1. Data Retrieving

ขั้นแรกทำได้โดยการนำเข้า library ต่างๆที่ต้องใช้ในการประมวลผลข้อมูล และทำการอ่านไฟล์ csv

import numpy as np

import pandas as pd

from pandas\_profiling import ProfileReport

import matplotlib.pyplot as plt

import seaborn as sns

df\_original = pd.read\_csv('Diabetes.csv')

profile = ProfileReport(df\_original)

profile

2. Data Manipulation

ใช้ library ProfileReport เพื่อดูข้อมูลต่างๆว่ามีข้อมูลแต่ละ column เป็นยังไง มีลักษณะยังไง มีข้อมูลสูญหายหรือไม่

ต่อจากนั้นทำการ dropna ทิ้งข้อมูลที่มีค่าว่างไปเลย เพราะดูจาก profile report แล้วมีจำนวนข้อมูลที่ missing น้อยกว่าจำนวนข้อมูลต้นฉบับมาก เพื่อความง่ายจึงเลือกที่ drop ทั้งหมดทิ้งไปเลย เปลี่ยน Male เป็น 0 และ Female เป็น 1  
นอกจากนี้ยังลบข้อมูลที่ smoking\_history เป็น No Info และเพิ่ม column ใหม่คือ smoking\_history\_rank ตามโค้ดที่กำหนดด้านล่าง และสุดท้ายคือเพิ่ม column bmi\_result เพื่อดูว่าค่า bmi ของแต่ละคนอยู่ในเกณฑ์ไหน

df = df\_original.copy()

df = df.dropna()

df['gender'] = df['gender'].replace({'Male': 0, 'Female': 1})

df.sample(5)

df = df[df['smoking\_history'] != 'No Info']

smoking\_rank\_map = {

    'never': 1,

    'former': 2,

    'current': 3,

    'ever': 4

}

df['smoking\_history\_rank'] = df['smoking\_history'].map(smoking\_rank\_map)

def categorize\_bmi(bmi):

    if bmi < 18.5:

        return 'Underweight'

    elif 18.5 <= bmi < 25:

        return 'Normal weight'

    elif 25 <= bmi < 30:

        return 'Overweight'

    else:

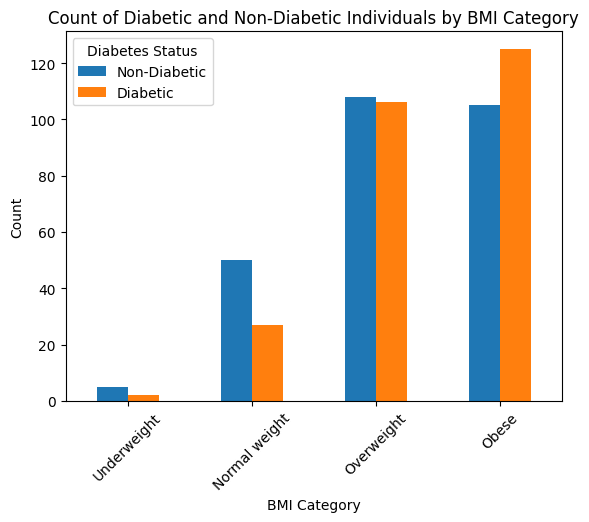
        return 'Obese'

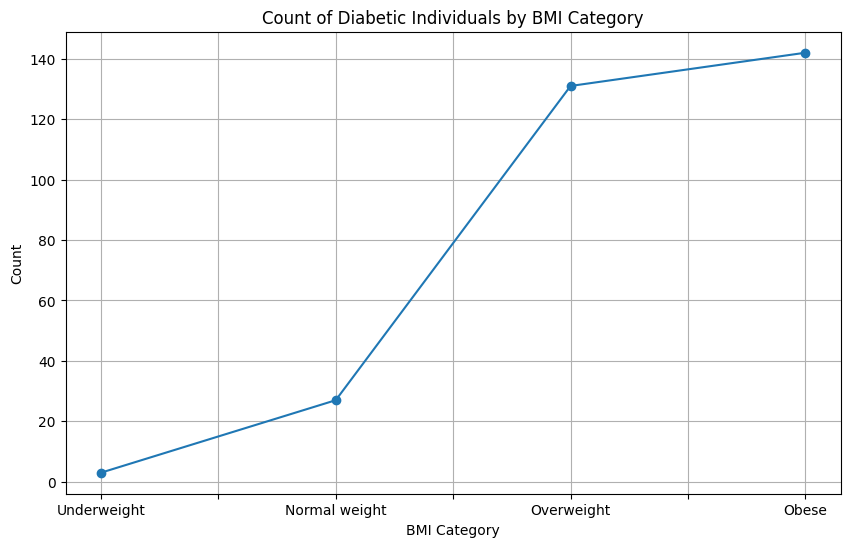
df.dropna(subset=['bmi'], inplace=True)

df['bmi\_result'] = df['bmi'].apply(categorize\_bmi)

3. Data Visualization

ได้ทำการลอง plot ดูความสัมพันธ์ระหว่างตัวแปรต่างๆ (แนบมาพร้อมกับไฟล์นี้) แต่จะขอนำมาแสดงเฉพาะส่วนที่คิดว่าน่าสนใจ ดังนี้





คือความสัมพันธ์ระหว่าง bmi และการเป็นเบาหวานแสดงให้เห็นว่ายิ่ง bmi มากจะทำให้มีแนวโน้มเป็นเบาหวานได้เยอะขึ้น นอกจากนี้ยังมี

A graph of diabetes

Description automatically generated with medium confidence

A graph of a diabetes

Description automatically generated with medium confidence

ที่อาจจะสังเกตุด้วยตาเปล่าได้ยาก จึงต้องใช้การคำนวณทางสถิติมาช่วย

4. Data Analysis

เมื่อทำการหา linear regression model

X = df[["age","gender","hypertension","heart\_disease","smoking\_history\_rank", "bmi", "HbA1c\_level", "blood\_glucose\_level"]]

y = df["diabetes"]

X = X.dropna()

y = y[X.index]

X = sm.add\_constant(X)

model = sm.OLS(y, X).fit()

print(model.summary())

จะได้ผลลัพธ์ดังนี้

OLS Regression Results

==============================================================================

Dep. Variable: diabetes R-squared: 0.587

Model: OLS Adj. R-squared: 0.580

Method: Least Squares F-statistic: 83.96

Date: Tue, 26 Dec 2023 Prob (F-statistic): 8.99e-86

Time: 21:19:45 Log-Likelihood: -136.13

No. Observations: 481 AIC: 290.3

Df Residuals: 472 BIC: 327.9

Df Model: 8

Covariance Type: nonrobust

========================================================================================

coef std err t P>|t| [0.025 0.975]

----------------------------------------------------------------------------------------

const -0.9325 0.134 -6.951 0.000 -1.196 -0.669

age 0.0004 0.001 0.379 0.705 -0.002 0.003

gender 0.0318 0.031 1.037 0.300 -0.028 0.092

hypertension -0.2365 0.033 -7.159 0.000 -0.301 -0.172

heart\_disease -0.2113 0.035 -6.048 0.000 -0.280 -0.143

smoking\_history\_rank -0.0059 0.015 -0.389 0.697 -0.036 0.024

bmi 0.0058 0.002 2.544 0.011 0.001 0.010

HbA1c\_level 0.1420 0.012 11.542 0.000 0.118 0.166

blood\_glucose\_level 0.0032 0.000 11.915 0.000 0.003 0.004

==============================================================================

Omnibus: 13.664 Durbin-Watson: 2.086

เมื่อพิจารณาจากข้อมูลนี้แล้วจะได้ว่าไม่ควรพิจารณา age, gender และ smoking\_history\_rank เนื่องจากมีค่า  
p-value ที่มากกว่า 0.05 จึงไม่ปฎิเสธสมมติฐานหลักที่ว่าสัมประสิทธิ์ควรเป็น 0 (ฉะนั้นสัมประสิทธิ์จึงเป็น 0) ซึ่งถ้าดูจากกราฟที่ plot ไว้ก็จะสังเกตุได้ว่าไม่น่าจะมีความเกี่ยวข้องกัน เช่น

A graph of different colored bars

Description automatically generated with medium confidence

A blue and orange circle with text

Description automatically generated

ที่อายุและเพศดูเหมือนจะไม่มีผลกับการเป็นเบาหวาน

หลังจากนั้นจะนำ column ที่ควรมาคำนวณใหม่จะได้ดังนี้

X = df[["hypertension","heart\_disease", "bmi", "HbA1c\_level", "blood\_glucose\_level"]]

y = df["diabetes"]

X = X.dropna()

y = y[X.index]

X = sm.add\_constant(X)

model = sm.OLS(y, X).fit()

print(model.summary())

OLS Regression Results

==============================================================================

Dep. Variable: diabetes R-squared: 0.586

Model: OLS Adj. R-squared: 0.582

Method: Least Squares F-statistic: 147.9

Date: Tue, 26 Dec 2023 Prob (F-statistic): 1.41e-97

Time: 21:24:23 Log-Likelihood: -150.23

No. Observations: 528 AIC: 312.5

Df Residuals: 522 BIC: 338.1

Df Model: 5

Covariance Type: nonrobust

=======================================================================================

coef std err t P>|t| [0.025 0.975]

---------------------------------------------------------------------------------------

const -0.9101 0.107 -8.470 0.000 -1.121 -0.699

hypertension -0.2327 0.031 -7.430 0.000 -0.294 -0.171

heart\_disease -0.2147 0.031 -6.904 0.000 -0.276 -0.154

bmi 0.0056 0.002 2.561 0.011 0.001 0.010

HbA1c\_level 0.1411 0.012 11.906 0.000 0.118 0.164

blood\_glucose\_level 0.0033 0.000 13.243 0.000 0.003 0.004

==============================================================================

Omnibus: 15.598 Durbin-Watson: 2.042

Prob(Omnibus): 0.000 Jarque-Bera (JB): 14.390

Skew: 0.350 Prob(JB): 0.000750

Kurtosis: 2.596 Cond. No. 1.39e+03

นอกจากนี้ยังสามารถใช้วิธีการทางสถิติอื่นๆ ยกตัวอย่างเช่น

import scipy.stats as stats

# Splitting the dataset into two groups

group1 = df[df['diabetes'] == 1]['bmi']

group2 = df[df['diabetes'] == 0]['bmi']

statistic1, p\_value1 = stats.shapiro(group1)

statistic2, p\_value2 = stats.shapiro(group2)

print(f"statistic1: {statistic1}, p\_value1: {p\_value1}")

print(f"statistic2: {statistic2}, p\_value2: {p\_value2}")

u\_statistic, u\_p\_value = stats.mannwhitneyu(group1.dropna(), group2.dropna())

print("Mann-Whitney U test statistic:", u\_statistic)

print("P-value:", u\_p\_value)

จะได้ผลลัพธ์

statistic1: 0.9422594904899597, p\_value1: 1.3836563894642495e-08

statistic2: 0.9484157562255859, p\_value2: 4.093081784617425e-08

Mann-Whitney U test statistic: 40238.5

P-value: 0.0020398990301492537

ซึ่งสามารถตีความได้ว่าคนที่เป็นเบาหวานและไม่เป็นเบาหวานมีค่า meadian ของ bmi แตกต่างกันหรือไม่ ซึ่งจากข้อมูลที่ได้มาในส่วนแรกจาก Shapiro ค่า p-values ของทั้ง group1 (เป็นเบาหวาน) และ group2 (ไม่เป็นเบาหวาน) มีค่าน้อยกว่า 0.05 แปลว่าไม่มีการแจกแจงปกติ จึงต้องใช้ Mann-Whitney U Test และผลลัพธ์ออกมาน้อยกว่า 0.05 จึงสรุปได้ว่าผู้ที่เป็นเบาหวานและไม่เป็นเบาหวานมี meadian ของ bmi ที่ไม่เท่ากัน