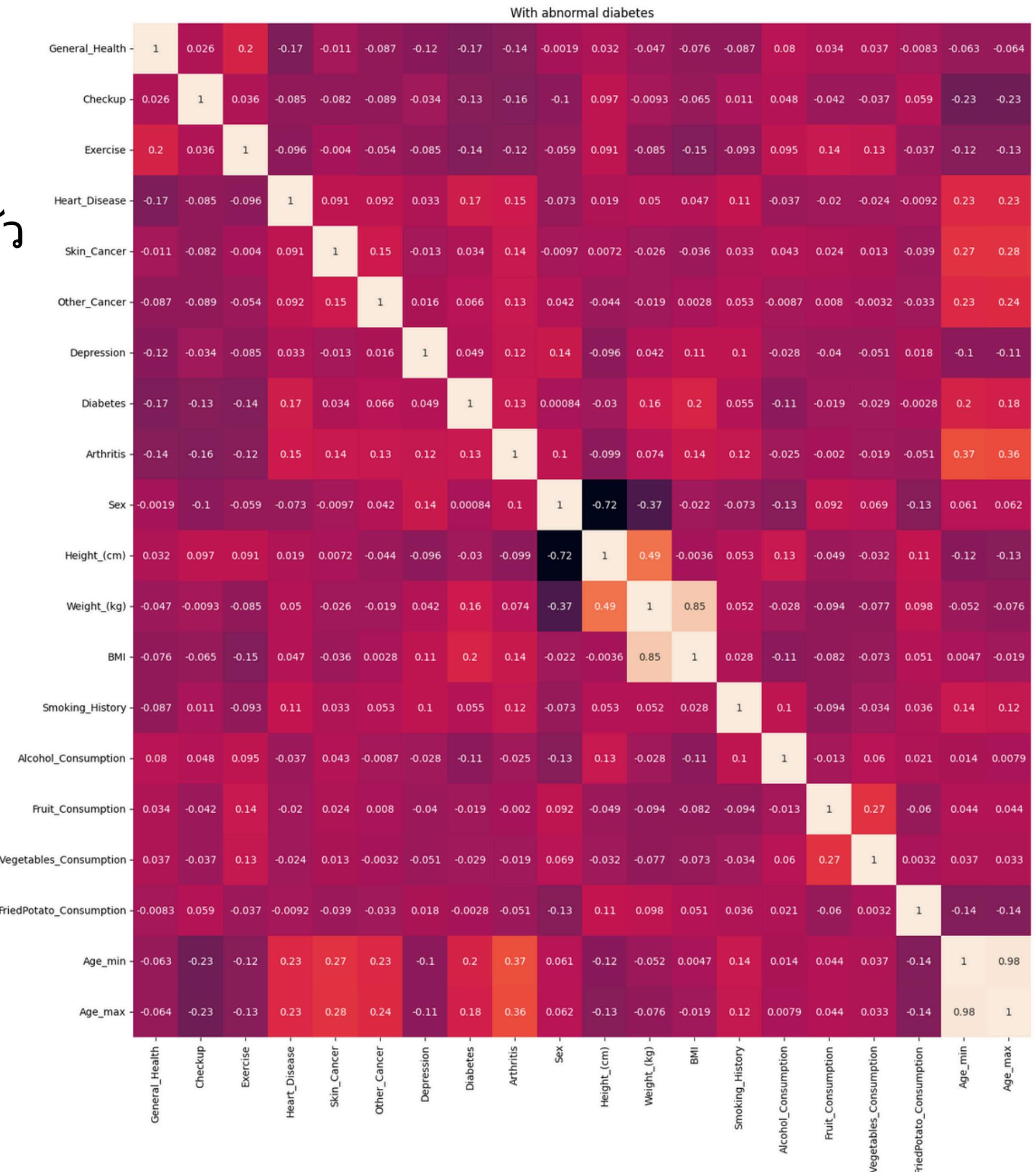
แสดงข้อมูลความสัมพันธ์ของข้อมูลแต่ละตัว  
ด้วยตัวเลข ยิ่งมีค่าความสัมพันธ์มาก



# EDA

## Heat Maps (Abnormal Data)

แสดงข้อมูลความสัมพันธ์ของข้อมูลแต่ละตัว  
ด้วยตัวเลข ยิ่งมีค่าความสัมพันธ์มาก

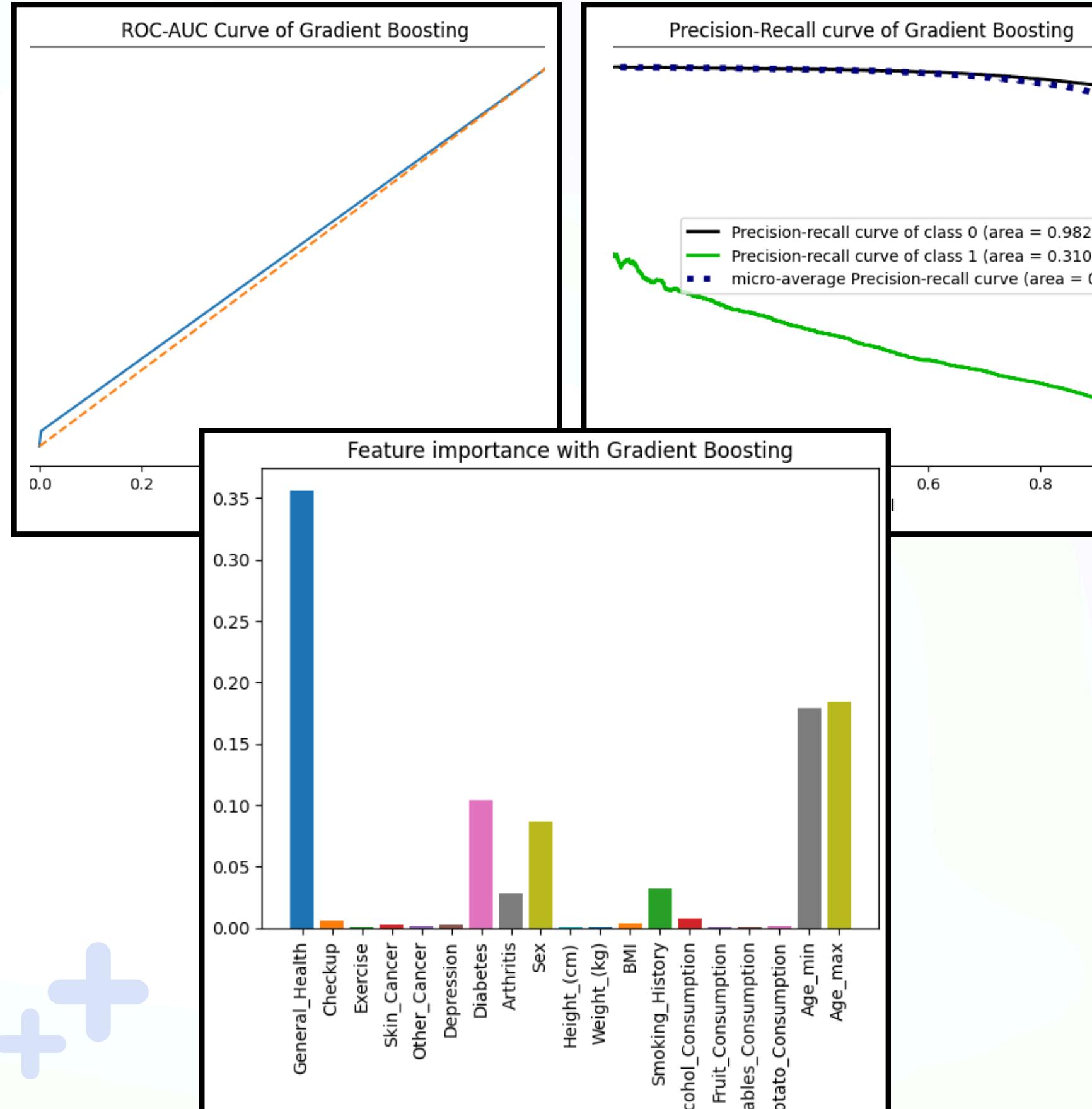


**ACC-->0.920023**

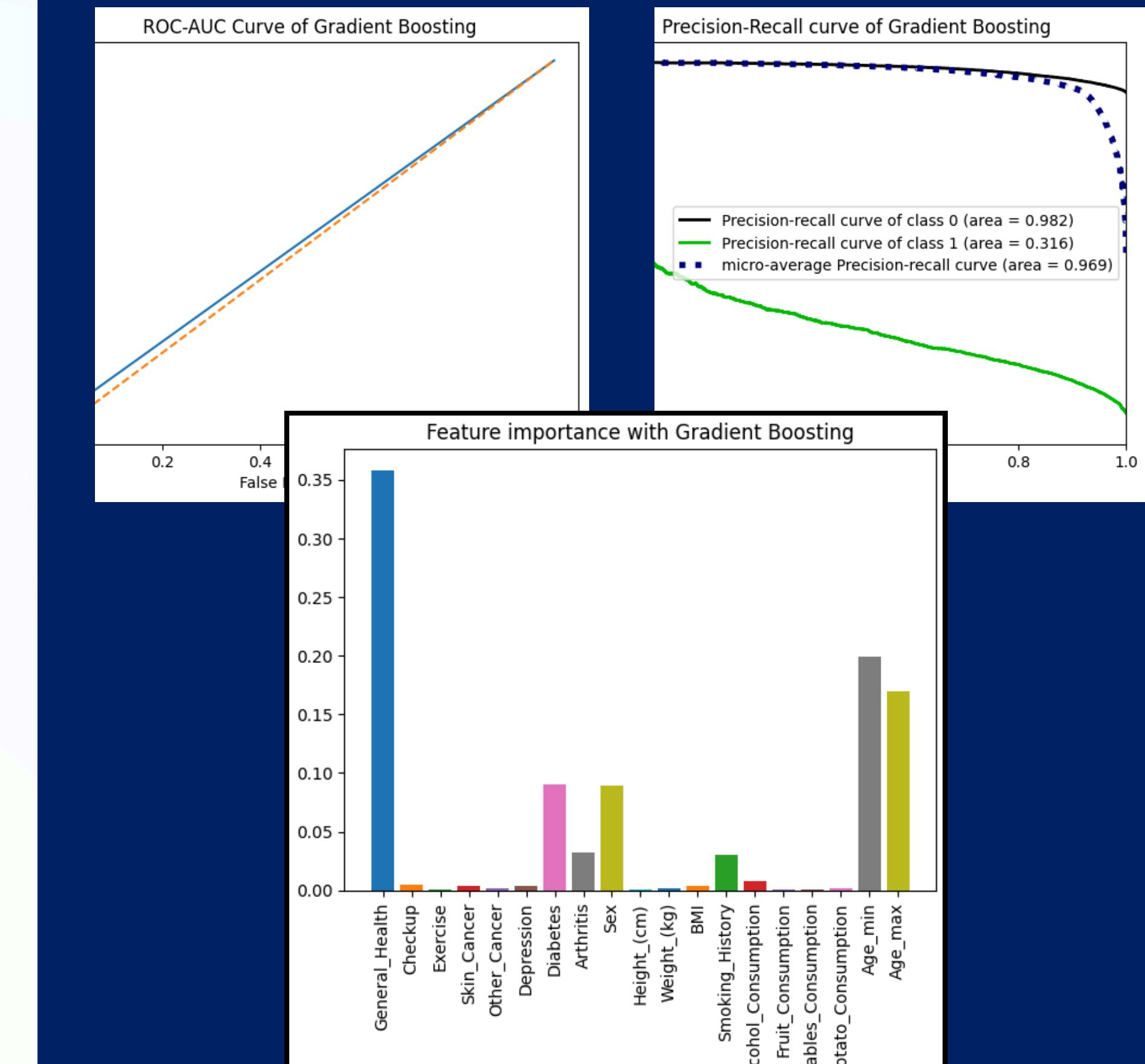
## Gradient Boosting Model

**ACC-->0.919625**

**train model with no Abnormal data**



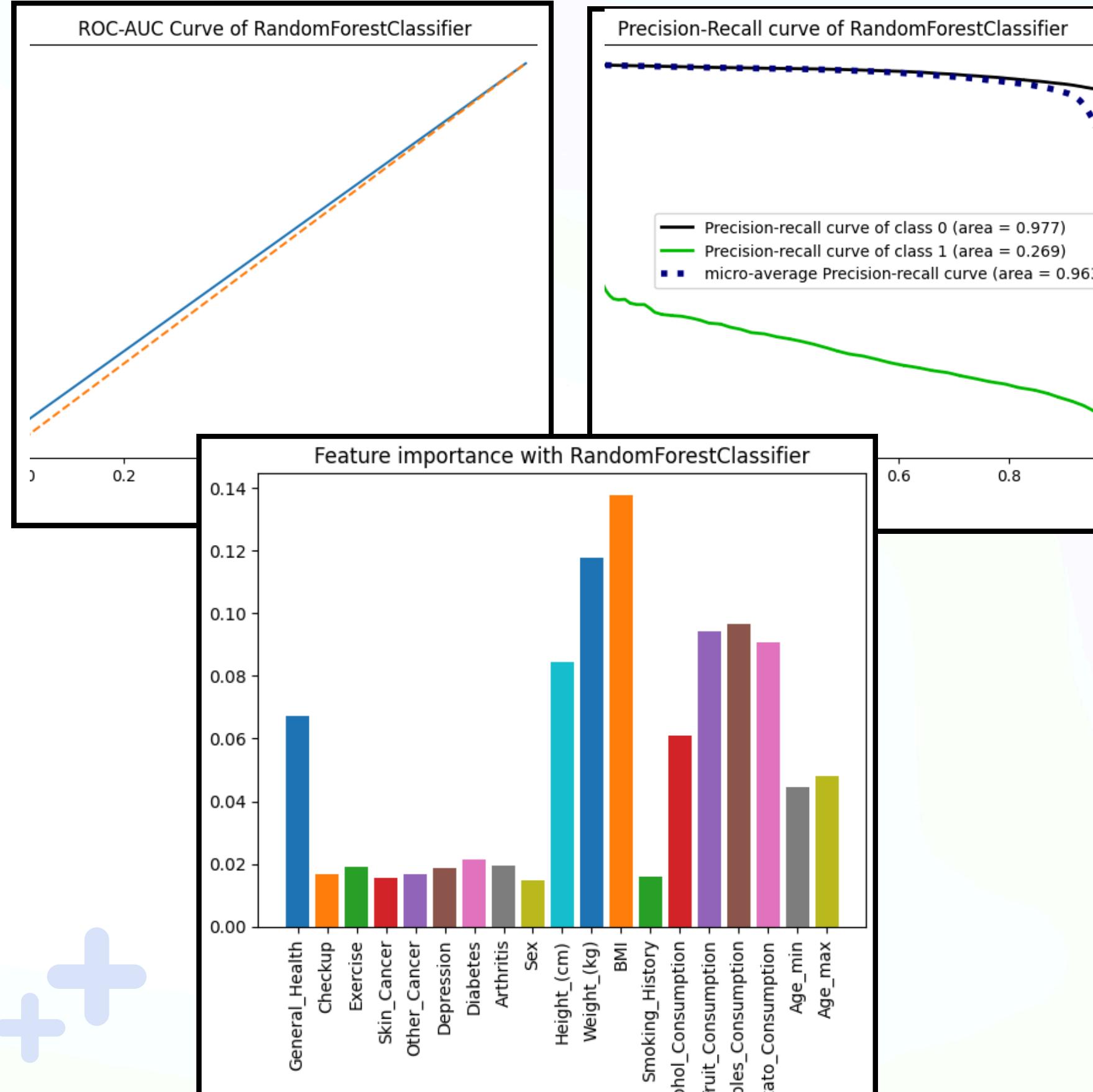
**train model with Abnormal data**



# ACC-->0.918590

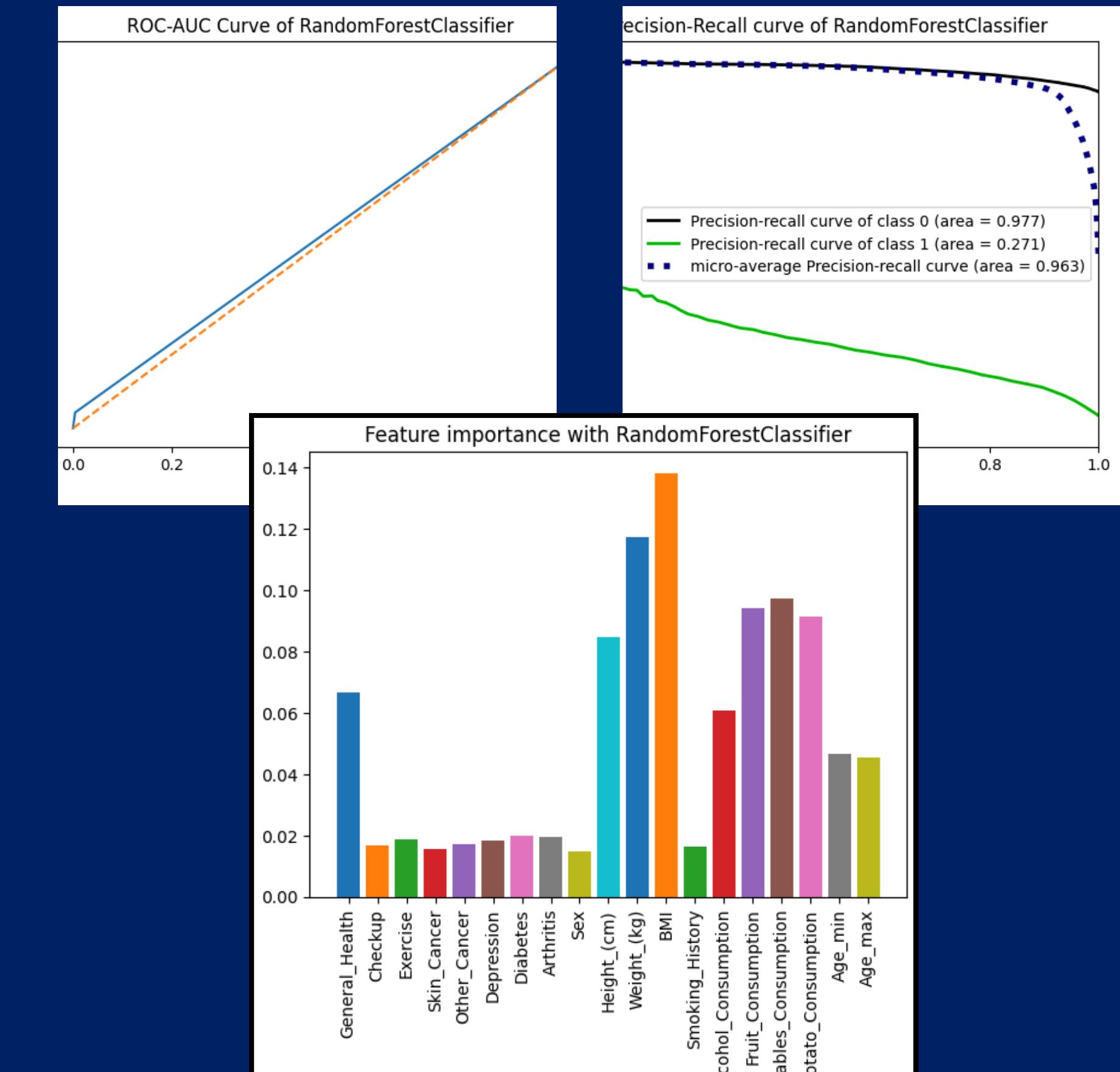
## RandomForestClassifie Model

### train model with no Abnormal data



# ACC--> 0.918327

### train model with Abnormal data

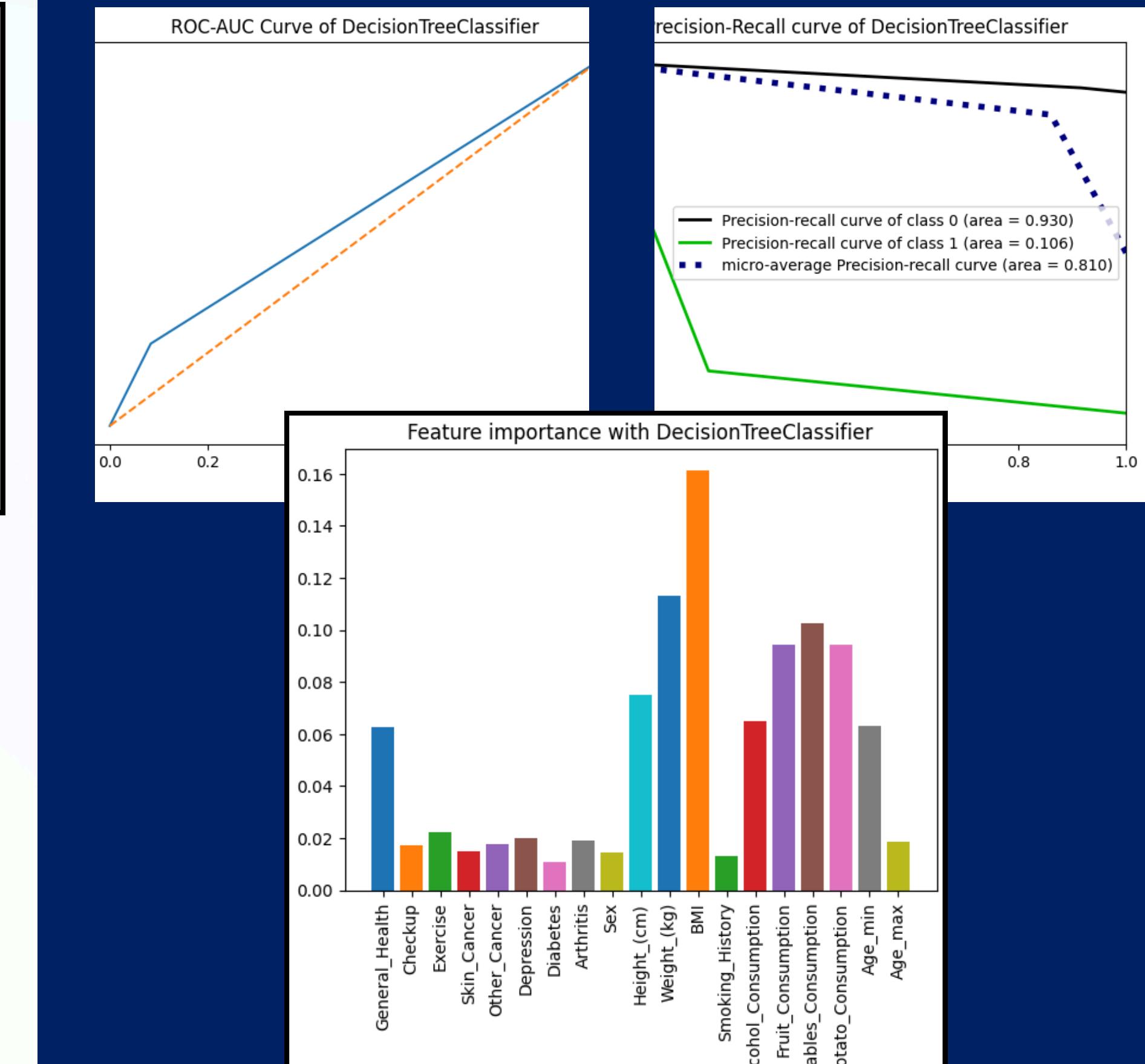
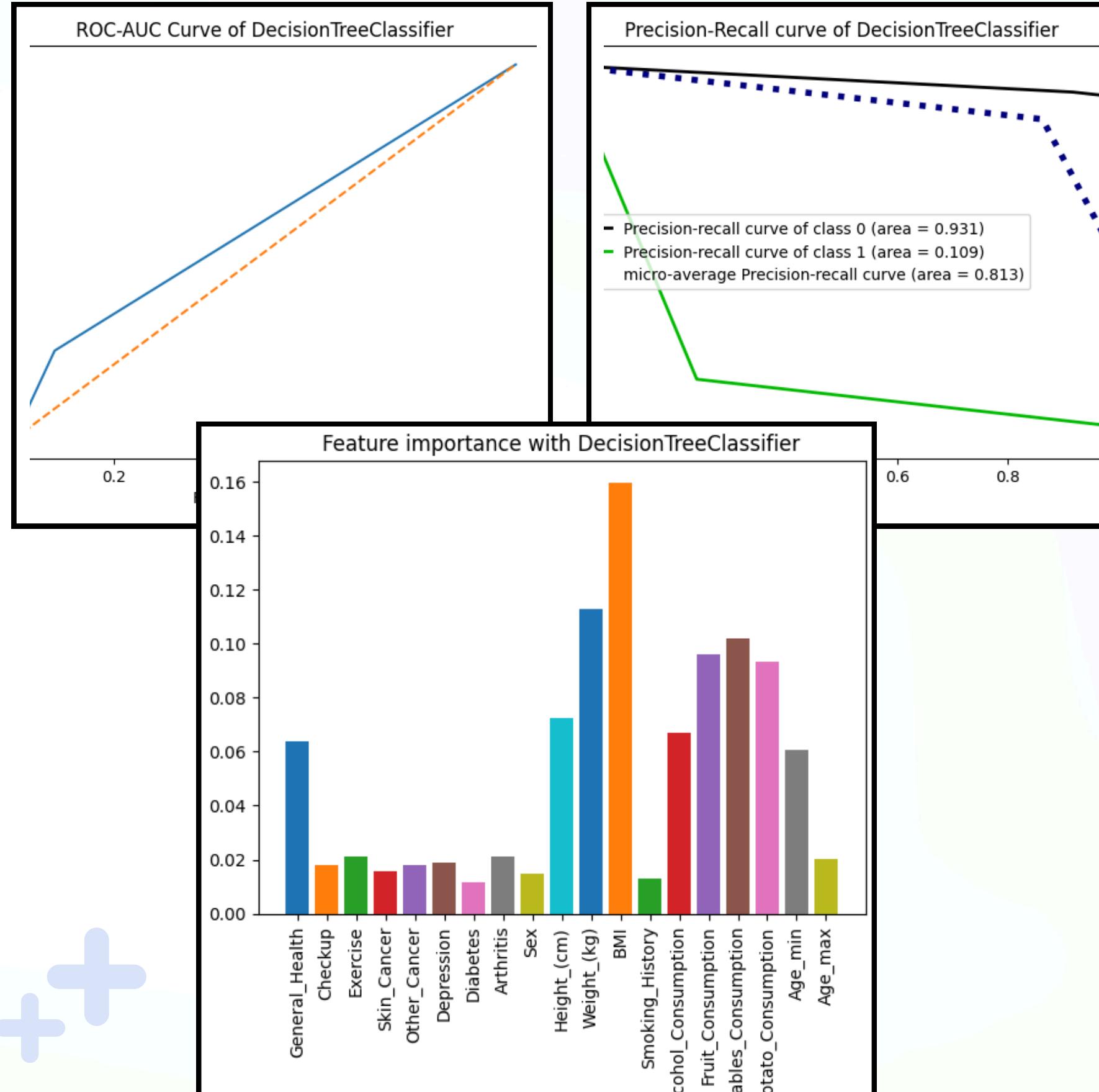


**ACC--> 0.911494**

## DecisionTreeClassifier Model

**ACC--> 0.910621**

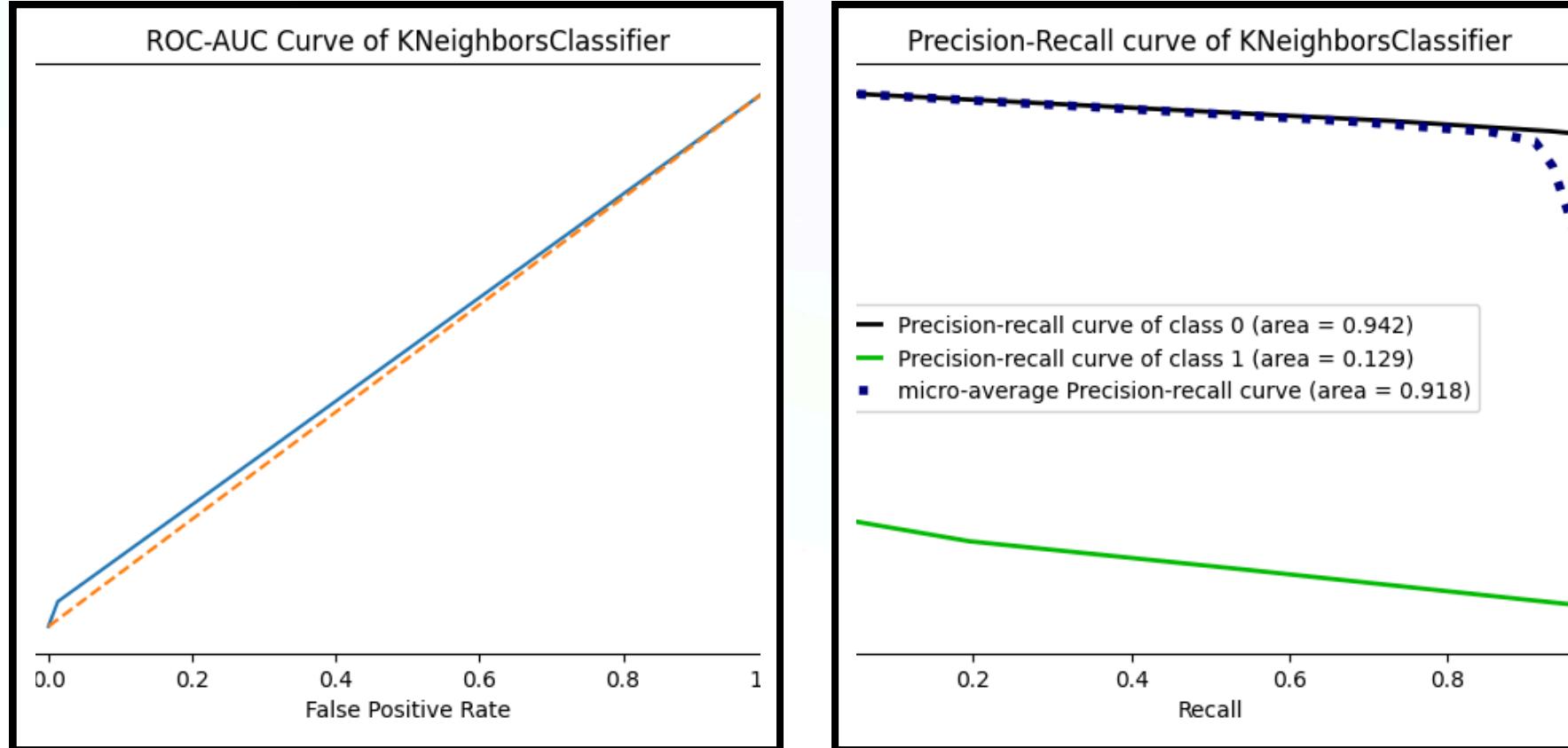
**train model with no Abnormal data**



# ACC-->0.863470

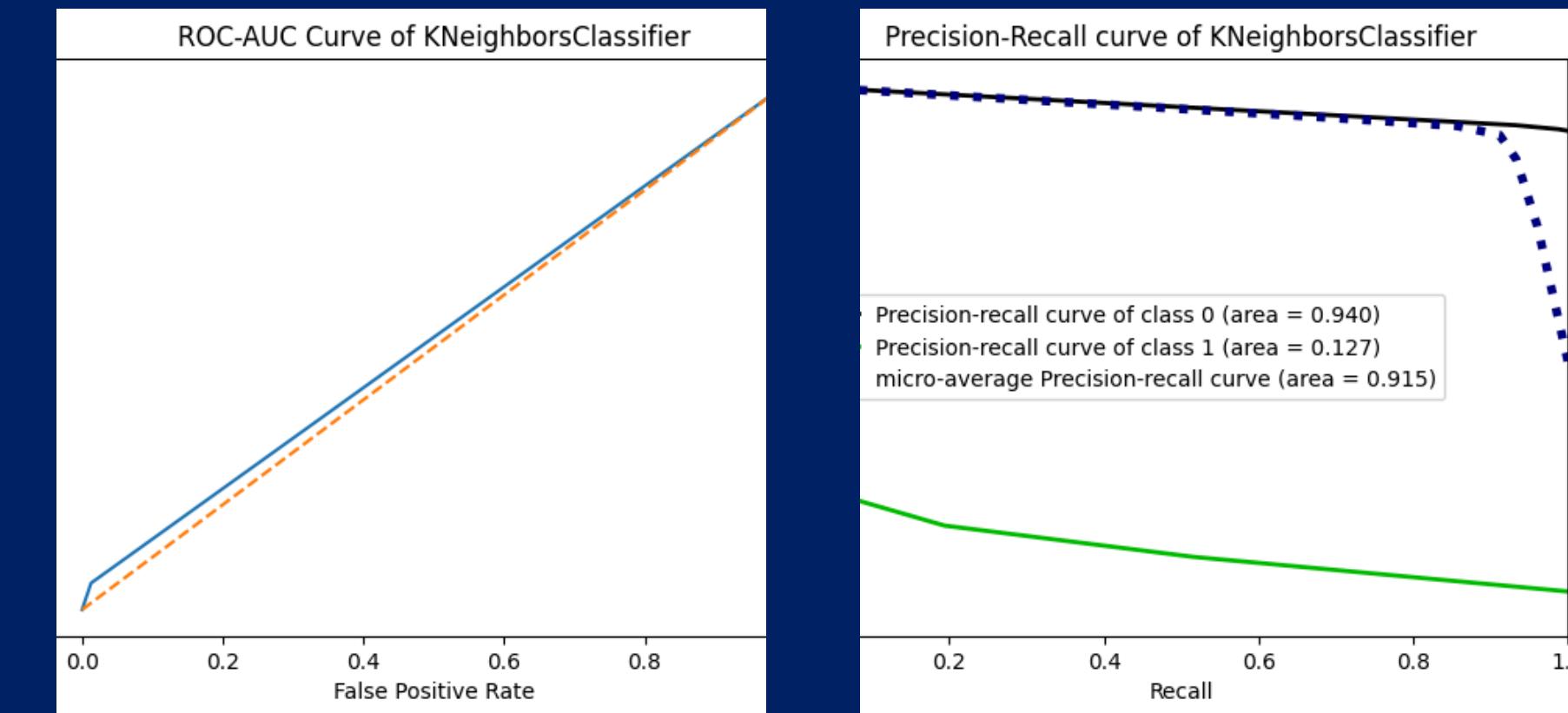
## KNeighborsClassifier Model

### train model with no Abnormal data



# ACC-->0.861909

### train model with Abnormal data



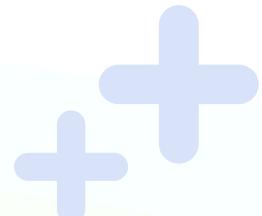
Classification report of KNeighborsClassifier :				
	precision	recall	f1-score	support
0	0.92	0.99	0.95	55046
1	0.24	0.05	0.08	4816
accuracy			0.91	59862
macro avg	0.58	0.52	0.52	59862
weighted avg	0.87	0.91	0.88	59862

	precision	recall	f1-score	support
0	0.92	0.99	0.95	56776
1	0.26	0.05	0.08	4994
accuracy			0.91	61770
macro avg	0.59	0.52	0.52	61770
weighted avg	0.87	0.91	0.88	61770

# Model Result .

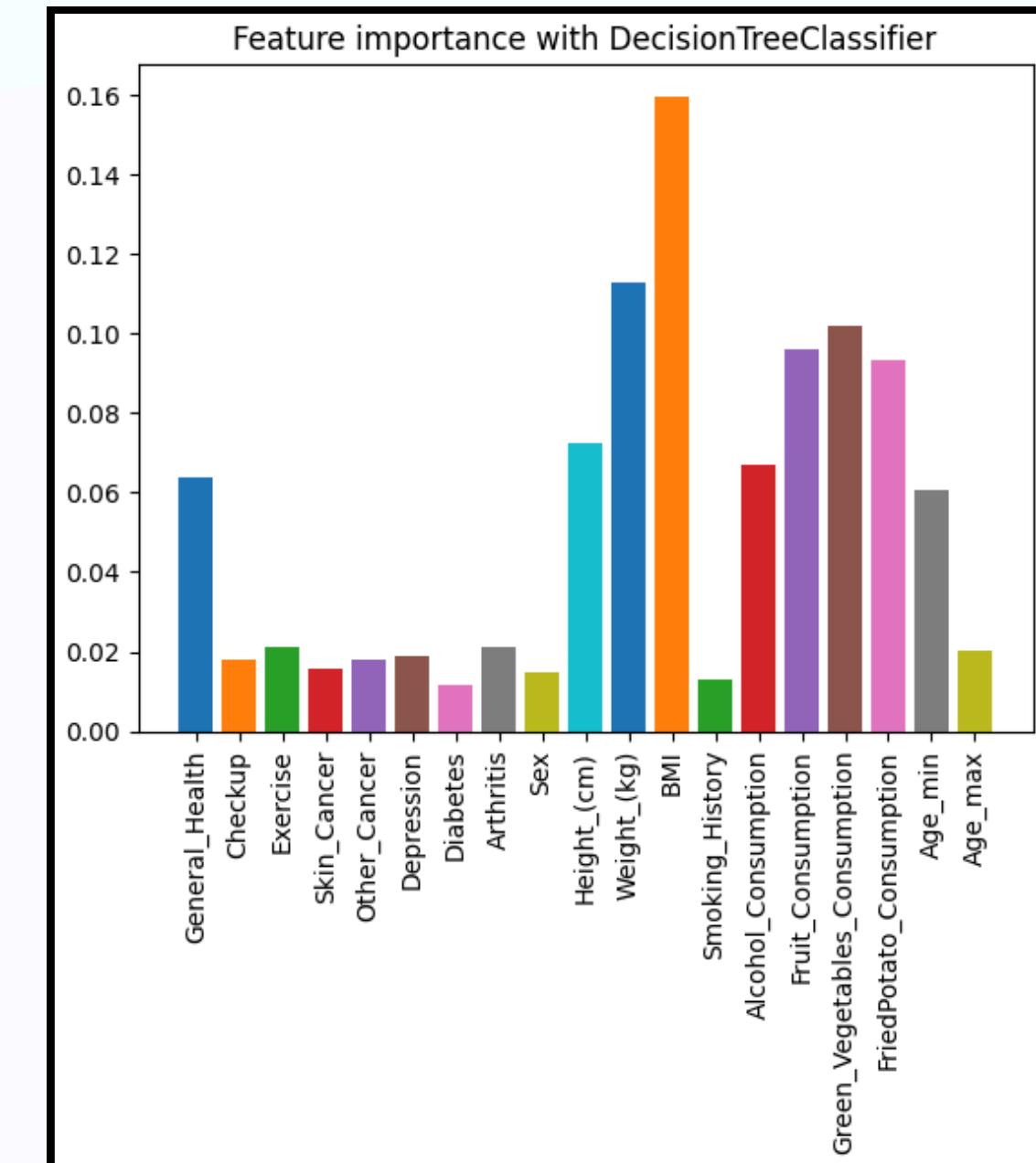
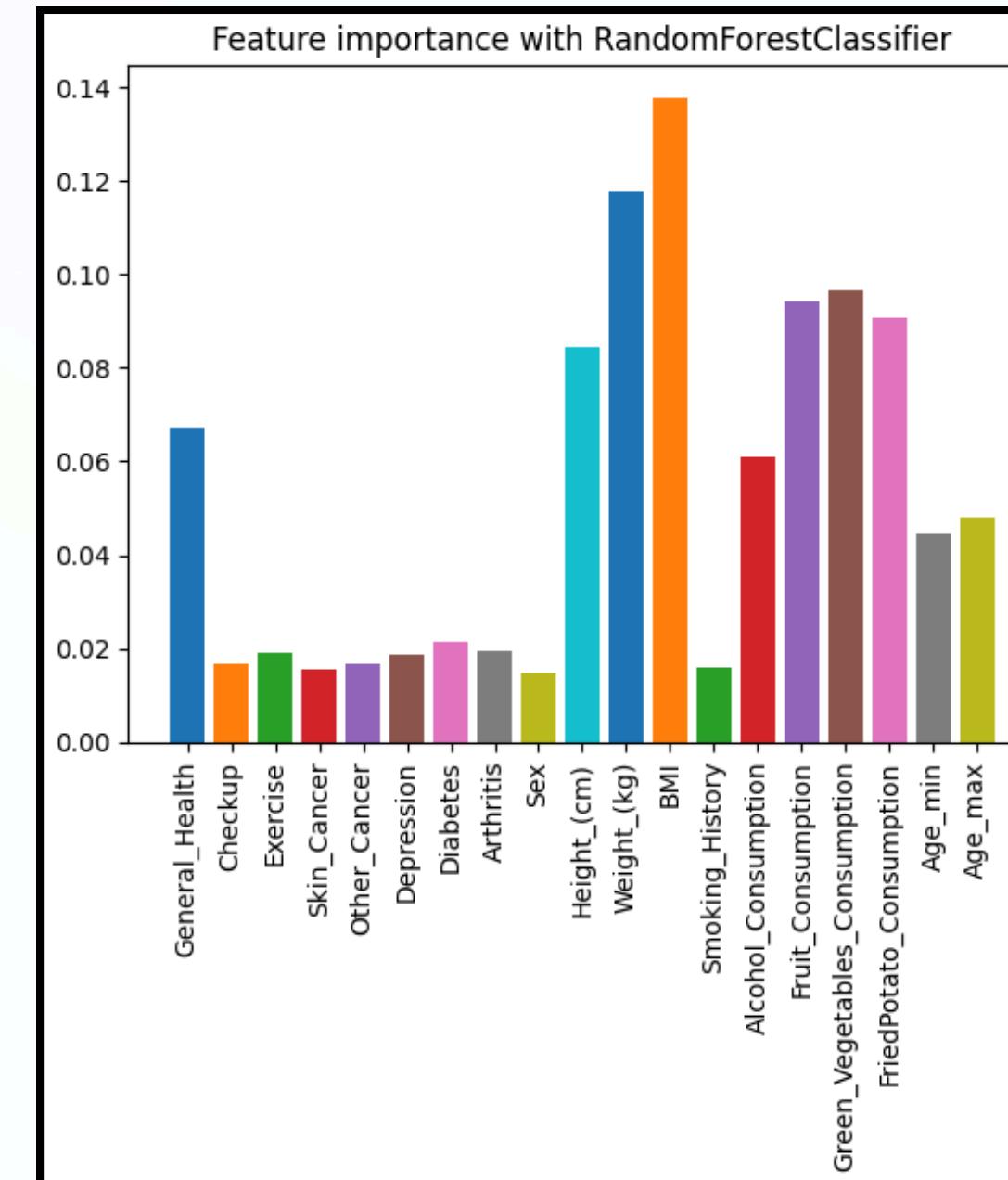
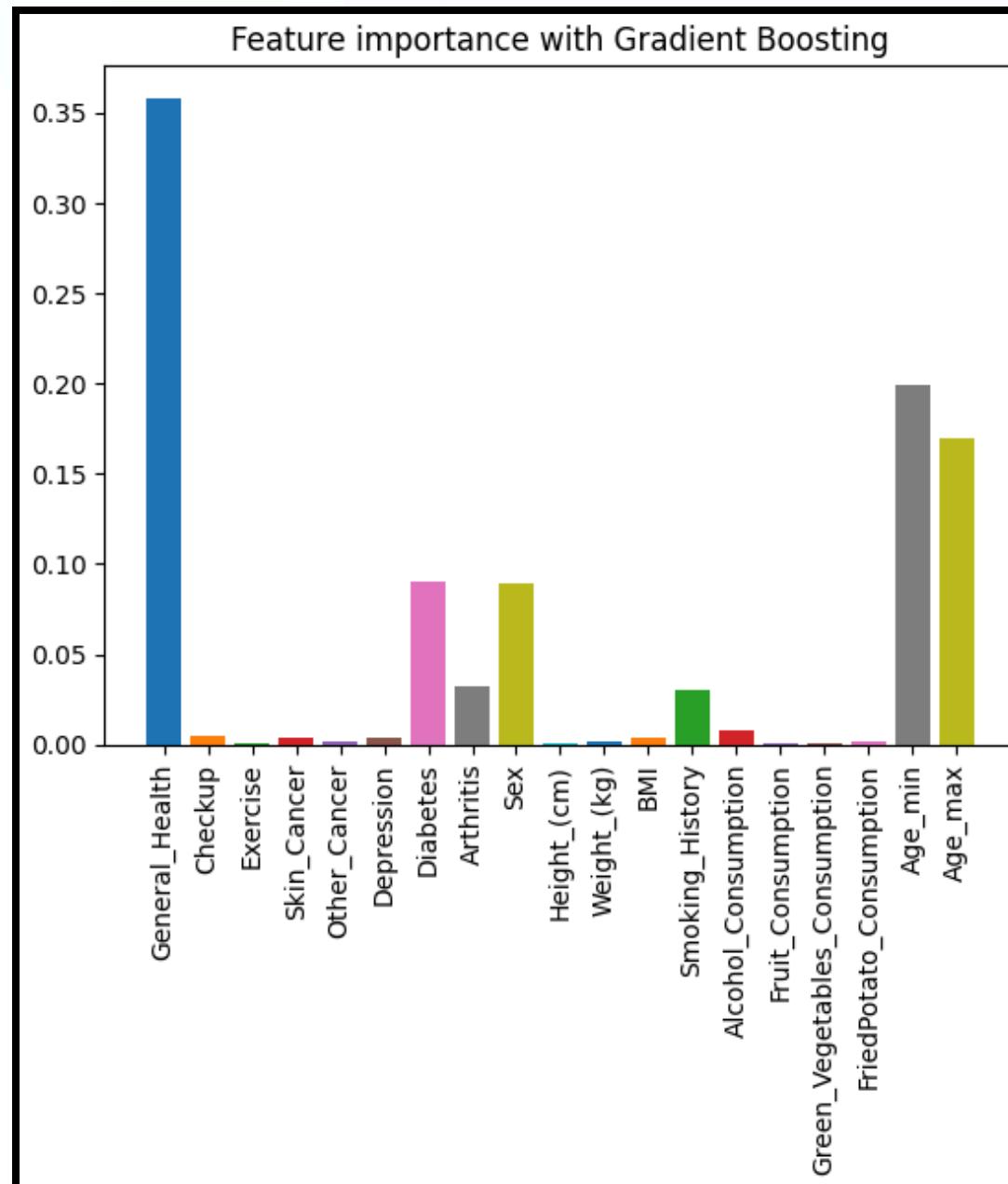
สรุป โมเดลที่มีค่าความแม่นย์มากที่สุด คือ **Gradient Boost** และ ตามมาด้วย **Random Forrest** โดยมีค่าความแม่นย์โดยเฉลี่นอยู่ที่ 0.919 และ 0.918 ตามลำดับ

	<i>dia</i>	<i>no_dia</i>	<i>Row_Average</i>
<b>GradientBoost</b>	0.919625	0.920023	0.919824
<b>RandomForest</b>	0.918327	0.918590	0.918459
<b>KNN</b>	0.910621	0.911494	0.911057
<b>DecisionTree</b>	0.861909	0.863470	0.862690



# สรุปผล

สรุปผลได้ว่า ปัจจัยต่างๆ ส่งผลต่อการเกิดโรคหลอดเหลือดและหัวใจไม่มากก็น้อย โดยถ้าเราสามารถตรวจสอบได้ทัน เราสามารถปรับเปลี่ยนวิธีการใช้ชีวิตเพื่อลดอัตราเสี่ยงในการเกิดโรคได้



โดยปัจจัยที่มีผลมากคือ BMI สุขภาพทั่วไป และอัตราการบริโภคต่างๆ

# Thank You!



โมเดลท่านายได้แม่นยำขึ้นในแต่ละรอบ.

Gradient Boosting มีลักษณะการทำงานดังนี้:

- 1. การสร้างโมเดลเบื้องต้น (Base Model):** Gradient Boosting จะเริ่มต้นด้วยโมเดลเบื้องต้นที่ง่าย (เช่น Decision Tree หนึ่งต้น) เพื่อท่านายค่าเฉลี่ยของข้อมูลหรือค่าต่าง ๆ ที่ต้องการทำนาย.
- 2. การคำนวณค่าคาดคะن (Residuals):** หลังจากการทำนายโดยโมเดลเบื้องต้น เราจะคำนวณค่าคาดคะนหรือค่าความผิดพลาดระหว่างค่าที่ทำนายได้กับค่าจริงของข้อมูล.
- 3. การสร้างโมเดลเพิ่มเติม (Boosting):** Gradient Boosting จะสร้างโมเดลเพิ่มเติมที่จะทำนายค่าคาดคะน (residuals) จากขั้นตอนก่อนหน้า โดยการปรับค่าน้ำหนักของตัวแปรในโมเดลใหม่ เพื่อลดค่าคาดคะน.
- 4. การรวมโมเดล (Aggregation):** โมเดลที่ได้จากขั้นตอนก่อนหน้าจะถูกรวมกันเพื่อทำนายค่าที่ต้องการ โดยปรับน้ำหนักให้แต่ละโมเดลเท่าๆ กันหรือตามประสิทธิภาพของแต่ละโมเดล.

Gradient Boosting มีความยืดหยุ่นสูง และสามารถใช้กับหลายปัญหาทำนายที่ต่างกันได้ เช่น การทำนายค่าต contine reading »' างการทำนายที่เป็นค่าเชิงบิริโภค (regression) หรือการทำนายประเภทของข้อมูล (classification) โดยที่ Gradient Boosting นั้นจะให้ประสิทธิภาพที่ดีในการทำนายที่มีโครงสร้างข้อมูลที่ซับซ้อนหรือแอนช์มเบิล (non-linear)



Precision-Recall Curve (PR Curve) เป็นกราฟที่ใช้ในการประเมินประสิทธิภาพของโมเดลทำนาย (predictive model) ในงานการจำแนกประเภท (classification). PR Curve นั้นแสดงความสัมพันธ์ระหว่าง precision และ recall ของโมเดล.

1. **Precision (Positive Predictive Value):** Precision คืออัตราส่วนของตัวอย่างที่ถูกตรวจพบว่าเป็น positive จริงๆ ต่อทั้งหมดของตัวอย่างที่โมเดลทำนายว่าเป็น positive. Precision วัดว่าโมเดลทำนาย positive อย่างไร่ในที่สุดเมื่อมีการทำนายว่าเป็น positive.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

2. **Recall (Sensitivity, True Positive Rate):** Recall หมายถึงอัตราส่วนของตัวอย่าง positive ทั้งหมดที่โมเดลทำนายถูกต้องว่าเป็น positive. Recall วัดว่าโมเดลมีความสามารถในการตรวจพบ positive อย่างไร่ในที่สุด.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

PR Curve ทำงานโดยการเปรียบเทียบค่า Precision และ Recall ในแต่ละระดับของค่าที่ใช้ในการตัดสินใจ (threshold) เช่นในการทำนาย binary classification, threshold จะถูกปรับเพื่อกำหนดว่าตัวอย่างควรถูกทำนายเป็น positive หรือ negative.

PR Curve มีความสำคัญเมื่อค่า positive class มีอัตราส่วนที่ต่ำหรือไม่สมดุลกับ negative class ในข้อมูล และในบางค่าสั่งของการประเมินโมเดล คุณสมบัติของ PR Curve อย่าง F1 Score หรือ AUC (Area Under the Curve) สามารถใช้เพื่อประเมินประสิทธิภาพของโมเดลได้.

⟳ Regenerate

แสดงความสัมพันธ์ระหว่างความไวต่อจำนวน False Positive Rate (FPR) และ True Positive Rate (TPR) ของโมเดลท่านาย โดยใช้ค่า threshold ในการตัดสินใจว่าตัวอย่างควรถูกท่านายเป็น positive หรือ negative.

1. **True Positive Rate (TPR):** TPR หมายถึงอัตราส่วนของตัวอย่าง positive ทั้งหมดที่โมเดลท่านายถูกต้องว่าเป็น positive หรือ Sensitivity.

$$TPR = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

2. **False Positive Rate (FPR):** FPR หมายถึงอัตราส่วนของตัวอย่าง negative ทั้งหมดที่โมเดลท่านายผิดว่าเป็น positive.

$$FPR = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

ROC-AUC Curve มีรูปร่างเป็นเส้นโค้งที่ได้จากการวาดกราฟของค่า TPR ต่อค่า FPR ในแต่ละระดับของ threshold ที่ใช้ในการท่านาย. หากโมเดลมีประสิทธิภาพดีในการจำแนกประเภท, ROC-AUC Curve จะมีความโน้มไปทางด้านบนที่เส้นแทนการจำแนกที่ดี.

ROC-AUC Score คือพื้นที่ใต้เส้น ROC-AUC Curve ซึ่งมีค่าอยู่ระหว่าง 0 ถึง 1 โดยที่ 1 หมายถึงโมเดลท่านายได้ดีที่สุด และ 0.5 หมายถึงการท่านายแบบสุ่มไม่แม่นยำมากเกินไป. ROC-AUC Score มีความสามารถในการแยกแยะระหว่าง positive class และ negative class ของโมเดล โดยที่ค่าสูงแสดงถึงประสิทธิภาพที่ดี.

Regenerate

Send a message