Masters in Applied Statistics and Data Science (WM-ASDS)

Department of Statistics

Jahangirnagar University

Savar, Dhaka-1342, Bangladesh.

A Machine Learning Assignment on Diabetes prediction using health data

WM-ASDS04: Introduction to Data Science with Python

<u>Submitted to</u>

Farhana Afrin Duty

<u>Submitted by</u>

WM-ASDS04 9th Batch, Section-A

Al-Farabi Akash Roll-20229003	**Md. Mainuddin Akash** Roll-20229055	**Yeasmin Sultana Eva** Roll-20229004	**Md Taufiq us Samad Tonmoy** Roll-20229001

ABSTRACT

Diabetes is a chronic metabolic disorder that affects millions of people worldwide, and early diagnosis is crucial in preventing complications. In this project, we aim to analyze survey data on health indicators associated with diabetes to identify the indicators that are most strongly associated with the disease. We will then develop a predictive model that can accurately identify individuals at risk of developing diabetes.To achieve these objectives, we will begin by exploring the data and examining the relationship between various health indicators and the occurrence of diabetes. We will then compare the performance of several models and select the ones that demonstrate the highest predictive accuracy. These models will be used to predict diabetes in a test dataset to assess their performance and determine their suitability for clinical use.In addition to identifying the indicators most strongly associated with diabetes, this study has the potential to contribute to the development of effective preventive strategies. By accurately identifying individuals at risk of developing diabetes, healthcare professionals can implement early interventions and prevent the onset of complications.Overall, this project aims to improve our understanding of the risk factors associated with diabetes and develop better predictive models for the early detection of this disease, ultimately contributing to the development of more effective prevention and management strategies.

INDEX

CONTENTS **Page**

1. INTRODUCTION 1

2. OBJECTIVES 2

3. LITERATURE REVIEW 3

4. REQUIREMENTS AND ANALYSIS

4.1 System Requirement 4

4.2 Knowledge Requirement 4

5. Project Implementations 5

5.1 The Python Libraries used in this project 5

5.2 Dataset 6

5.3 Exploratory data analysis 7

5.4 Categorical variables 11

5.5 Numerical variables 12

5.6 Relationship between features and target 13

5.7 Most important indicators based on correlation results 14

5.8 Machine learning models 15

5.9 Classification Report, Accuracy Score, Confusion Matrix and ROC Curve 20

5.10 Classification Results 22

6 CONCLUSION 22

7 OUTLOOK 22

1. INTRODUCTION

Diabetes is a growing public health concern, affecting millions of people worldwide. It is a chronic metabolic disorder characterized by elevated blood glucose levels, which can cause a range of complications if left untreated. Early detection of diabetes and timely intervention are critical in preventing these complications.

In this project, we aim to explore the survey data on health indicators that may be associated with diabetes. By analyzing this data, we hope to identify the key factors that are most strongly associated with diabetes. This will allow us to develop a better understanding of the risk factors for this disease and potentially help in developing targeted prevention strategies.

The two main objectives of this project are to identify the indicators most associated with diabetes and to build a model to predict diabetes. To achieve these aims, we will first explore the data to gain insights into the variables that are most strongly associated with diabetes. We will then use this information to build a predictive model that can accurately identify individuals at risk of developing diabetes.

Finally, we will compare the performance of several models and select the ones that demonstrate the highest predictive accuracy. These models will be applied to a test dataset to assess their performance and determine their suitability for predicting diabetes. By achieving these objectives, we hope to contribute to the ongoing efforts to improve the diagnosis and prevention of diabetes, ultimately improving the health outcomes of those affected by this disease.

2. OBJECTIVES

- To identify the health indicators that are most strongly associated with diabetes by analyzing survey data. This will provide valuable insights into the risk factors for the disease and inform targeted prevention strategies.
- To build a predictive model for diabetes using the identified health indicators. This will enable early detection of the disease and prompt intervention, reducing the risk of complications.
- To compare the performance of different predictive models and select the most effective ones for predicting diabetes in a test dataset. This will ensure the accuracy and reliability of the model in detecting the disease.

Overall, this project aims to improve our understanding of diabetes risk factors and develop a powerful tool for predicting the disease. By identifying the most important indicators and developing an accurate predictive model, healthcare professionals can better target interventions and prevention strategies, leading to improved health outcomes for individuals at risk of developing diabetes.

3. LITERATURE REVIEW

Diabetes is a chronic disease characterized by high levels of glucose in the blood. According to the World Health Organization (WHO), there were 422 million people worldwide with diabetes in 2014, and this number is expected to rise to 642 million by 2040 (1). Early diagnosis of diabetes is essential to prevent the onset of complications such as blindness, kidney failure, heart disease, and stroke (2).

Several health indicators have been associated with diabetes, including obesity, sedentary lifestyle, family history, high blood pressure, and abnormal lipid profiles (3,4). Obesity is a well-known risk factor for type 2 diabetes, as excess body fat can affect insulin sensitivity and glucose metabolism (5). Similarly, a sedentary lifestyle has been linked to insulin resistance and impaired glucose tolerance (6). Family history of diabetes has also been associated with an increased risk of developing the disease, possibly due to shared genetic and environmental factors (7). Hypertension and abnormal lipid profiles have been identified as independent risk factors for diabetes and cardiovascular disease (8,9).

Machine learning models have been used to predict diabetes using health indicators. In one study, logistic regression models were developed to predict the risk of diabetes in a Chinese population using age, sex, body mass index, blood pressure, and lipid profiles as predictors (10). The models achieved an accuracy of 83.7%, indicating that these health indicators were important predictors of diabetes. Another study used a random forest algorithm to predict diabetes in an Indian population, achieving an accuracy of 79% using age, body mass index, blood pressure, and fasting glucose levels as predictors (11).

Various machine learning models have been used to predict diabetes, including support vector machines, decision trees, and neural networks (12,13). The performance of these models has been evaluated using metrics such as accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC). The AUC-ROC is a commonly used metric that measures the ability of a model to distinguish between positive and negative cases, with values ranging from 0.5 (random prediction) to 1 (perfect prediction).

In summary, diabetes is a major public health issue that requires early diagnosis and intervention to prevent complications. Several health indicators have been associated with diabetes, including obesity, sedentary lifestyle, family history, high blood pressure, and abnormal lipid profiles. Machine learning models have been used to predict diabetes using these health indicators, achieving high accuracy in some cases. The performance of these models can be evaluated using metrics such as AUC-ROC, which can help identify the most effective models for predicting diabetes.

4. REQUIREMENTS AND ANALYSIS

4.1 System Requirement

Jupyter Notebook:

1. Operating System: Windows, macOS, or Linux.
2. Minimum system requirements: 4 GB of RAM, 5 GB of disk space.
3. Python version: Jupyter Notebook requires Python 3.3 or greater, as well as a number of Python packages.
4. Web browser: Jupyter Notebook is accessed through a web browser.

Google Colab:

1. Internet connection: Google Colab requires a stable internet connection to work.
2. Google Account: Google Colab is accessed through a Google account.
3. Web browser: Google Colab is accessed through a web browser.
4. GPU (optional): For certain applications, Google Colab allows you to use a GPU to speed up computations. If you require a GPU, you will need to select a GPU runtime when creating a new notebook.

4.2 Knowledge Requirement

Python Programming:

1. Basic understanding of programming concepts such as variables, data types, loops, conditional statements, functions, and objects.
2. Knowledge of Python language syntax and semantics.
3. Familiarity with the Python standard library and popular third-party modules.
4. Ability to use an IDE to write and debug Python code.
5. Understanding of how to read and write data to files using Python.

Python for Data Science:

1. Knowledge of basic statistics and probability theory.
2. Understanding of data structures and algorithms.
3. Familiarity with common data types used in data science such as pandas DataFrames and NumPy arrays.

4. Ability to manipulate and visualize data using Python libraries such as pandas and matplotlib.
5. Understanding of machine learning concepts and techniques.

Python for Machine Learning:

1. Understanding of machine learning concepts and techniques such as regression, classification, clustering, and deep learning.
2. Familiarity with machine learning algorithms and their implementation in Python.
3. Knowledge of how to preprocess data for use in machine learning models.
4. Ability to evaluate the performance of machine learning models using metrics such as accuracy, precision, and recall.
5. Understanding of how to tune machine learning models to optimize their performance.

5. Project Implementations

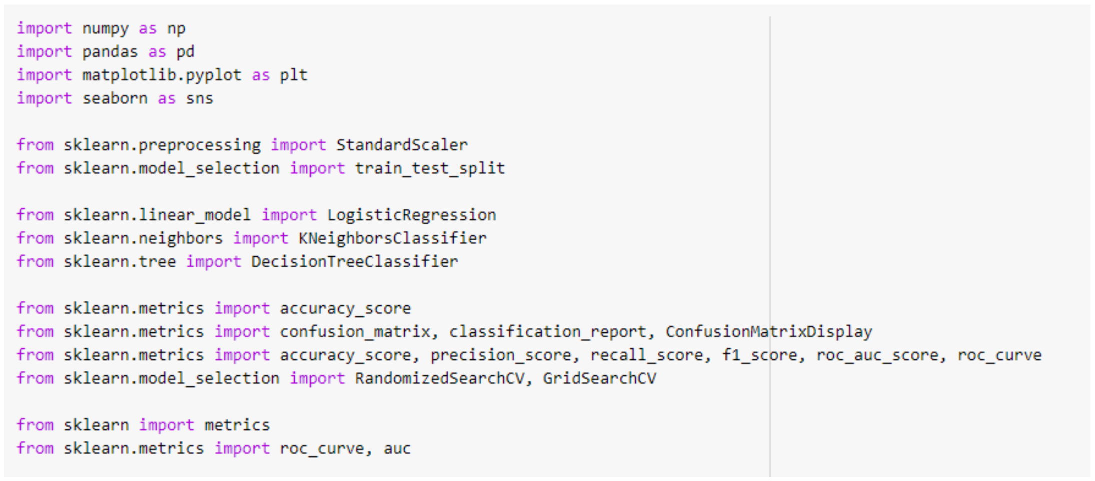
5.1 The Python Libraries used in this project

NumPy: A library for numerical computing and analysis in Python, used for handling large datasets and mathematical operations.

Pandas: A library for data manipulation and analysis in Python, providing data structures and functions for filtering, transforming, and aggregating data.

Matplotlib.pyplot: A library for creating visualizations and plots in Python, used for exploring and presenting data.

scikit-learn (sklearn): A machine learning library in Python, providing a wide range of algorithms for classification, regression, clustering, and dimensionality reduction, as well as tools for model selection, evaluation, and preprocessing of data.



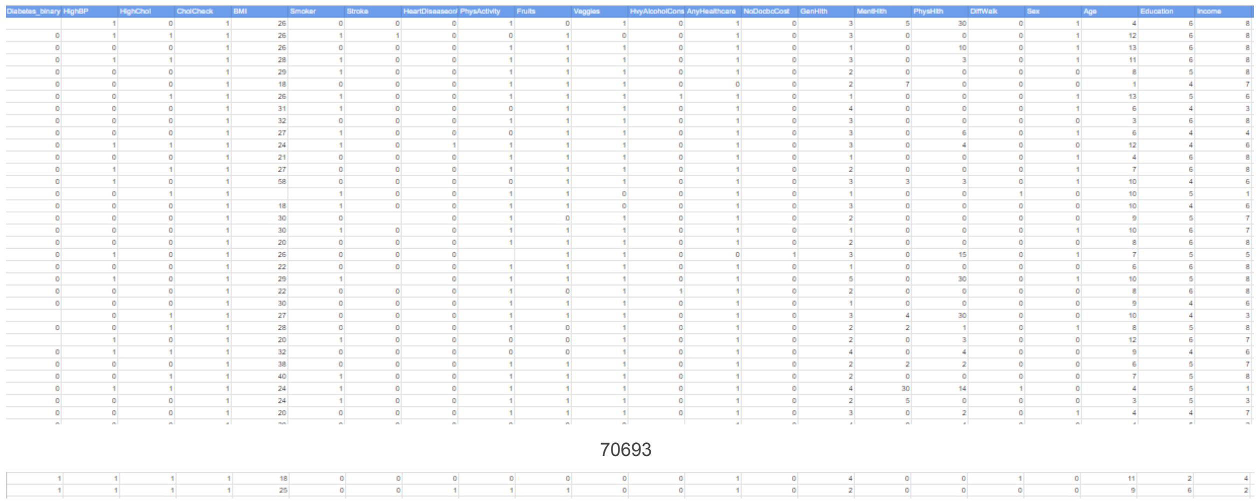
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, classification_report, ConfusionMatrixDisplay
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, roc_curve
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV

from sklearn import metrics
from sklearn.metrics import roc_curve, auc
```

Libraries used in this project.

5.2 Dataset

The Dataset used in the project:

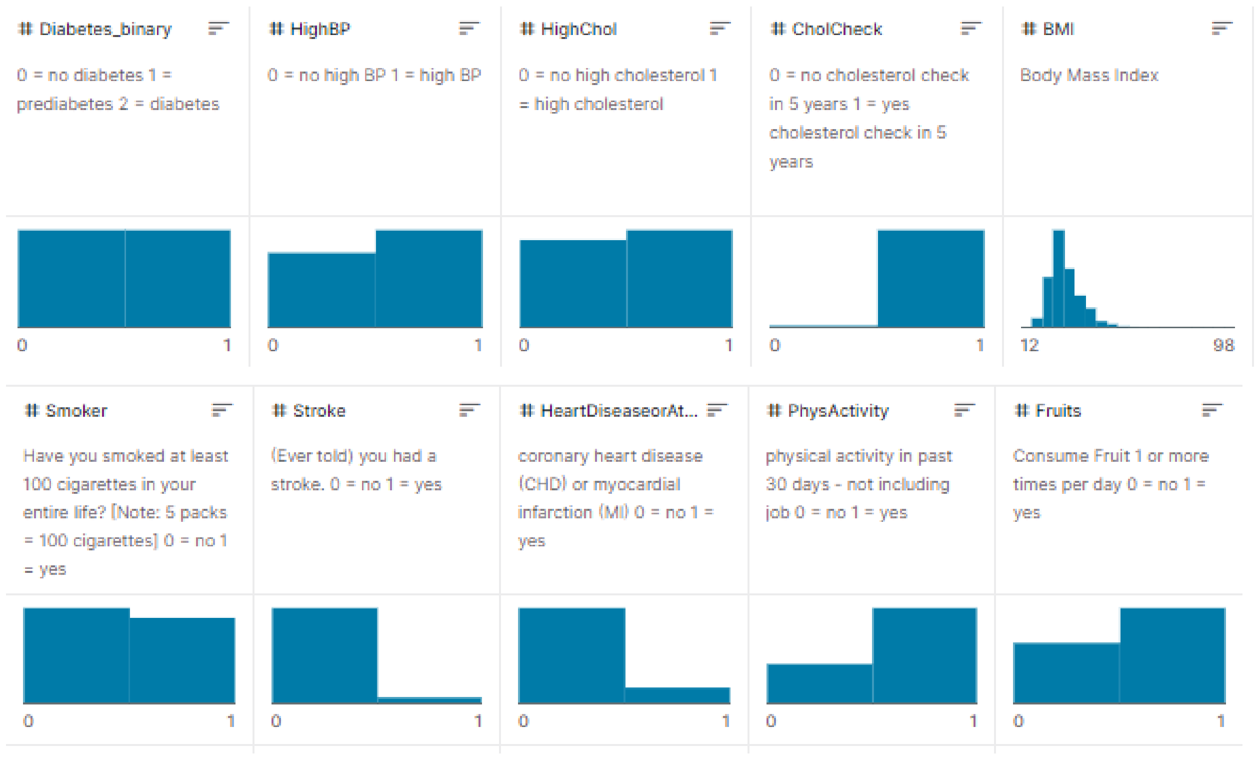
This is a dataset of 70,692 survey responses to the CDC's BRFSS2015. It has an equal 50-50 split of respondents with no diabetes and with either prediabetes or diabetes. The target variable Diabetes_binary has 2 classes. 0 is for no diabetes, and 1 is for prediabetes or diabetes. This dataset has 21 feature variables and is balanced.



Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseor	PhysActivity	Fruits	Veggies	HvyAlcoholCons	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
	1	0	1	26	0	0	0	1	0	1	0	1	0	3	5	30	0	1	4	6	8
0	1	1	1	26	1	1	0	0	1	0	0	1	0	3	0	0	0	1	12	6	8
0	0	0	1	26	0	0	0	1	1	1	0	1	0	1	0	10	0	1	13	6	8
0	1	1	1	28	1	0	0	1	1	1	0	1	0	3	0	3	0	1	11	6	8
0	0	0	1	29	1	0	0	1	1	1	0	1	0	2	0	0	0	0	8	5	8
0	0	0	1	18	0	0	0	1	1	1	0	0	0	2	7	0	0	0	1	4	7
0	0	1	1	26	1	0	0	1	1	1	1	1	0	1	0	0	0	1	13	5	6
0	0	0	1	31	1	0	0	0	1	1	0	1	0	4	0	0	0	1	6	4	3
0	0	0	1	32	0	0	0	1	1	1	0	1	0	3	0	0	0	0	3	6	8
0	0	0	1	27	1	0	0	0	1	1	0	1	0	3	0	6	0	1	6	4	4
0	1	1	1	24	1	0	1	1	1	1	0	1	0	3	0	4	0	0	12	4	6
0	0	0	1	21	0	0	0	1	1	1	0	1	0	1	0	0	0	1	4	6	8
0	1	1	1	27	0	0	0	1	1	1	0	1	0	2	0	0	0	1	7	6	8
0	1	0	1	58	0	0	0	0	1	1	0	1	0	3	3	3	0	1	10	4	6
0	0	1	1		1	0	0	1	1	0	0	1	0	1	0	0	1	0	10	5	1
0	0	0	1	18	1	0	0	1	1	0	0	1	0	3	0	0	0	0	10	4	6
0	0	0	1	30	0		0	1	0	1	0	1	0	2	0	0	0	0	9	5	7
0	0	0	1	30	1	0	0	1	1	1	0	1	0	1	0	0	0	1	10	6	7
0	0	0	1	20	0	0	0	1	1	1	0	1	0	2	0	0	0	0	8	6	8
0	1	0	1	26	0	0	0		1	1	0	0	1	3	0	15	0	1	7	5	5
0	0	0	1	22	0	0	0	1	1	1	0	1	0	1	0	0	0	0	6	6	8
0	1	0	1	29	1		0	1	1	1	0	1	0	5	0	30	0	1	10	5	8
0	0	0	1	22	0	0	0	1	0	1	1	1	0	2	0	0	0	0	8	6	8
0	0	0	1	30	0	0	0	1	1	1	0	1	0	1	0	0	0	0	9	4	6
	0	1	1	27	0	0	0	1	1	1	0	1	0	3	4	30	0	0	10	4	3
0	0	1	1	28	0	0	0	1	0	1	0	1	0	2	2	1	0	1	8	5	8
	1	0	1	20	1	0	0	0	0	1	0	1	0	2	0	3	0	0	12	6	7
0	1	1	1	32	0	0	0	0	0	1	0	1	0	4	0	4	0	0	9	4	6
0	0	0	1	38	0	0	0	1	1	1	0	1	0	2	2	2	0	0	6	5	7
0	0	1	1	40	1	0	0	1	1	1	0	1	0	2	0	0	0	0	7	5	8
0	1	1	1	24	1	0	0	1	1	1	0	1	0	4	30	14	1	0	4	5	1
0	0	0	1	24	1	0	0	1	1	1	0	1	0	2	5	0	0	0	3	5	3
0	0	0	1	20	0	0	0	1	1	1	0	1	0	3	0	2	0	1	4	4	7

70693

1	1	1	1	18	0	0	0	0	0	0	0	1	0	4	0	0	1	0	11	2	4
1	1	1	1	25	0	0	1	1	1	0	0	1	0	2	0	0	0	0	9	6	2


Diabetes_binary
0 = no diabetes 1 = prediabetes 2 = diabetes

HighBP
0 = no high BP 1 = high BP

HighChol
0 = no high cholesterol 1 = high cholesterol

CholCheck
0 = no cholesterol check in 5 years 1 = yes cholesterol check in 5 years

BMI
Body Mass Index

Smoker
Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] 0 = no 1 = yes

Stroke
(Ever told) you had a stroke. 0 = no 1 = yes

HeartDiseaseorAt...
coronary heart disease (CHD) or myocardial infarction (MI) 0 = no 1 = yes

PhysActivity
physical activity in past 30 days - not including job 0 = no 1 = yes

Fruits
Consume Fruit 1 or more times per day 0 = no 1 = yes

5.3 Exploratory data analysis

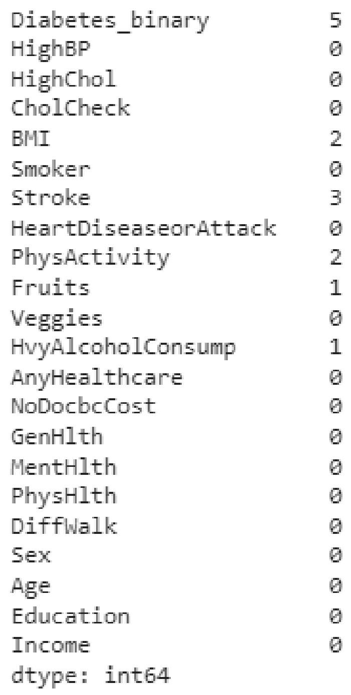
In this section, We explored the training data to find out more about the data.

Note that the EDA is applied to all subjects, including both diabetes and non-diabetes groups.

The unique values for all variable

```
Diabetes_binary: [ 0.  1. nan]
HighBP: [0 1]
HighChol: [0 1]
CholCheck: [0 1]
BMI: [12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29.
 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47.
 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65.
 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83.
 84. 85. 86. 87. 89. 92. 95. 98. nan]
Smoker: [0 1]
Stroke: [ 0.  1. nan]
HeartDiseaseorAttack: [0 1]
PhysActivity: [ 0.  1. nan]
Fruits: [ 0.  1. nan]
Veggies: [0 1]
HvyAlcoholConsump: [ 0.  1. nan]
AnyHealthcare: [0 1]
NoDocbcCost: [0 1]
GenHlth: [1 2 3 4 5]
MentHlth: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30]
PhysHlth: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30]
DiffWalk: [0 1]
Sex: [0 1]
Age: [ 1  2  3  4  5  6  7  8  9 10 11 12 13]
Education: [1 2 3 4 5 6]
Income: [1 2 3 4 5 6 7 8]
```

Check missing values

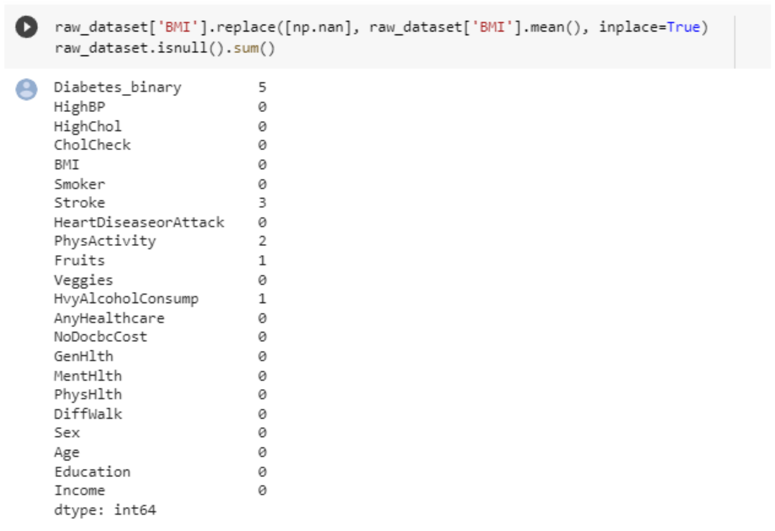


```
Diabetes_binary         5
HighBP                  0
HighChol                0
CholCheck               0
BMI                     2
Smoker                  0
Stroke                  3
HeartDiseaseorAttack    0
PhysActivity            2
Fruits                  1
Veggies                 0
HvyAlcoholConsump       1
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64
```

We can see that the data contains some missing values. We will try to remove them. There are missing fields in the following rows:

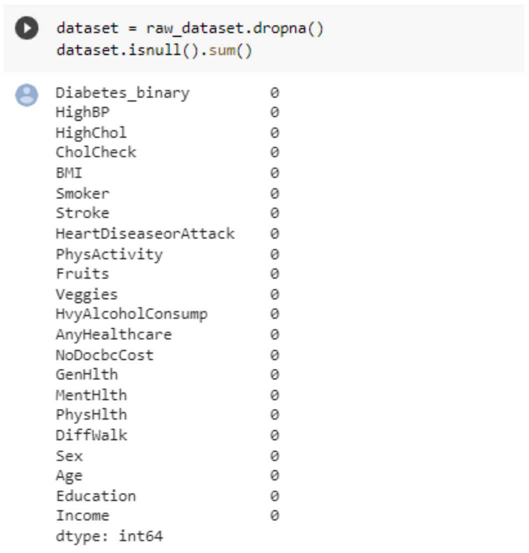
- Diabetes_binary
- BMI
- Stroke
- PhysActivity
- Fruits
- HvyAlcoholConsump

Since all the column values except BMI are Binary, we will replace the BMI missing values with mean and For the other columns we will remove the rows entirely.



```
raw_dataset['BMI'].replace([np.nan], raw_dataset['BMI'].mean(), inplace=True)
raw_dataset.isnull().sum()
```

```
Diabetes_binary         5
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  3
HeartDiseaseorAttack    0
PhysActivity            2
Fruits                  1
Veggies                 0
HvyAlcoholConsump       1
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64
```

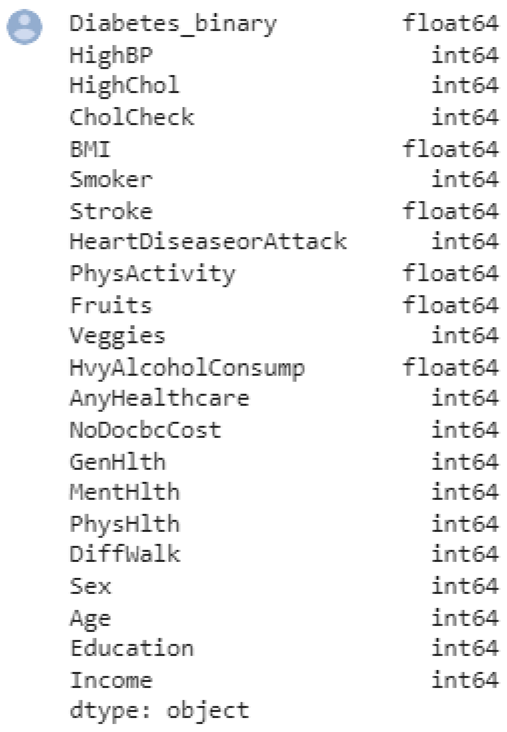


```
dataset = raw_dataset.dropna()
dataset.isnull().sum()
```

```
Diabetes_binary         0
HighBP                  0
HighChol                0
CholCheck               0
BMI                     0
Smoker                  0
Stroke                  0
HeartDiseaseorAttack    0
PhysActivity            0
Fruits                  0
Veggies                 0
HvyAlcoholConsump       0
AnyHealthcare           0
NoDocbcCost             0
GenHlth                 0
MentHlth                0
PhysHlth                0
DiffWalk                0
Sex                     0
Age                     0
Education               0
Income                  0
dtype: int64
```

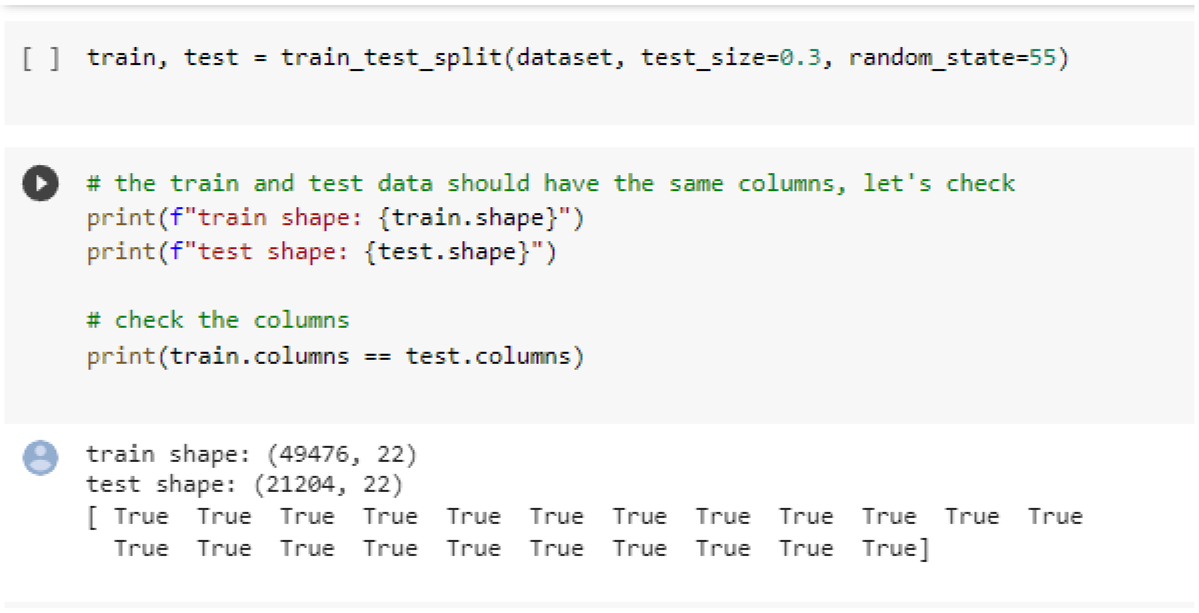
After the cleaning process it looks like we have clean datasets. There are no missing values and all the data types are float (even they are categorical).

Checking the data types



```
Diabetes_binary          float64
HighBP                     int64
HighChol                   int64
CholCheck                  int64
BMI                      float64
Smoker                     int64
Stroke                   float64
HeartDiseaseorAttack       int64
PhysActivity             float64
Fruits                   float64
Veggies                    int64
HvyAlcoholConsump        float64
AnyHealthcare              int64
NoDocbcCost                int64
GenHlth                    int64
MentHlth                   int64
PhysHlth                   int64
DiffWalk                   int64
Sex                        int64
Age                        int64
Education                  int64
Income                     int64
dtype: object
```

Defining train & test data

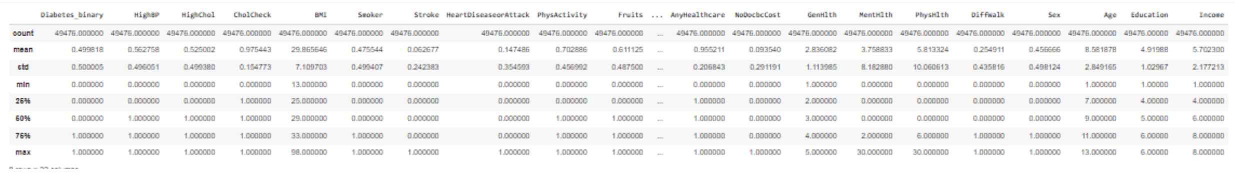


```
[ ]  train, test = train_test_split(dataset, test_size=0.3, random_state=55)
```

```
# the train and test data should have the same columns, let's check
print(f"train shape: {train.shape}")
print(f"test shape: {test.shape}")

# check the columns
print(train.columns == test.columns)
```

```
train shape: (49476, 22)
test shape: (21204, 22)
[ True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True  True  True  True]
```

Train data



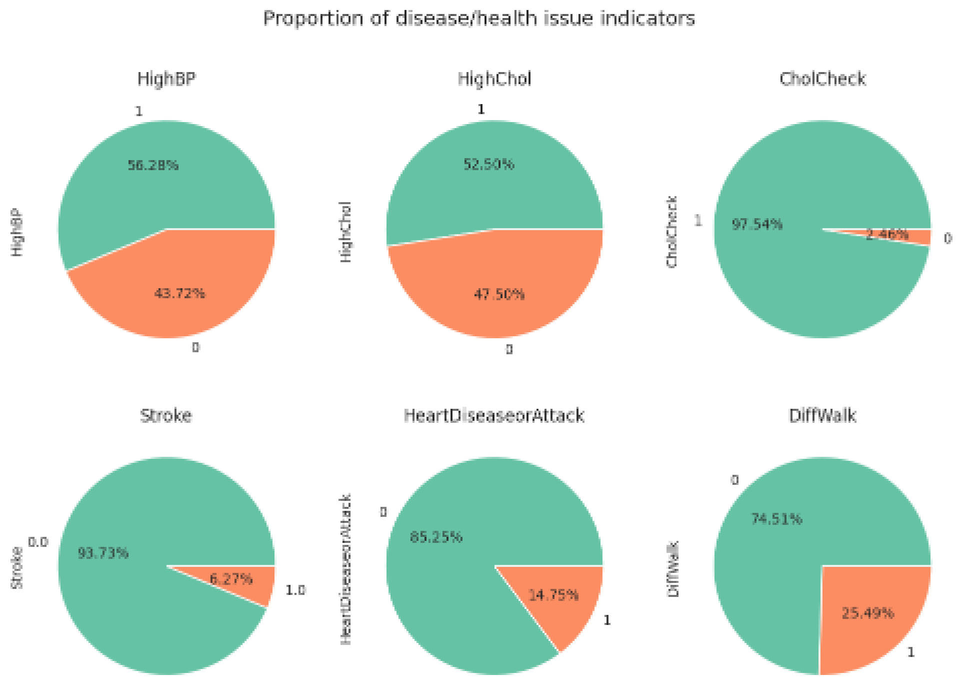
	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	...
count	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	...
mean	0.499818	0.562758	0.525002	0.975443	29.865646	0.475544	0.062677	0.147486	0.702886	0.611125	...
std	0.500005	0.496051	0.499380	0.154773	7.109703	0.499407	0.242383	0.354593	0.456992	0.487500	...
min	0.000000	0.000000	0.000000	0.000000	13.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
25%	0.000000	0.000000	0.000000	1.000000	25.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
50%	0.000000	1.000000	1.000000	1.000000	29.000000	0.000000	0.000000	0.000000	1.000000	1.000000	...
75%	1.000000	1.000000	1.000000	1.000000	33.000000	1.000000	0.000000	0.000000	1.000000	1.000000	...
max	1.000000	1.000000	1.000000	1.000000	98.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...

	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
count	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.000000	49476.00000	49476.000000
mean	0.955211	0.093540	2.836082	3.758833	5.813324	0.254911	0.456666	8.581878	4.91988	5.702300
std	0.206843	0.291191	1.113985	8.182880	10.060613	0.435816	0.498124	2.849165	1.02967	2.177213
min	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.00000	1.000000
25%	1.000000	0.000000	2.000000	0.000000	0.000000	0.000000	0.000000	7.000000	4.00000	4.000000
50%	1.000000	0.000000	3.000000	0.000000	0.000000	0.000000	0.000000	9.000000	5.00000	6.000000
75%	1.000000	0.000000	4.000000	2.000000	6.000000	1.000000	1.000000	11.000000	6.00000	8.000000
max	1.000000	1.000000	5.000000	30.000000	30.000000	1.000000	1.000000	13.000000	6.00000	8.000000

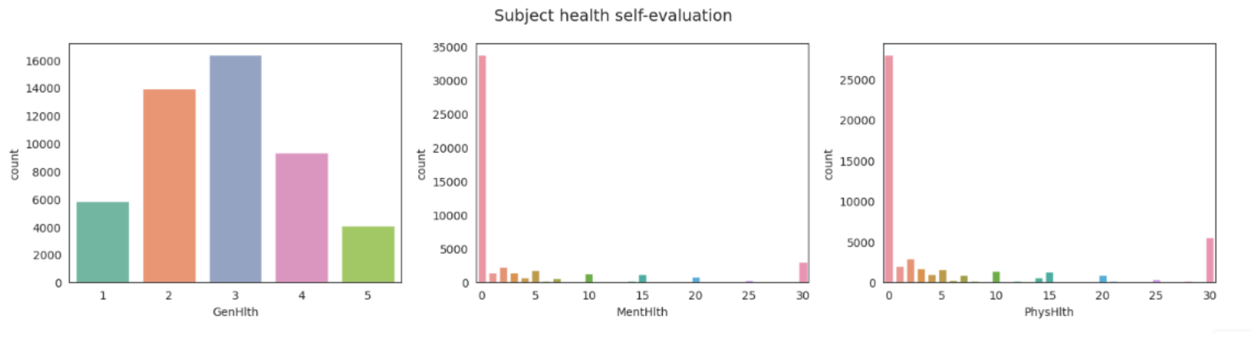
5.4 Categorical variables

```
[ ] # categorical columns - We roughly grouped them
    cat_socialecom = ['Age', 'Sex', 'Education', 'Income', 'AnyHealthcare', 'NoDocbcCost']
    cat_disease = ['HighBP', 'HighChol', 'CholCheck', 'Stroke', 'HeartDiseaseorAttack', 'DiffWalk']
    cat_health = ['GenHlth', 'MentHlth', 'PhysHlth']
    cat_habit = ['PhysActivity', 'Smoker', 'Fruits', 'Veggies', 'HvyAlcoholConsump']
```

Checking the distribution of binary health indicators using a pie chart

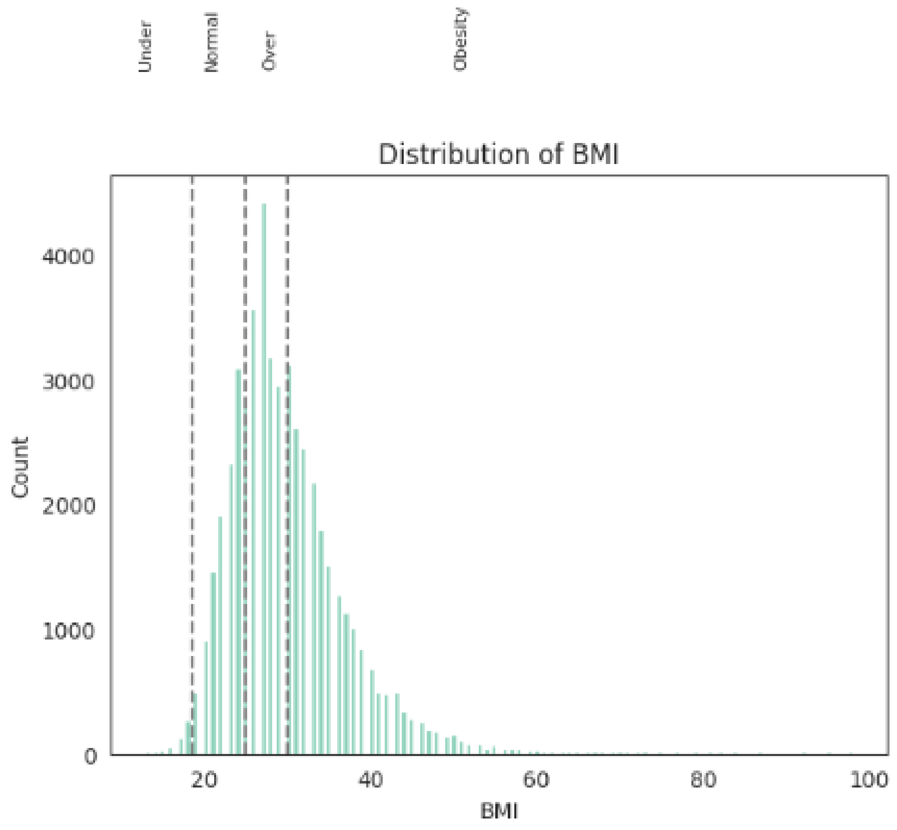


Checking the health self-evaluation



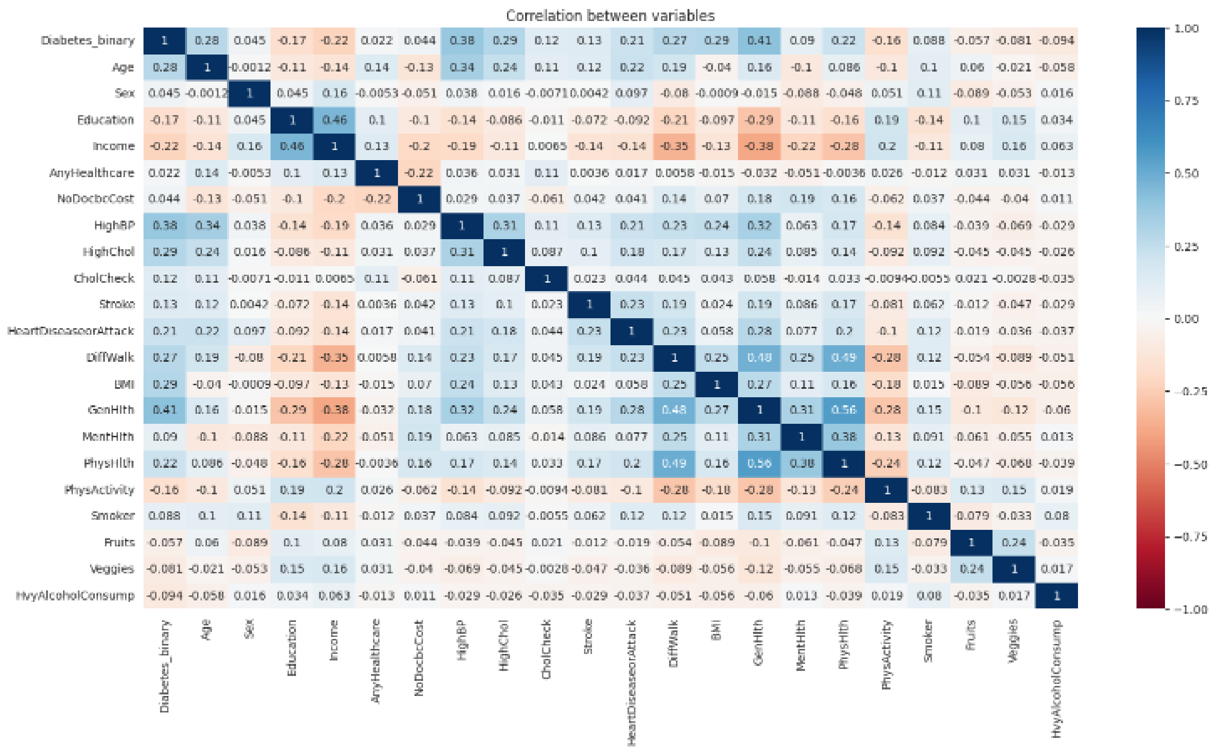
5.5 Numerical variables

The only numerical variable analyzed in this section is the BMI index, although it is probably rounded to the nearest integer (thus they can be regarded as categorical variables in some sense).

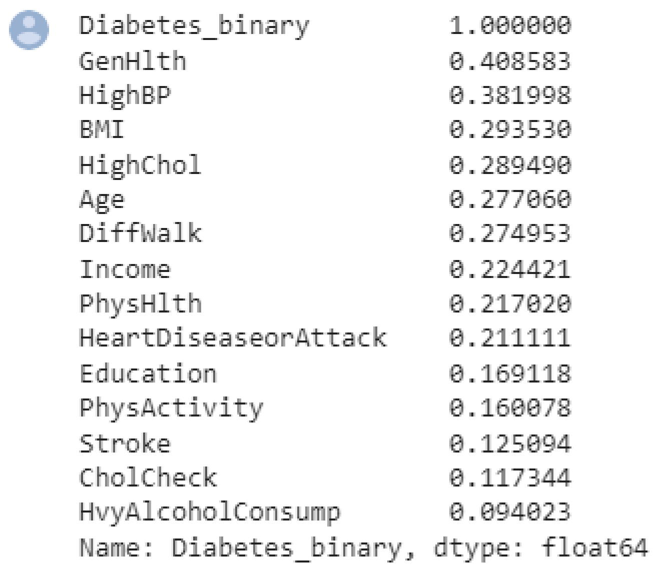


5.6 Relationship between features and target

```
# plot a heatmap to check the correlation between the variables

# for better grouping and interpretation,sort columns by the list of columns created above
cols = ['Diabetes_binary'] + cat_socialecom + cat_disease +  ['BMI'] + cat_health + cat_habit

plt.figure(figsize=(18, 9))
# nomralize the color scale
sns.heatmap(train[cols].corr(), annot=True, cmap='RdBu', vmin=-1, vmax=1)
plt.title('Correlation between variables')
plt.show()
```



Top 15 variables that are correlated with the target variable - use absolute value to get the absolute correlation



```
Diabetes_binary         1.000000
GenHlth                 0.408583
HighBP                  0.381998
BMI                     0.293530
HighChol                0.289490
Age                     0.277060
DiffWalk                0.274953
Income                  0.224421
PhysHlth                0.217020
HeartDiseaseorAttack    0.211111
Education               0.169118
PhysActivity            0.160078
Stroke                  0.125094
CholCheck               0.117344
HvyAlcoholConsump       0.094023
Name: Diabetes_binary, dtype: float64
```

5.7 Most important indicators based on correlation results

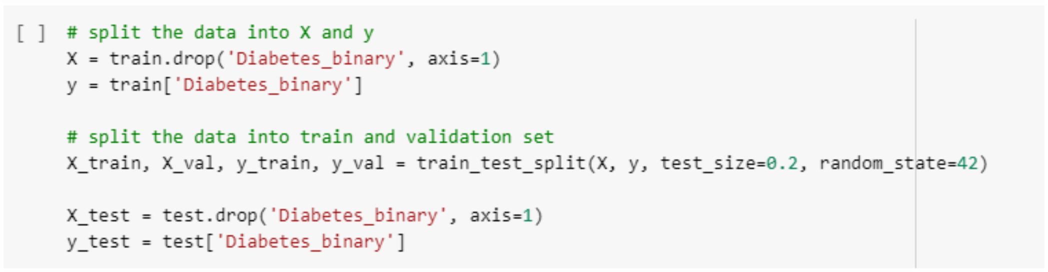
Diabetes looks positively correlated with age and negatively correlated with income.
It seems that subjects fairly evaluated their health status: if they think their health is poor, they are more likely to have diabetes. Similar correlation can be found between having diabetes amd the evaluation of physical status.
The physical measures are correlated with diabetes, which is expected. High BP, high BMI, high Cholesterol are correlated with having diabetes.
Health issues and diseases are more correlated with diabetes than health habits. For example, the chance of having a heart attack, stroke, or difficulties in walking is correlated with having diabetes.
Among the health habits, whether having physical activity in the past 30 days (PhysActivity) is the most correlated with diabetes (in a negative way).
Overall, correlation results show some indicators that are most effective in predicting diabetes. Note that it cannot imply causation. For example, the correlation between diabetes and BP is positive, but it does not mean that high BP causes diabetes. For the next step, we will use machine learning models to predict diabetes based on the indicators.

5.8 Machine learning models

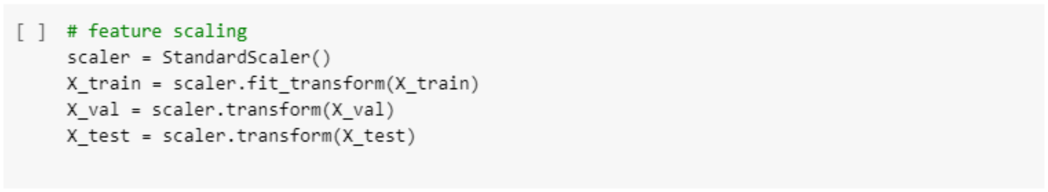
The target variable of the prediction is Diabetes (1 for (pre-)diabetes, 0 for non-diabetes). The feature variables are all the other variables. Therefore, the problem is a binary classification problem. I will try the following models to predict diabetes:

- Logistic regression
- Decision tree
- K-nearest neighbors

The model will be trained on the training data and tested on the test data.

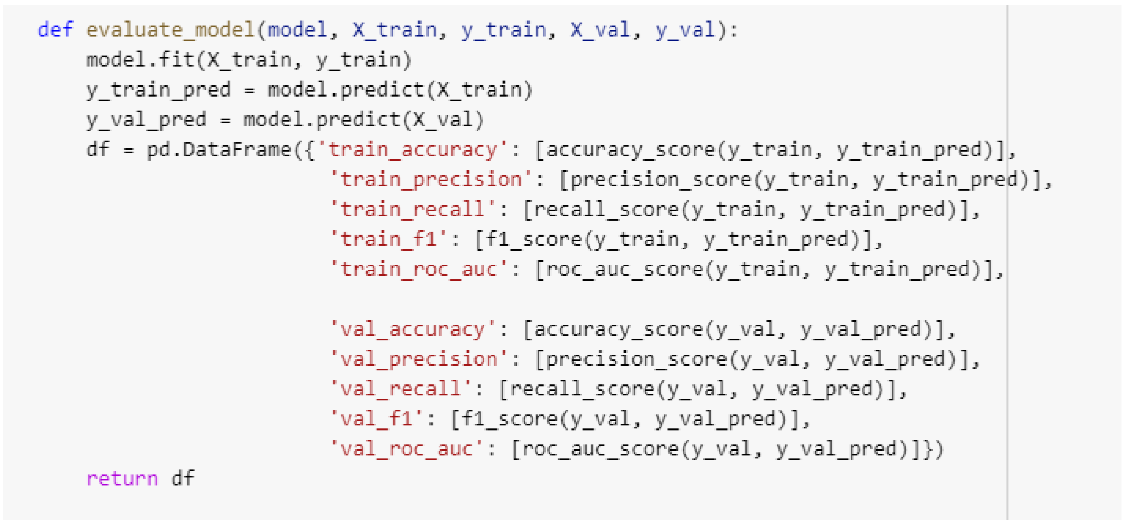


```
[ ] # split the data into X and y
    X = train.drop('Diabetes_binary', axis=1)
    y = train['Diabetes_binary']

    # split the data into train and validation set
    X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

    X_test = test.drop('Diabetes_binary', axis=1)
    y_test = test['Diabetes_binary']
```



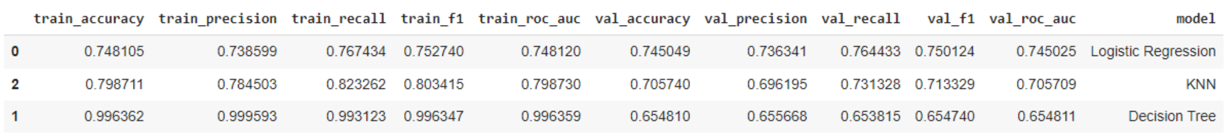
```
[ ] # feature scaling
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_val = scaler.transform(X_val)
    X_test = scaler.transform(X_test)
```

Model fitting and evaluation



```
def evaluate_model(model, X_train, y_train, X_val, y_val):
    model.fit(X_train, y_train)
    y_train_pred = model.predict(X_train)
    y_val_pred = model.predict(X_val)
    df = pd.DataFrame({'train_accuracy': [accuracy_score(y_train, y_train_pred)],
                       'train_precision': [precision_score(y_train, y_train_pred)],
                       'train_recall': [recall_score(y_train, y_train_pred)],
                       'train_f1': [f1_score(y_train, y_train_pred)],
                       'train_roc_auc': [roc_auc_score(y_train, y_train_pred)],

                       'val_accuracy': [accuracy_score(y_val, y_val_pred)],
                       'val_precision': [precision_score(y_val, y_val_pred)],
                       'val_recall': [recall_score(y_val, y_val_pred)],
                       'val_f1': [f1_score(y_val, y_val_pred)],
                       'val_roc_auc': [roc_auc_score(y_val, y_val_pred)]})
    return df
```

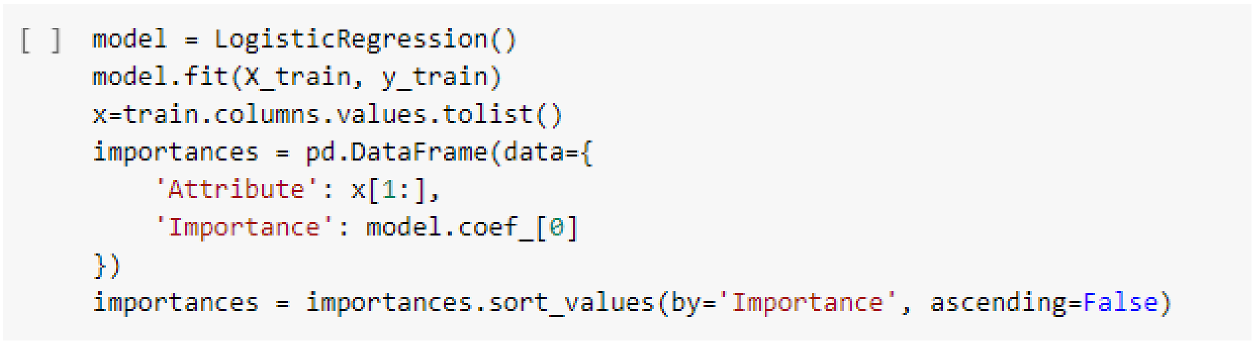
Result based on validation accuracy



	train_accuracy	train_precision	train_recall	train_f1	train_roc_auc	val_accuracy	val_precision	val_recall	val_f1	val_roc_auc	model
0	0.748105	0.738599	0.767434	0.752740	0.748120	0.745049	0.736341	0.764433	0.750124	0.745025	Logistic Regression
2	0.798711	0.784503	0.823262	0.803415	0.798730	0.705740	0.696195	0.731328	0.713329	0.705709	KNN
1	0.996362	0.999593	0.993123	0.996347	0.996359	0.654810	0.655668	0.653815	0.654740	0.654811	Decision Tree

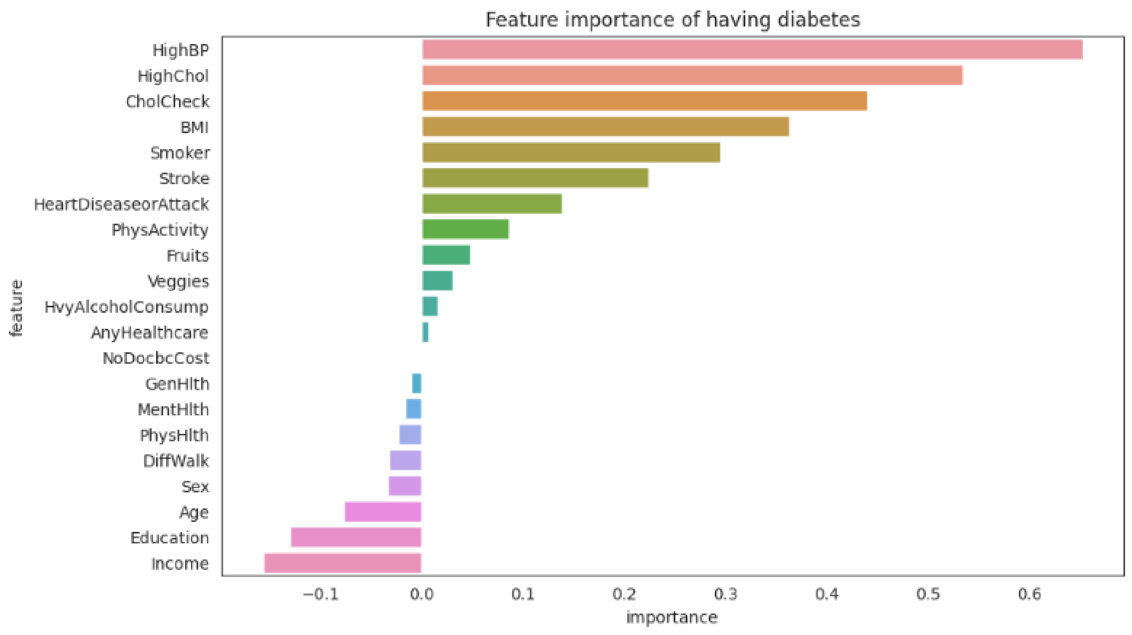
The Decision Tree model seems overfitting: the accuracy score on the training data is much higher than that on the test data.

Logistic Regression



```
[ ]  model = LogisticRegression()
     model.fit(X_train, y_train)
     x=train.columns.values.tolist()
     importances = pd.DataFrame(data={
         'Attribute': x[1:],
         'Importance': model.coef_[0]
     })
     importances = importances.sort_values(by='Importance', ascending=False)
```

Plot of the feature importance



The total importance of the top 8 features

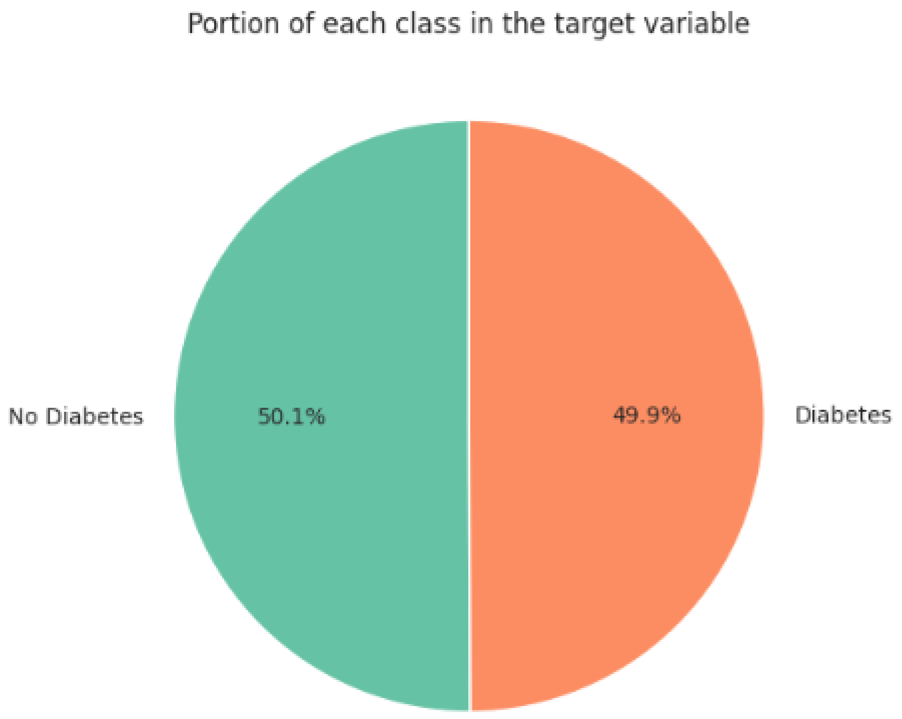
```
feature_importance.head(8)['importance'].sum()
```

```
2.7372730231624556
```

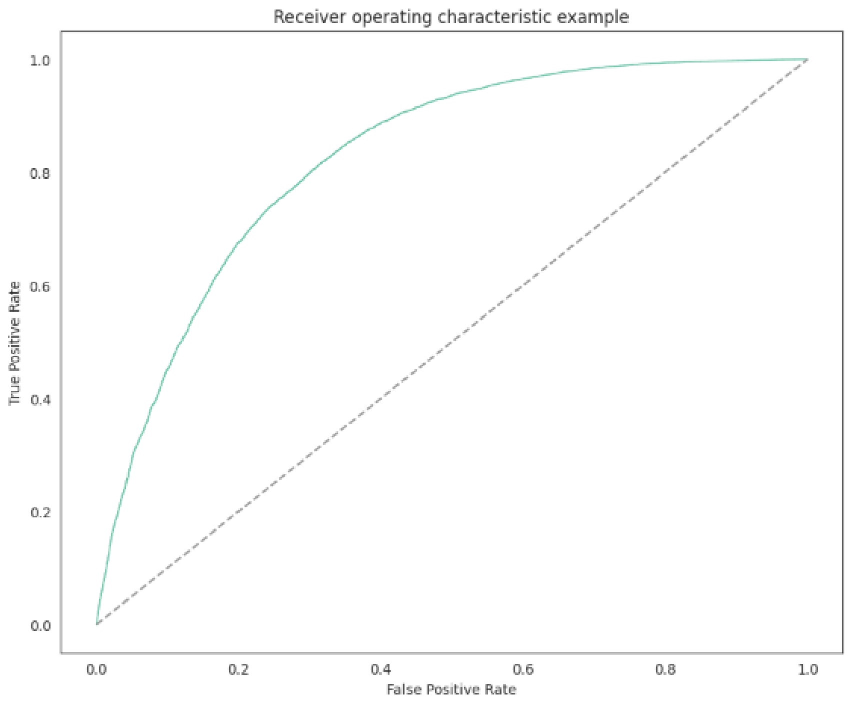
We can find that feature importance analysis results are roughly consistent with the correlation results.

Predict test data

```
# plot the portion of each class in the target variable as pie chart
plt.figure(figsize=(6, 6))
plt.pie(y_test.value_counts(), labels=['No Diabetes', 'Diabetes'], autopct='%1.1f%%', startangle=90)
plt.title('Portion of each class in the target variable')
plt.show()
```



Receiver operating characteristic example of Logistic Regression

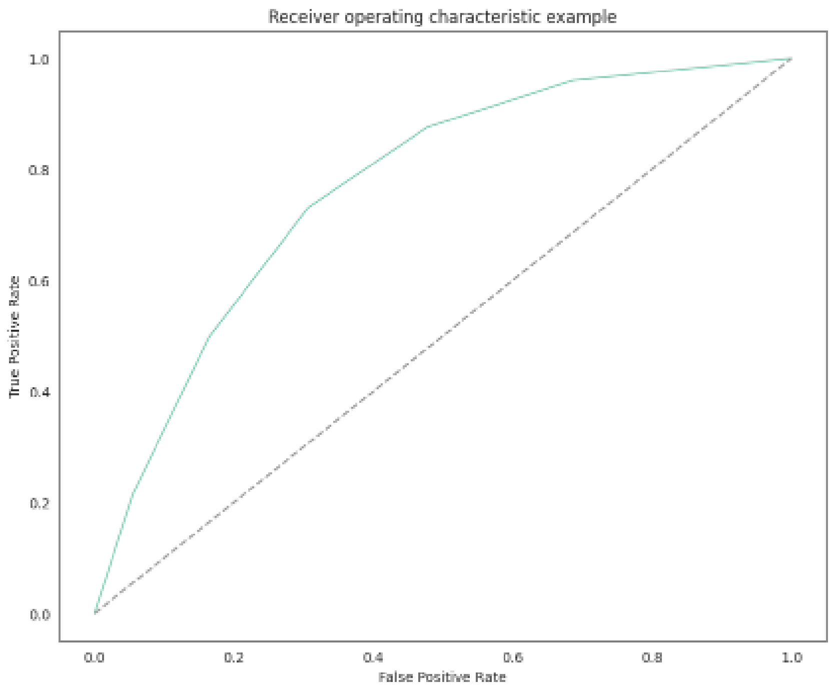

Receiver operating characteristic example
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

K-Nearest Neighbors (KNN)

```
print('Train Score is : ' , knn.score(X_train, y_train))
print('Test Score is : ' , knn.score(X_test, y_test))
```

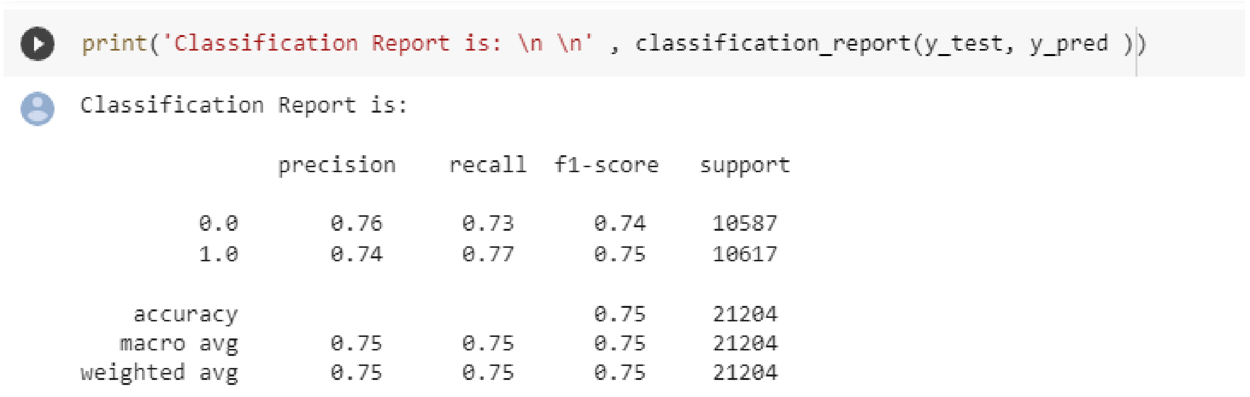
```
Train Score is :  0.798711470439616
Test Score is :  0.7123184304848141
```

Receiver operating characteristic example of K-Nearest Neighbors (KNN)



5.9 Classification Report, Accuracy Score, Confusion Matrix and ROC Curve

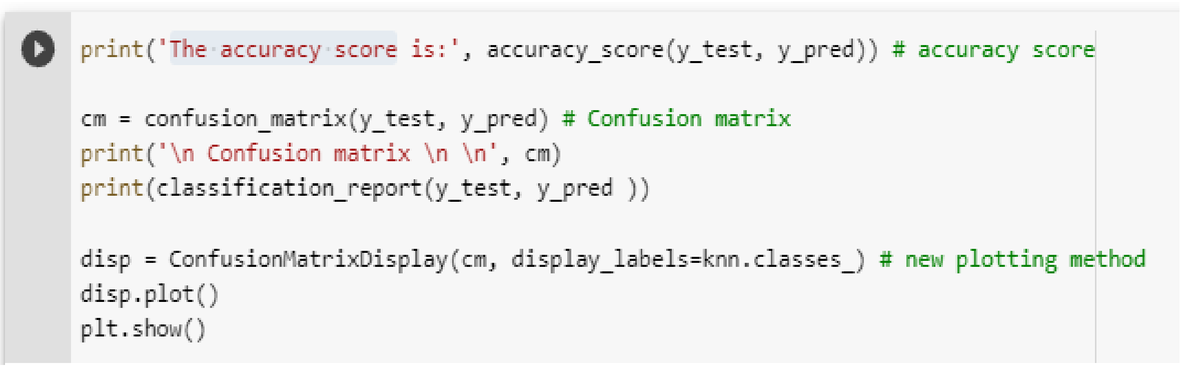
Classification Report



```
print('Classification Report is: \n \n' , classification_report(y_test, y_pred ))
```

```
Classification Report is:

              precision    recall  f1-score   support

         0.0       0.76      0.73      0.74     10587
         1.0       0.74      0.77      0.75     10617

    accuracy                           0.75     21204
   macro avg       0.75      0.75      0.75     21204
weighted avg       0.75      0.75      0.75     21204
```

The accuracy score & Confusion matrix



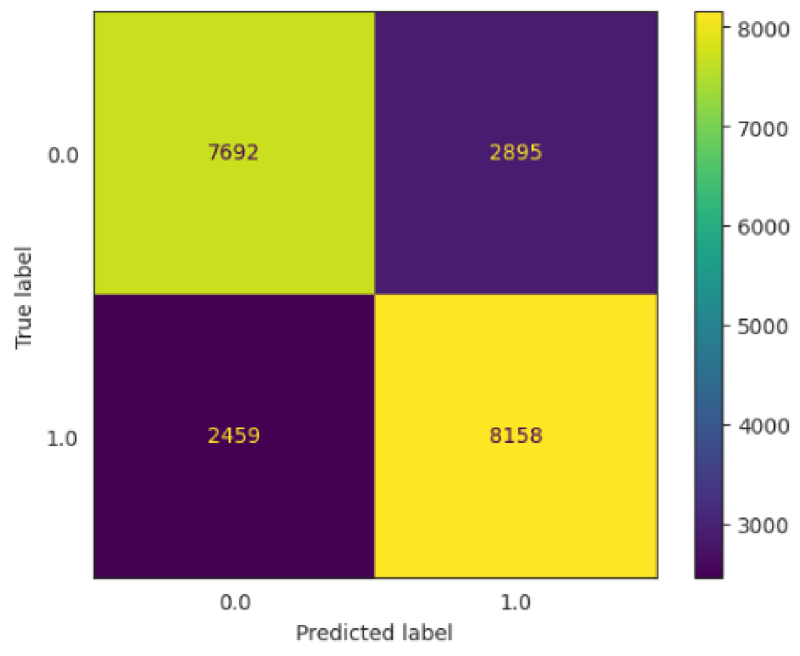
```
print('The accuracy score is:', accuracy_score(y_test, y_pred)) # accuracy score

cm = confusion_matrix(y_test, y_pred) # Confusion matrix
print('\n Confusion matrix \n \n', cm)
print(classification_report(y_test, y_pred ))

disp = ConfusionMatrixDisplay(cm, display_labels=knn.classes_) # new plotting method
disp.plot()
plt.show()
```

The accuracy score

```
The accuracy score is: 0.7475004716091304
```

Confusion matrix

```
[[7692 2895]
 [2459 8158]]
              precision    recall  f1-score   support

         0.0       0.76      0.73      0.74     10587
         1.0       0.74      0.77      0.75     10617

    accuracy                           0.75     21204
   macro avg       0.75      0.75      0.75     21204
weighted avg       0.75      0.75      0.75     21204
```



ROC AUC Score

0.7474708742078746

5.10 Classification Results

The model prediction looks okay, with 0.75 accuracy score. Note that the precision for the positive class (Diabetes) is very low. This is because the dataset is imbalanced.

The ROC AUC score is 0.75, which is not bad. We can also find that in the check auc-roc curve.

6. CONCLUSION

What are the most important indicators for diabetes?

- ○ Both correlation and feature importance analysis show that the most important indicators are:
    - ■ High blood pressure
    - ■ High BMI
    - ■ High cholesterol
    - ■ Smoking Habit
    - ■ Having heart disease or attack
    - ■ Age
    - ■ DIfficulties in walking
    - ■ Income
- ○ Interestingly, health habits are not the most important indicators.

- We can predict diabetes with a 0.74 accuracy score after model selection.
- The indicators listed above together can contribute to the majority of the performance of the model.

7. OUTLOOK

It is interesting to think about the causation. Further analysis can be done to find out the causal relationship between the indicators and diabetes.