โครงงานเรื่อง Diabetes

โดย

63010791	ยงยุทธ์	แก้วดวงน้อย
63010852	วรวิชญ์	ธรรมารักษ์วัฒนะ
63011017	สุรธันย์	บุญผ่อง
63011018	สุรพัศ	วงศ์ประไพพักตร์

โครงงาน นี้เป็นส่วนหนึ่งของการศึกษา

วิชา 01076032 ELEMENTARY DIFFERENTIAL EQUATIONS AND LINEAR ALGEBRA

เสนอ

รศ.ดร.อรฉัตร จิตต์โสภักตร์

ปีการศึกษา 2564

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

ชื่อโครงงาน : Diabetes

ชื่อผู้จัดทำโครงงาน : ยงยุทธ์ แก้วดวงน้อย, วรวิชญ์ ธรรมารักษ์วัฒนะ, สุรธันย์ บุญผ่อง, สุรพัศ วงศ์ประไพพักตร์

ปีที่จัดทำโครงงาน : 2564

การวิจัยในครั้งนี้มีวัตถุประสงค์เพื่อศึกษา วิเคราะห์ เกี่ยวกับ linear Algebra เพื่อนำมาใช้ในการทำนายโรคเบาหวาน เนื่องจาก ในปัจจุบัน มีผู้คนจำนวนมากที่มีความเสี่ยงที่จะเป็นเบาหวาน หรือเป็นเบาหวาน แต่ไม่ได้เข้ารับการตรวจ อาจด้วยความวิตกกังวล ความหวาดกลัว หรือด้วยเหตุผลใด ๆ ก็ตาม ทางคณะผู้จัดทำจึงตั้งใจที่จะสร้างโมเดลในการทำนายเพื่อเป็นส่วนหนึ่งในการประกอบการตัดสินใจให้กับผู้ใช้ได้ลองตรวจสอบก่อนที่จะไปพบแพทย์ ซึ่งโมเดลจะทำนายผลเพื่อช่วยในการคัดกรองผู้ที่มีความเสี่ยงที่จะเป็นโรคเบาหวาน ให้ได้ไปพบแพทย์ เพื่อตรวจสอบตามขั้นตอนมาตรฐานทางการแพทย์ต่อไป

โดยผลการศึกษาพบว่า ผลลัพธ์ที่ได้ค่อนข้างตรงกับที่คณะผู้จัดทำคาดหวังไว้ตามจุดประสงค์ของการทำโครงงานครั้งนี้ คือโมเดลสามารถทำนายได้โดยมีประสิทธิภาพกว่า 80% ซึ่งทำให้โครงงานนี้มีโอกาสนำไปประยุกต์ใช้ได้ในอนาคต โดยคณะผู้จัดทำเชื่อว่าโครงงานนี้จะสามารถต่อยอดเพื่อนำไปช่วยเหลือได้ต่อไป

คำสำคัญ

Pregnancies หมายถึง จำนวนครั้งที่ตั้งท้อง

Glucose หมายถึง ค่า Glucose หลังจากทำ oral glucose tolerance test (OGTT)

BloodPressure หมายถึง ความดันเลือด (mm/Hg)

BMI หมายถึง ค่า BMI

DiabetesPedigreeFunction หมายถึง ฟังก์ชันที่ให้คะแนนแนวโน้มของโรคเบาหวานตามประวัติครอบครัว

Age หมายถึง อายุ

คำนำ

โครงงานเรื่องนี้จัดทำขึ้นเพื่อช่วยคัดกรองผู้ที่มีความเสี่ยงจะเป็นโรคเบาหวาน สามารถตรวจสอบตามวิธีที่ถูกต้องและน่าเชื่อถือได้ตรงตามมาตรฐานทางการแพทย์ โครงงานเล่มนี้เป็นโครงงานคณิตศาสตร์ ซึ่งจัดทำเพื่อศึกษาเกี่ยวกับการใช้ linear algebra ให้เกิดประโยชน์ เพื่อที่จะสามารถนำมาใช้กับชีวิตประจำวันหรือทำประโยชน์ต่อส่วนร่วมได้

โดยผู้จัดทำได้ออกศึกษาค้นคว้าจากอินเทอร์เน็ต ผู้จัดทำได้ความรู้เรื่องเกี่ยวกับ vector, matrix และอื่น ๆ โดยหวังเป็นอย่างยิ่งว่าโครงงานเล่มนี้จะมีประโยชน์ต่อผู้ที่คิดจะตรวจสอบตัวเองว่าเสี่ยงต่อการเป็นโรคเบาหวานหรือไม่ ขอขอบคุณ

คณะผู้จัดทำ

สารบัญ

บทที่ 1
บทนำ

ที่มาของโครงงาน

ในปัจจุบัน มีผู้คนจำนวนมากที่มีความเสี่ยงที่จะเป็นเบาหวาน หรือเป็นเบาหวาน แต่ไม่ได้เข้ารับการตรวจ อาจด้วยความวิตกกังวล ความหวาดกลัว หรือด้วยเหตุผลใด ๆ ก็ตาม ทางคณะผู้จัดทำจึงตั้งใจที่จะสร้างโมเดลในการทำนายเพื่อเป็นส่วนหนึ่งในการประกอบการตัดสินใจให้กับผู้ใช้ได้ลองตรวจสอบก่อนที่จะไปพบแพทย์ โดยบุคคลที่ต้องการทดสอบเพื่อใช้งานโมเดล จะต้องกรอกข้อมูลต่าง ๆ ได้แก่ Pregnancies, BloodPressure, Glucose, BMI, DiabetesPedigreeFunction, Age ซึ่งโมเดลจะทำนายผลเพื่อช่วยในการคัดกรองผู้ที่มีความเสี่ยงที่จะเป็นโรคเบาหวาน ให้ได้ไปพบแพทย์ เพื่อตรวจสอบตามขั้นตอนมาตรฐานทางการแพทย์ต่อไป

จุดประสงค์ของโครงงาน

1. เพื่อให้โมเดลช่วยทำนายถึงผู้ที่มีความเสี่ยงจะเป็นโรคเบาหวาน เป็นผลประกอบการตัดสินใจ เพื่อให้ผู้ใช้ได้ไปตรวจด้วยวิธีที่ถูกต้องตามมาตรฐานทางการแพทย์
2. สามารถทำ Data Visualization เพื่อให้สามารถวิเคราะห์ข้อมูลได้ง่าย
3. โมเดลในการทำนายที่เราจัดทำขึ้นมาจะมีประสิทธิภาพในด้านความแม่นยำมากกว่า 60 เปอร์เซ็นต์ขึ้นไป
4. สามารถประยุกต์ใช้ความรู้จากเวกเตอร์ และ เมทริกซ์ในโครงงานได้

บทที่ 2

ภาพรวมการออกแบบระบบ

ภาพรวมขั้นตอนการทำงานทั้งหมด

1. หาข้อมูลที่น่าสนใจจากเว็ปไซต์ Kaggle
2. Clearing Data
    2.1. จัดการชุดข้อมูลที่มี NULL
    2.2. ลบบางคอลัมน์ของข้อมูล
    2.3. จัดการชุดข้อมูลที่มี 0 ในส่วนข้อมูลที่ไม่สามารถเป็น 0 ได้
3. Data Visualization
4. Prediction
    4.1. เทรนโมเดล
    4.2. วัดประสิทธิภาพของโมเดล
5. Test
    5.1. ผ่านโมเดล
    5.2. ผ่าน Euclidean Distance

รายละเอียดข้อมูลที่เกี่ยวข้อง

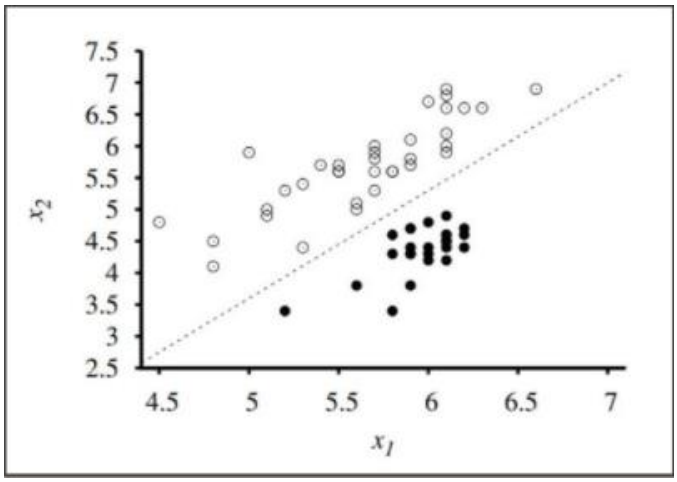
ส่วนเสริมต่าง ๆ ในการเขียนโปรแกรมเพื่อทำ Data Visualization, Prediction สำหรับโปรเจคนี้

Library	คำอธิบาย
NumPy	NumPy เป็นชื่อของ library ที่ใช้ในการคำนวนทางคณิตศาสตร์ในภาษา Python ซึ่งภายในถูกเขียนด้วยภาษา C จึงทำงานได้เร็วและมีประสิทธิภาพ โดย NumPy มีความสามารถในการจัดการกับอาเรย์หลายมิติและข้อมูลแบบเมทริกซ์
Pandas	Pandas นั้นคือ Library หนึ่งในภาษา Python ที่ทำให้เราสามารถจัดการข้อมูลต่างๆได้ง่ายขึ้น เช่น การโหลดข้อมูลไฟล์ CSV เข้ามาแล้วแสดงข้อมูลให้ออกมาในรูปแบบคล้ายกับ Table โดยมีการแบ่งข้อมูลเป็น Row กับ Column เราเรียกสิ่งนี้ว่า Data Frame
Matplotlib	matplotlib เป็นมอดูลสำหรับวาดกราฟที่ใช้งานได้หลากหลายยืดหยุ่น โดยกราฟที่วาดได้ใน matplotlib นั้นมีหลายชนิดด้วยกัน เช่นกราฟเส้นธรรมดา, แผนภูมิแท่ง, แผนภูมิวงกลม, การกระจาย, ฯลฯ
Seaborn	เป็นไลบรารีที่ใช้ Matplotlib เพื่อพล็อตกราฟ มันจะถูกใช้เพื่อให้เห็นภาพการแจกแจงแบบสุ่ม
Scikit-learn	Scikit-learn เป็นโมดูลหนึ่งของภาษา Python เป็นแพ็กเกจที่รวบรวม Library ด้าน Machine Learning เอาไว้ และถูกออกแบบมาให้ทำงานร่วมกับ Library ของภาษา Python อย่าง NumPy และ SciPy ได้ดี

Logistic regression

เนื่องจากตัวแปรตามของข้อมูลในโครงงานนี้คือ เป็นเบาหวาน (1) กับไม่เป็นเบาหวาน (0) แสดงว่าโมเดลนี้ต้องใช้หลักการของ logistic regression

Classification เป็นการกำหนดเส้นแบ่งแยกข้อมูลที่เรามี เพื่อที่จะตอบว่าในอนาคตหากมีข้อมูลเข้ามาใหม่ มันจะเป็นแบบ A หรือแบบ B เช่น



ตัวอย่างข้อมูล

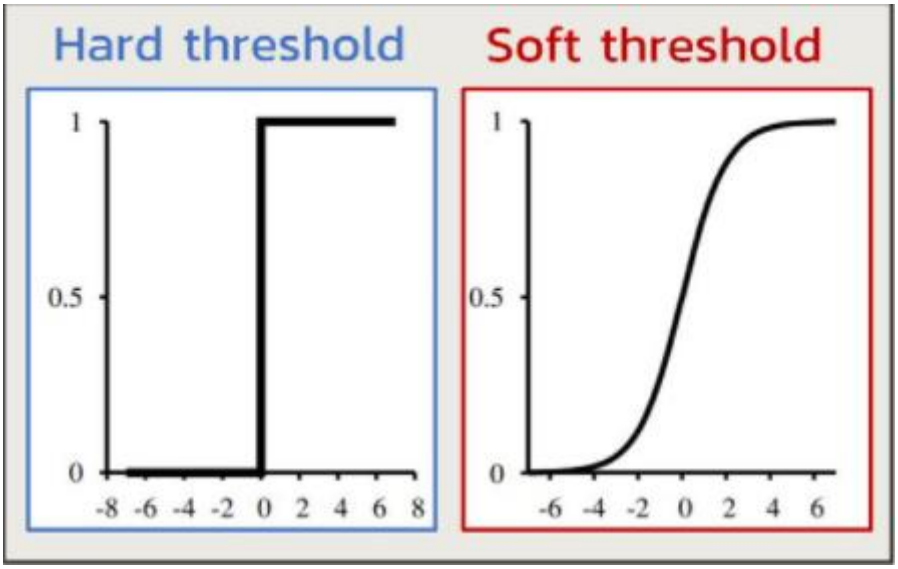
จากรูป หากลากเส้นตรงไปเฉย ๆ จะได้เป็นเส้นประ ซึ่งสามารถแบ่งข้อมูลได้อย่างสมบูรณ์ ซึ่งเรียกกรณีนี้ว่าการแยกแยะความแตกต่างโดยใช้เส้นตรงเส้นเดียว (Linearly separable)
โดยถ้าสมการเส้นตรงเป็น -4.9 + 1.7x – y = 0

- ฝั่งขวาของเส้นตรง -4.9 + 1.7x – y > 0 จะเป็นพื้นที่ของคลาสสีดำ (กำหนดให้เป็น 1)
- ฝั่งซ้ายของเส้นตรง -4.9 + 1.7x – y < 0 จะเป็นพื้นที่ของคลาสสีขาว (กำหนดให้เป็น 0)

แต่วิธีนี้ไม่ค่อยดี เนื่องจาก

$$h_{\mathbf{w}}(\mathbf{x}) = 1 \; if \; \mathbf{w} \cdot \mathbf{x} \geq 0 \; and \; 0 \; otherwise$$
$$h_{\mathbf{w}}(\mathbf{x}) = Threshold(\mathbf{w} \cdot \mathbf{x}) \; where \; Threshold(z) = 1 \; if \; z \geq 0 \; and \; 0 \; otherwise$$

เพราะ Gradient เป็น 0 เกือบทุกจุด (หาค่าไม่ได้ที่ z = 0) ทำให้ weights ของสมมติฐานไม่ได้ถูกอัปเดต จึงทำ Gradient descent ไม่ได้ แล้วพอจะ diff ให้เท่ากับ 0 ทำให้ได้คำตอบออกมามีหลายค่าและทำให้หา Closed form ไม่ได้ ซึ่งเป็นปัญหาทั่วไปของ Hard threshold ซึ่งเป็นการมองคำตอบจากการทำนายเป็นแค่ 0 หรือ 1 ไม่บอก confidence ซึ่งวิธีแก้คือ เราหา Soft threshold มาครอบ Multiple regression



Hard threshold, Soft threshold

ซึ่ง Soft threshold ที่นิยมใช้กันมากคือ Logistic function

$$Logistic(z) = \frac{1}{1 + e^{-z}}$$

ซึ่ง Logistic function จะ diff แบบปกติได้ทุกจุด และคำตอบของเราไม่ใช่แค่ 0 หรือ 1 แต่เป็น confidence ที่จะตอบคลาส 1 แทน เราให้ 0.5 เป็นขอบของการทำนาย

- หากผลลัพธ์ของการทำนายมากกว่า 0.5 ตอบ 1
- หากผลลัพธ์ของการทำนายมากกว่า 0.5 ตอบ 0

หาสมการอัปเดต Gradient descent

$$\begin{aligned}
\frac{\partial}{\partial w_i} Loss(\mathbf{w}) &= \frac{\partial}{\partial w_i}(y - h_{\mathbf{w}}(\mathbf{x}))^2 \\
&= 2(y - h_{\mathbf{w}}(\mathbf{x})) \times \frac{\partial}{\partial w_i}(y - h_{\mathbf{w}}(\mathbf{x})) \\
&= -2(y - h_{\mathbf{w}}(\mathbf{x})) \times g'(\mathbf{w} \cdot \mathbf{x}) \times \frac{\partial}{\partial w_i}\mathbf{w} \cdot \mathbf{x} \\
&= -2(y - h_{\mathbf{w}}(\mathbf{x})) \times g'(\mathbf{w} \cdot \mathbf{x}) \times x_i
\end{aligned}$$

ให้ g′ เป็น logistic function

$$g'(\mathbf{w} \cdot \mathbf{x}) = g(\mathbf{w} \cdot \mathbf{x})\big(1 - g(\mathbf{w} \cdot \mathbf{x})\big) = h_{\mathbf{w}}(\mathbf{x})\big(1 - h_{\mathbf{w}}(\mathbf{x})\big)$$

ดังนั้น สมการสำหรับการอัพเดท weights เพื่อ minimize loss function จะได้เป็นสมการ

$$w_i \leftarrow w_i + \alpha\big(y - h_{\mathbf{w}}(\mathbf{x})\big) \times h_{\mathbf{w}}(\mathbf{x})\big(1 - h_{\mathbf{w}}(\mathbf{x})\big) \times x_i$$

อธิบายขั้นตอนย่อยแต่ละขั้น

1. Clearing Data

เป็นขั้นตอนในการจัดการข้อมูลให้เหมาะสมก่อนนำไปทำ data visualization, machine learning เพื่อให้การวิเคราะห์ข้อมูลมีประสิทธิภาพมากที่สุด

1.1. ข้อมูลเริ่มต้น

```
import io
diabetes_data = pd.read_csv(io.BytesIO(uploaded['diabetes.csv']))
diabetes_data
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0
...	...	...	...	...	...	...	...	...	...
1995	2	75	64	24	55	29.7	0.370	33	0
1996	8	179	72	42	130	32.7	0.719	36	1
1997	6	85	78	0	0	31.2	0.382	42	0
1998	0	129	110	46	130	67.1	0.319	26	1
1999	2	81	72	15	76	30.1	0.547	25	0

2000 rows × 9 columns

1.2. ตรวจสอบประเภทของข้อมูลทั้งหมด

```
diabetes_data.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               2000 non-null   int64
 1   Glucose                   2000 non-null   int64
 2   BloodPressure             2000 non-null   int64
 3   SkinThickness             2000 non-null   int64
 4   Insulin                   2000 non-null   int64
 5   BMI                       2000 non-null   float64
 6   DiabetesPedigreeFunction  2000 non-null   float64
 7   Age                       2000 non-null   int64
 8   Outcome                   2000 non-null   int64
dtypes: float64(2), int64(7)
memory usage: 140.8 KB
```

1.3. ตรวจสอบข้อมูลว่ามีช่องที่เป็น Null หรือไม่ เพื่อตัดชุดข้อมูลนั้นทิ้ง

```
print(diabetes_data.isnull().sum())
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

จากผลการตรวจสอบพบว่าไม่มีช่องที่เป็น Null

1.4. เนื่องจากข้อมูลในช่อง SkinThickness และ Insulin มีข้อมูลเป็น 0 จำนวนมาก เนื่องจากข้อมูลเหล่านี้มีหลักการในการวัดค่าไม่ชัดเจน และทางผู้จัดทำเห็นว่าข้อมูลที่เป็น 0 จะส่งผลกระทบกับโมเดลมากเกินไป จึงทำการลบข้อมูลในคอลัมน์ของ SkinThickness, Insulin

```
[ ] diabetes_data = diabetes_data.drop(columns=['SkinThickness','Insulin'])
    diabetes_data
```

	Pregnancies	Glucose	BloodPressure	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	33.6	0.127	47	1
1	0	84	82	38.2	0.233	23	0
2	0	145	0	44.2	0.630	31	1
3	0	135	68	42.3	0.365	24	1
4	1	139	62	40.7	0.536	21	0
...	...	...	...	...	...	...	...
1995	2	75	64	29.7	0.370	33	0
1996	8	179	72	32.7	0.719	36	1
1997	6	85	78	31.2	0.382	42	0
1998	0	129	110	67.1	0.319	26	1
1999	2	81	72	30.1	0.547	25	0

2000 rows × 7 columns

1.5. ลบชุดข้อมูลอื่น ๆ ที่มีส่วนใดส่วนหนึ่งของข้อมูล BloodPressure, BMI, Glucose เป็น 0

```
diabetes_data = diabetes_data.drop(diabetes_data.index[diabetes_data['BloodPressure']==0])
diabetes_data = diabetes_data.drop(diabetes_data.index[diabetes_data['BMI']==0])
diabetes_data = diabetes_data.drop(diabetes_data.index[diabetes_data['Glucose']==0])
diabetes_data
```

	Pregnancies	Glucose	BloodPressure	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	33.6	0.127	47	1
1	0	84	82	38.2	0.233	23	0
3	0	135	68	42.3	0.365	24	1
4	1	139	62	40.7	0.536	21	0
5	0	173	78	46.5	1.159	58	0
...	...	...	...	...	...	...	...
1995	2	75	64	29.7	0.370	33	0
1996	8	179	72	32.7	0.719	36	1
1997	6	85	78	31.2	0.382	42	0
1998	0	129	110	67.1	0.319	26	1
1999	2	81	72	30.1	0.547	25	0

1888 rows × 7 columns

1.6. ตรวจสอบข้อมูลทั้งหมดอีกครั้ง

```
diabetes_data.info(verbose=True)
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1888 entries, 0 to 1999
Data columns (total 7 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Pregnancies              1888 non-null   int64
 1   Glucose                  1888 non-null   int64
 2   BloodPressure            1888 non-null   int64
 3   BMI                      1888 non-null   float64
 4   DiabetesPedigreeFunction 1888 non-null   float64
 5   Age                      1888 non-null   int64
 6   Outcome                  1888 non-null   int64
dtypes: float64(2), int64(5)
memory usage: 118.0 KB
```

```
print(diabetes_data.isnull().sum())
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

เปรียบเทียบข้อมูลเริ่มต้นและหลังจากจัดการข้อมูล

ข้อมูลเริ่มต้น

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	35	0	33.6	0.127	47	1
1	0	84	82	31	125	38.2	0.233	23	0
2	0	145	0	0	0	44.2	0.630	31	1
3	0	135	68	42	250	42.3	0.365	24	1
4	1	139	62	41	480	40.7	0.536	21	0
...	...	...	...	...	...	...	...	...	...
1995	2	75	64	24	55	29.7	0.370	33	0
1996	8	179	72	42	130	32.7	0.719	36	1
1997	6	85	78	0	0	31.2	0.382	42	0
1998	0	129	110	46	130	67.1	0.319	26	1
1999	2	81	72	15	76	30.1	0.547	25	0

2000 rows × 9 columns

ข้อมูลหลังจากจัดการข้อมูล

	Pregnancies	Glucose	BloodPressure	BMI	DiabetesPedigreeFunction	Age	Outcome
0	2	138	62	33.6	0.127	47	1
1	0	84	82	38.2	0.233	23	0
3	0	135	68	42.3	0.365	24	1
4	1	139	62	40.7	0.536	21	0
5	0	173	78	46.5	1.159	58	0
...	...	...	...	...	...	...	...
1995	2	75	64	29.7	0.370	33	0
1996	8	179	72	32.7	0.719	36	1
1997	6	85	78	31.2	0.382	42	0
1998	0	129	110	67.1	0.319	26	1
1999	2	81	72	30.1	0.547	25	0

1888 rows × 7 columns

คิดเป็นสัดส่วนชุดข้อมูลที่สมบูรณ์ต่อชุดข้อมูลที่ตัดทิ้งคือ

1888 : 112

หรือคิดเป็นข้อมูลสมบูรณ์ 94.4%

2. Data visualization

เป็นการแสดงข้อมูลในรูปแบบรูปภาพนิ่งเพื่อให้ผู้จัดทำสามารถวิเคราะห์ข้อมูลได้ง่ายขึ้น

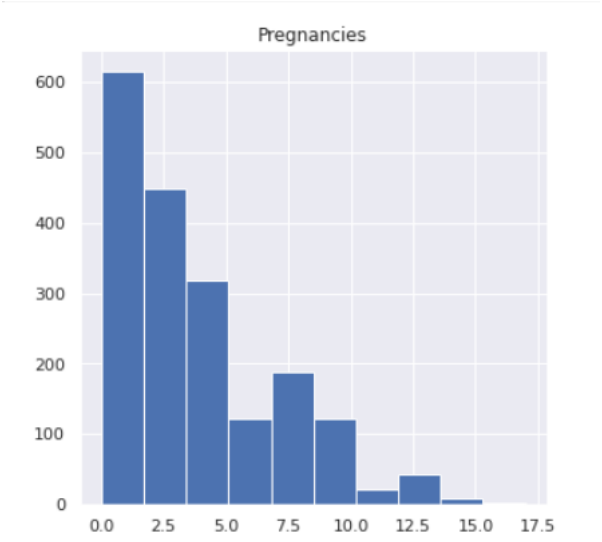
2.1. วิเคราะห์ข้อมูลทางสถิติเบื้องต้น

```
diabetes_data.describe().T
```

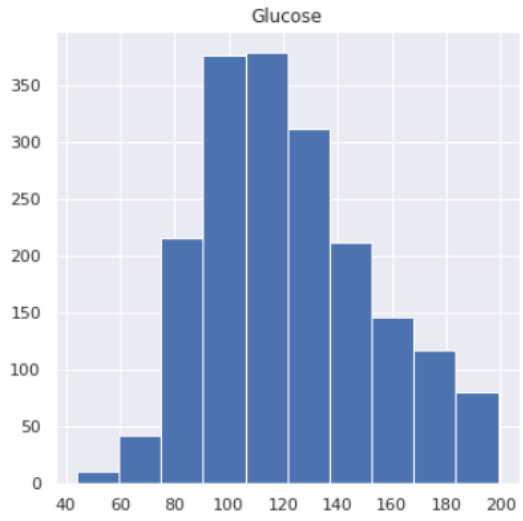
	count	mean	std	min	25%	50%	75%	max
Pregnancies	1888.0	3.742055	3.304971	0.000	1.000	3.00	6.000	17.00
Glucose	1888.0	122.163665	30.784603	44.000	99.000	117.00	142.000	199.00
BloodPressure	1888.0	72.423729	12.255992	24.000	64.000	72.00	80.000	122.00
BMI	1888.0	32.655508	7.196732	18.200	27.500	32.40	36.800	80.60
DiabetesPedigreeFunction	1888.0	0.472931	0.323601	0.078	0.245	0.38	0.624	2.42
Age	1888.0	33.217691	11.780350	21.000	24.000	29.00	40.250	81.00
Outcome	1888.0	0.338453	0.473309	0.000	0.000	0.00	1.000	1.00

ข้อมูลมีการวัดค่า mean, std, min, 25th percentile, 50th percentile, 75th percentile, max ตามลำดับ

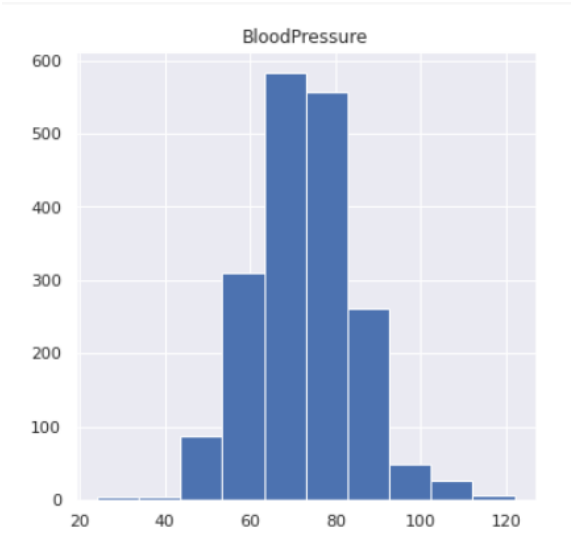
2.2. Bar plot แสดงความถี่ของตัวแปรต่าง ๆ ในข้อมูล



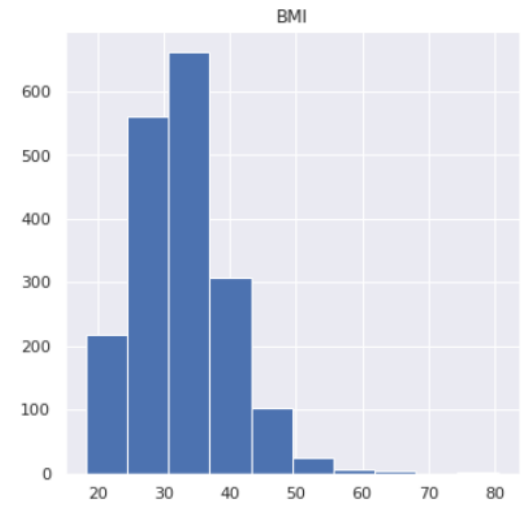
Bar plot แสดงความถี่ของค่า Pregnancies



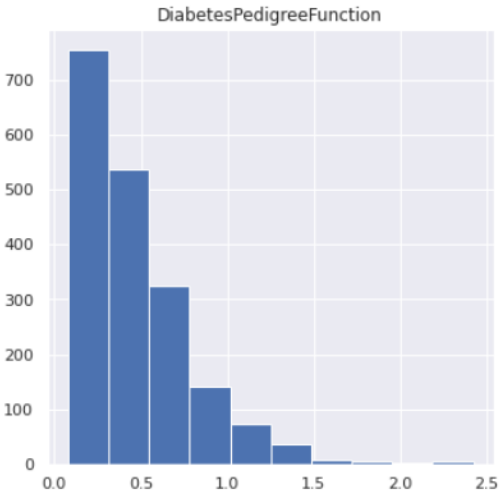
Bar plot แสดงความถี่ของค่า Glucose



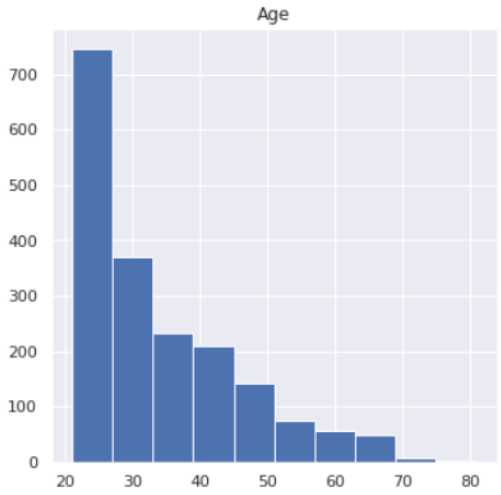
Bar plot แสดงความถี่ของค่า BloodPressure



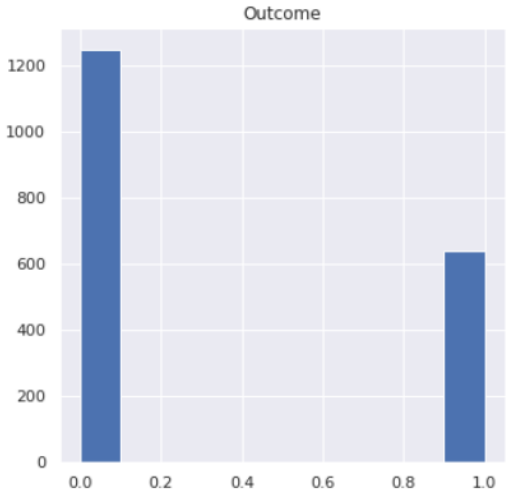
Bar plot แสดงความถี่ของค่า BMI



Bar plot แสดงความถี่ของค่า DiabetesPedigreeFunction

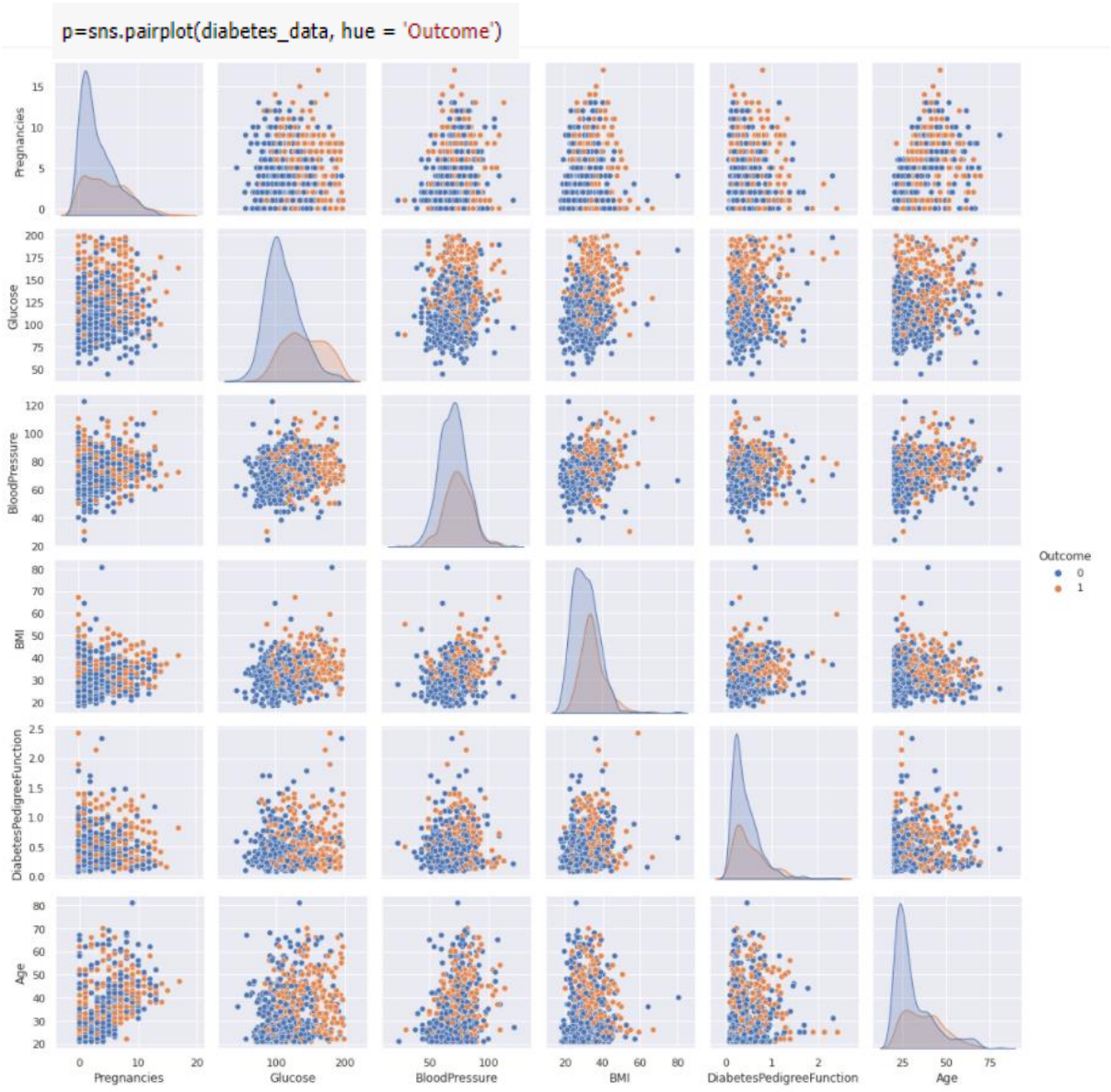


Bar plot แสดงความถี่ของค่า Age



Bar plot แสดงความถี่ของค่า Outcome

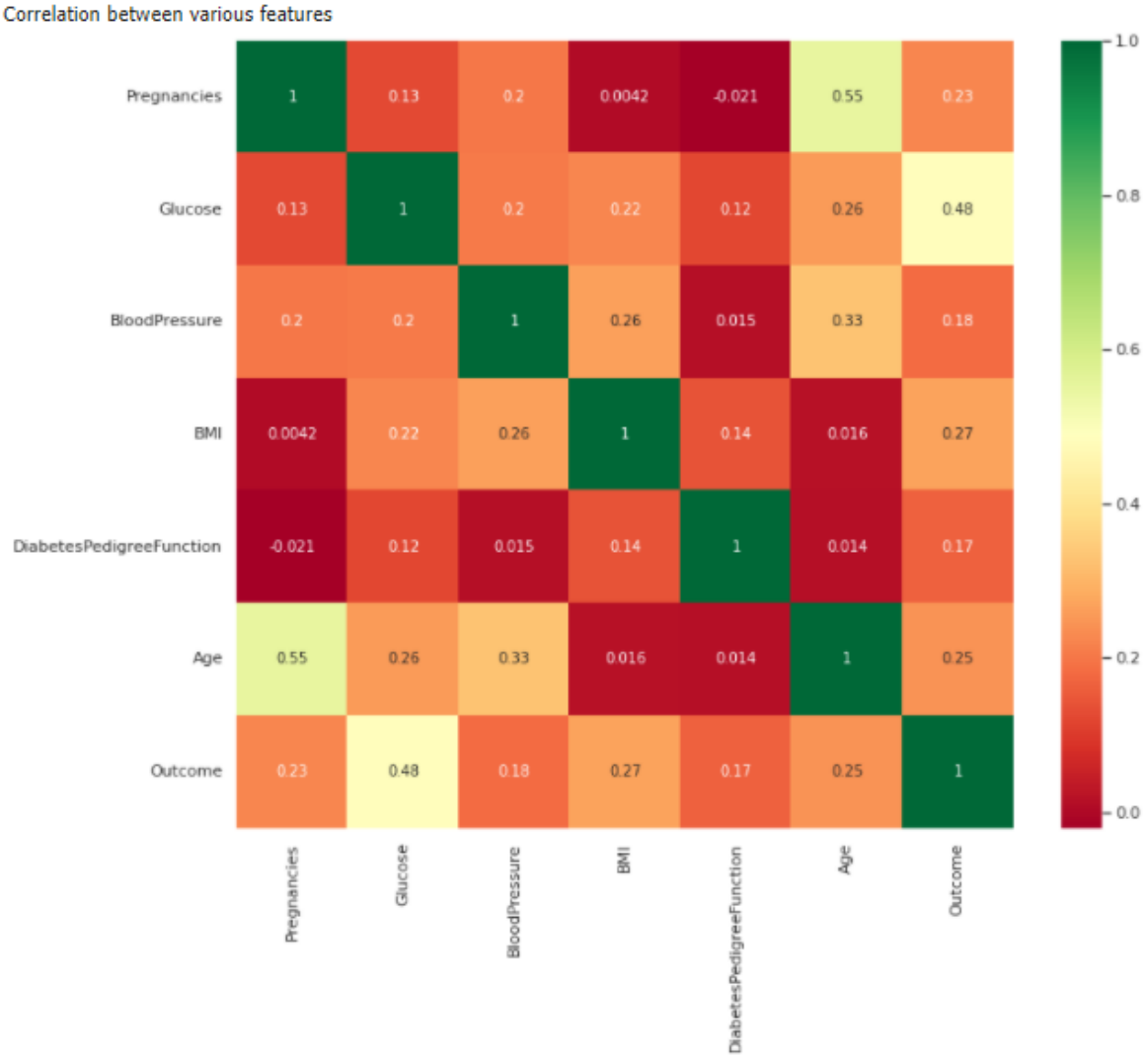
2.3. Pair plot ของตัวแปรต่าง ๆ โดย plot ตาม Outcome เพื่อสังเกตการจับกลุ่มของข้อมูล



Pair plot ของโปรเจค Diabetes

2.4. Heatmap เพื่อสังเกต Correlation ระหว่างตัวแปรในข้อมูล

```
plt.figure(figsize=(12,10))
print('Correlation between various features')
p=sns.heatmap(diabetes_data.corr(), annot=True,cmap ='RdYlGn')
```



Heatmap ของโปรเจค Diabetes

3. Prediction

3.1. แยก x กับ y เป็นตัวแปรต้น และตัวแปรตาม ตามลำดับ ซึ่งตัวแปรต้นจะมี Pregnancies, Glucose, BloodPressure, BMI, DiabetesPedigreeFunction, Age และตัวแปรตามคือ Outcome

```
# independent variable
x = diabetes_data.iloc[ : , : -1]

# dependent variable
y = diabetes_data.Outcome
```

3.2. แบ่งค่าเป็น train set และ test set โดยแบ่งอัตราส่วน 20% สำหรับ test set และแบ่ง 80% สำหรับ train set

โดยให้ random ส่วนที่จะ train และ test ทั้งหมด

```
#แบ่งค่าเป็น training set และ test set
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=326)
```

3.3. เทรนโมเดล โดยใช้หลักการของ logistic regression

```
#ส่วนโมเดล & วัดประสิทธิภาพ
model = LogisticRegression(max_iter=120)
model.fit(x_train, y_train)
```

3.4. ตรวจสอบประสิทธิภาพของโมเดล โดยการหา Accuracy หลังจากทำ Confusion matrix

```
score = model.score(x_test, y_test)
print(f"The model correctly classifies with {score*100:.2f}% accuracy.")
y_pred = model.predict(x_test)
cm = pd.DataFrame(confusion_matrix(y_test, y_pred))
cm
```

```
The model correctly classifies with 80.95% accuracy.
```

	0	1
0	239	25
1	47	67

พบว่าโมเดลมีประสิทธิภาพถึง 80.95% ซึ่งมากกว่าวัตถุประสงค์ที่ตั้งไว้คือมากกว่า 60%

3.5. วิเคราะห์และตีความ Confusion matrix ที่ได้

```
print(f"The model predicts disease with the following accuracy:\n\n\
Healthy people - classified as\n\t\
No disease = {cm[0][0]}/{cm.loc[0].sum()}\n\tDisease = {cm[1][0]}/{cm.loc[0].sum()}\n\n\
\
Sick people - diabetic - classified as\n\t\
No disease = {cm[0][1]}/{cm.loc[1].sum()}\n\tDisease = {cm[1][1]}/{cm.loc[1].sum()}")
```

```
The model predicts disease with the following accuracy:

Healthy people - classified as
        No disease = 239/264
        Disease = 25/264

Sick people - diabetic - classified as
        No disease = 47/114
        Disease = 67/114
```

จะได้ว่า

โมเดลทำการทำนายคนที่ไม่เป็นเบาหวาน และผลคือไม่เป็นเบาหวาน	239	ครั้ง
โมเดลทำการทำนายคนที่ไม่เป็นเบาหวาน แต่ว่าผลคือเป็นเบาหวาน	25	ครั้ง
โมเดลทำการทำนายคนที่เป็นเบาหวาน แต่ว่าผลคือไม่เป็นเบาหวาน	47	ครั้ง
โมเดลทำการทำนายคนที่เป็นเบาหวาน และผลคือเป็นเบาหวาน	67	ครั้ง

เมื่อคำนวณตามสูตรหา Accuracy จาก

$$\text{Accuracy}(\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%$$

จะได้

$$\text{Accuracy}(\%) = \frac{239 + 67}{239 + 67 + 25 + 47} \times 100\% = 80.95\%$$

แปลว่าโปรแกรมทำงานได้ถูกต้อง เพราะข้อมูลจากการคำนวณ ตรงกับข้อมูลที่แสดงผล

4. Test

4.1. รับ input ค่าตัวแปรต่าง ๆ เพื่อนำมาตรวจสอบโมเดล

```
# Pregnancies: Number of times pregnant
#PREGNANCIES = int(diabetes_data.Pregnancies.mean())
PREGNANCIES = int(input("PREGNANCIES : "))

# Plasma glucose concentration over 2 hours in an oral glucose tolerance test
#GLUCOSE = int(diabetes_data.Glucose.mean())
GLUCOSE = int(input("GLUCOSE : "))

# Diastolic blood pressure (mm Hg)
#BLOODPRESSURE = int(diabetes_data.BloodPressure.mean())
BLOODPRESSURE = int(input("BLOODPRESSURE : "))

# Body mass index (weight in kg/(height in m)2)
#BMI = diabetes_data.BMI.mean()
BMI = float(input("BMI : "))

# DiabetesPedigreeFunction: Diabetes pedigree function (a function which scores likelihood of diabetes based on family history)
#DIABETESPEDIGREEFUNCTION = diabetes_data.DiabetesPedigreeFunction.mean()
DIABETESPEDIGREEFUNCTION = float(input("DIABETESPEDIGREEFUNCTION : "))

# Age (years)
#AGE = int(diabetes_data.Age.mean())
AGE = int(input("AGE : "))
```

4.2. รวมข้อมูลที่ input ให้เป็น sample หนึ่ง

```
sample = {
    'Pregnancies': PREGNANCIES,
    'Glucose': GLUCOSE,
    'Bloodpressure': BLOODPRESSURE,
    'Bmi': BMI,
    'Diabetespedigreefunction': DIABETESPEDIGREEFUNCTION,
    'Age':AGE,
}
```

4.3. แสดงผลผลของโมเดลโดยมีตัวแปรจาก sample ในข้อก่อนหน้า

```
if None in sample.values():
    print("Please do not leave any variable with the 'None' value.")
else:
    trial = pd.DataFrame.from_dict(data=sample, orient='index').T
    print(f"Outcome: {model.predict(trial)[0]} - (0 if non-diabetic, 1 if diabetic)")
    print(f"The probability of no disease {model.predict_proba(trial)[0][0]*100:.2f}%\nThe probability of disease {model.predict_proba(trial)[0][1]*100:.2f}%")
```

ตรวจสอบผลที่ได้เทียบกับผลที่ได้จากการเทียบค่า Euclidean Distance

vector ข้อมูลเฉลี่ยของคนที่ไม่เป็นเบาหวาน	3.208967	111.5452	70.85829	31.27174	0.434562	31.15132		
vector ข้อมูลเฉลี่ยของคนเป็นเบาหวาน	4.784038	142.9186	75.48357	35.36025	0.547926	37.25665		
vector ที่นำมาทดสอบ	3	122	72	32.65551	0.472931	33		
Euclidean เทียบ vector ทดสอบ กับ vector เฉลี่ยของคนที่ไม่เป็นเบาหวาน	0.043667	109.3021	1.303509	1.914822	0.001472	3.417614	=	10.76955
Euclidean เทียบ vector ทดสอบ กับ vector เฉลี่ยของคนที่เป็นเบาหวาน	3.18279	437.5888	12.13525	7.315629	0.005624	18.11908	=	21.87115

Euclidean เมื่อเทียบ vector ที่ทดสอบกับ vector เฉลี่ยทั้ง 2 vector ค่า Euclidean ของ vector ทดสอบกับ vector เฉลี่ยที่มีค่าน้อยกว่า แปลว่า vector ทดสอบจะให้ผลเดียวกันกับ vector นั้น ๆ

บทที่ 3
การประยุกต์ใช้ทฤษฎี

การประยุกต์ใช้ทฤษฎีเวกเตอร์

- **Euclidean Distance**

การวัดระยะห่างแบบยูคลิเดียน เป็นการวัดค่าความห่างระหว่างข้อมูล 2 ข้อมูล ในระบบพิกัดคาร์ทีเซียน ที่มาจากทฤษฎีปีทาโกรัส ซึ่งถ้าข้อมูล 2 ตัวนี้มีความคล้ายกันมาก แสดงว่าข้อมูลแต่ละตัว จะอยู่ใกล้กันมาก จะทำให้ค่ายูคลิเดียนมีค่าน้อย ๆ เข้าใกล้ศูนย์ ซึ่งคำนวณได้จากสมการ

$$D_{Euclidean} = \sqrt{\sum_{i=1}^{N}(u_i - v_i)^2}$$

โดย

D(Euclidean) คือ ระยะห่างแบบยูคลิเดียน

u(i) คือ ข้อมูล u ตำแหน่งที่ i

v(i) คือ ข้อมูล v ตำแหน่งที่ i

N คือ จำนวนข้อมูลทั้งหมด

ตัวอย่างการนำมาใช้งานในโปรเจคนี้

จากข้อมูลของคนท้องมาแล้ว 3 คน ค่า Glucose 122 ความดันเลือด 72 โดยมี BMI 32.656 ค่า Diabetes Pedigree Function 0.473 และมีอายุ 33 ปี

vector ข้อมูลเฉลี่ยของคนที่ไม่เป็นเบาหวาน	1.743795	119.0697	70.43875	32.5277	0.481351	28.51161		
vector ข้อมูลเฉลี่ยของคนเป็นเบาหวาน	7.647887	128.2113	76.3036	32.90532	0.456471	42.41628		
vector ที่นำมาทดสอบ	3	122	72	32.65551	0.472931	33		
Euclidean เทียบ vector ทดสอบ กับ vector เฉลี่ยของคนที่ไม่เป็นเบาหวาน	1.578051	8.586918	2.437498	0.016334	7.09E-05	20.14565	=	5.72403
Euclidean เทียบ vector ทดสอบ กับ vector เฉลี่ยของคนที่เป็นเบาหวาน	21.60286	38.57985	18.52097	0.062406	0.000271	88.66624	=	12.93957

พบว่าเวกเตอร์ข้อมูลใกล้เคียงกับเวกเตอร์เฉลี่ยของคนที่ไม่เป็นเบาหวานมากกว่า จึงสามารถตีความได้คร่าว ๆ ว่าบุคคลนี้ ไม่ได้เป็นเบาหวาน

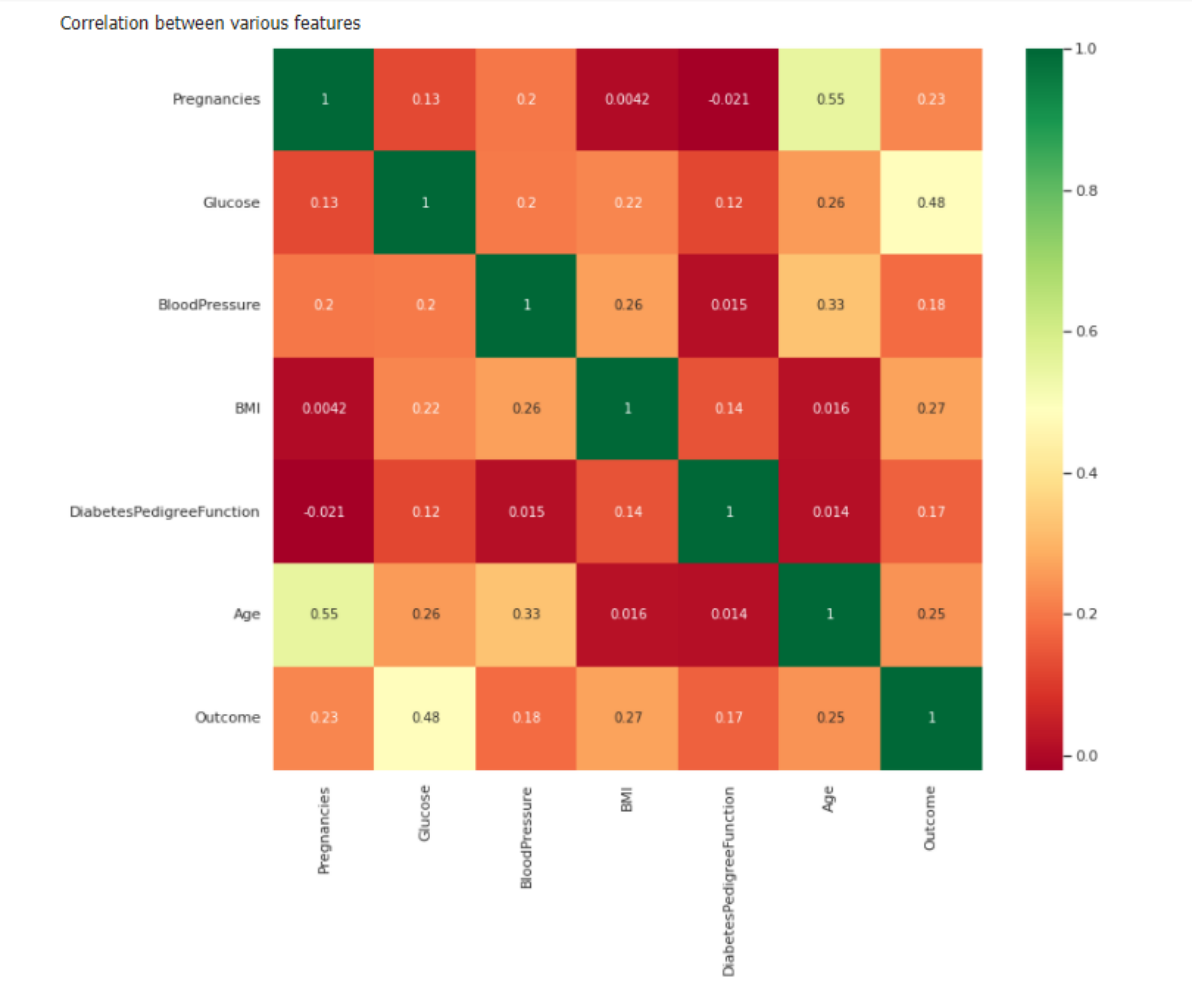
การประยุกต์ใช้ทฤษฎีเมทริกซ์

- **Heatmap**

ใช้นำเสนอข้อมูลแนวโน้มหรือความสัมพันธ์ระหว่างตัวแปรต่าง ๆ แบบง่าย ๆ โดยใช้ คู่สี และความเข้มข้นของสีแทนปริมาณหรือความถี่ ซึ่งในที่นี้ทางกลุ่มเราได้นำมาใช้ในการวิเคราะห์ความสัมพันธ์ระหว่างคู่ตัวแปรทั้งหมด โดยการวิเคราะห์ข้อมูลจาก Heatmap นั้น สามารถทำได้จากการดูค่า Correlation ของคู่ตัวแปรที่ช่องนั้น ๆ

- ถ้าค่า Correlation เข้าใกล้ -1 แปลว่าข้อมูลมีความสัมพันธ์ในทิศทางตรงข้ามกัน
- ถ้าค่า Correlation เข้าใกล้ 0 แปลว่าข้อมูลไม่มีความสัมพันธ์กัน
- ถ้าค่า Correlation เข้าใกล้ 1 แปลว่าข้อมูลมีความสัมพันธ์ในทิศทางเดียวกัน

ตัวอย่างการนำมาใช้งานในโปรเจคนี้

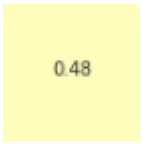


Heatmap ของโปรเจค diabetes

จาก Heatmap ข้างต้น สามารถนำมาวิเคราะห์ความสัมพันธ์ของตัวแปรต่าง ๆ ได้คร่าว ๆ ดังนี้

ค่า Correlation ของ BMI และ Age

จากค่า Correlation ในข้างต้น สามารถบอกได้ว่า BMI กับ Age ไม่ได้มีความสัมพันธ์ต่อกัน



ค่า Correlation ของ Glucose และ Outcome

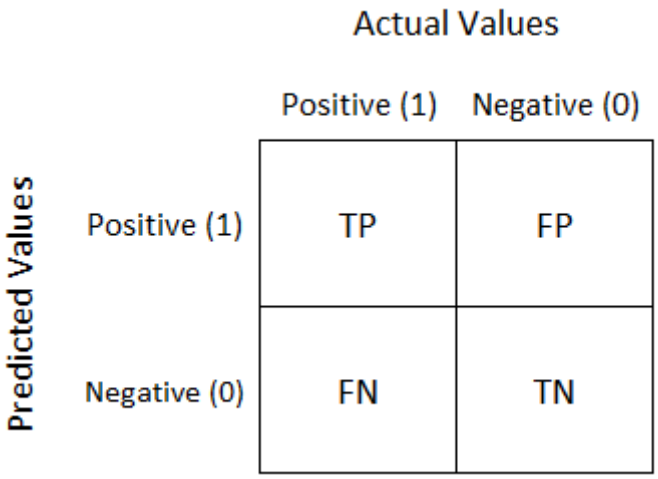
จากค่า Correlation ในข้างต้น สามารถบอกได้ว่า Glucose และ Outcome ไปในทิศทางเดียวกัน คือ ยิ่งมีค่า Glucose มาก ยิ่งทำให้โอกาสที่จะเป็นเบาหวานมากขึ้น

ค่า Correlation ของ BMI และ Outcome

จากค่า Correlation ในข้างต้น สามารถบอกได้ว่า BMI และ Outcome ค่อนข้างไปในทิศทางเดียวกัน คือ ยิ่งมีค่า BMI มาก ยิ่งทำให้โอกาสที่จะเป็นเบาหวานมากขึ้น

- **Confusion matrix**

Confusion Matrix คือตารางสำคัญในการวัดความสามารถของ machine learning ในการแก้ปัญหา classification



ตัวอย่าง Confusion Matrix ขนาด 2x2

True Positive (TP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” และมีค่าเป็น “จริง”
True Negative (TN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” และมีค่า “ไม่จริง”
False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” แต่มีค่าเป็น “ไม่จริง”
False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” แต่มีค่าเป็น “จริง”
โดยนำมาหา Accuracy ได้จาก

$$\text{Accuracy}(\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%$$

ตัวอย่างการนำมาใช้งานในโปรเจคนี้
นำมาใช้ในการหา Accuracy ของโมเดล

	0	**1**
0	239	25
1	47	67

Confusion Matrix ขนาด 2x2 ของโปรเจค diabetes

นำมาหา Accuracy ได้

$$\text{Accuracy}(\%) = \frac{239 + 67}{239 + 67 + 25 + 47} \times 100\% = 80.95\%$$

บทที่ 4

ผลการทดลอง

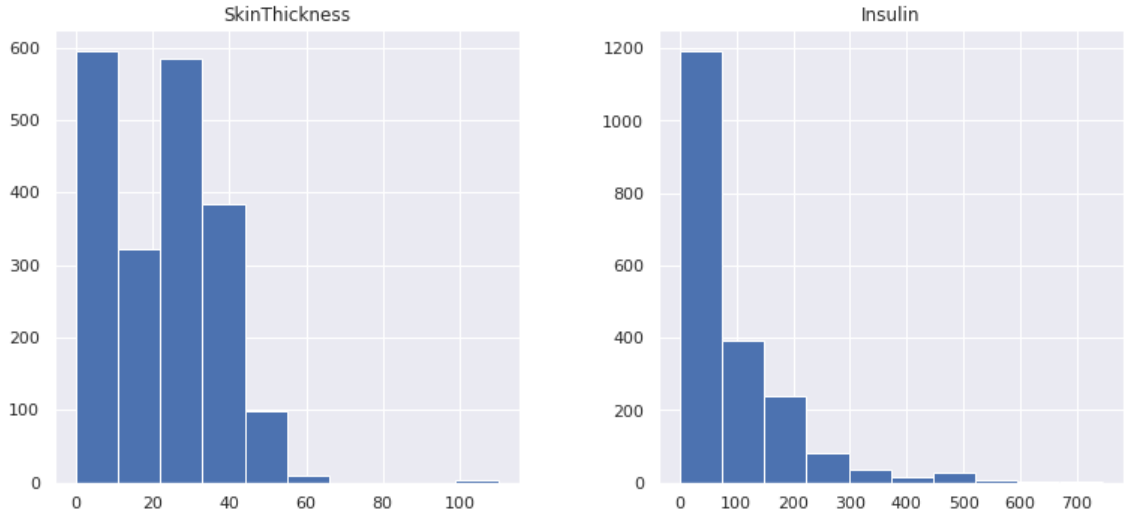
วิเคราะห์ผลหลังจากการ Clearing Data

- ข้อมูลมีจำนวน 2000 ชุด แต่ละชุดข้อมูลประกอบไปด้วยข้อมูล Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age, Outcome
- ค่า std ของข้อมูล Insulin และข้อมูล SkinThickness ค่อนข้างเยอะเป็นเพราะชุดข้อมูลส่วนหนึ่งมีปัญหา ทำให้ค่าค่อนข้างแกว่งมาก

	count	mean	std	min	25%	50%	75%	max
Pregnancies	2000.0	3.70350	3.306063	0.000	1.000	3.000	6.000	17.00
Glucose	2000.0	121.18250	32.068636	0.000	99.000	117.000	141.000	199.00
BloodPressure	2000.0	69.14550	19.188315	0.000	63.500	72.000	80.000	122.00
SkinThickness	2000.0	20.93500	16.103243	0.000	0.000	23.000	32.000	110.00
Insulin	2000.0	80.25400	111.180534	0.000	0.000	40.000	130.000	744.00
BMI	2000.0	32.19300	8.149901	0.000	27.375	32.300	36.800	80.60
DiabetesPedigreeFunction	2000.0	0.47093	0.323553	0.078	0.244	0.376	0.624	2.42
Age	2000.0	33.09050	11.786423	21.000	24.000	29.000	40.000	81.00
Outcome	2000.0	0.34200	0.474498	0.000	0.000	0.000	1.000	1.00

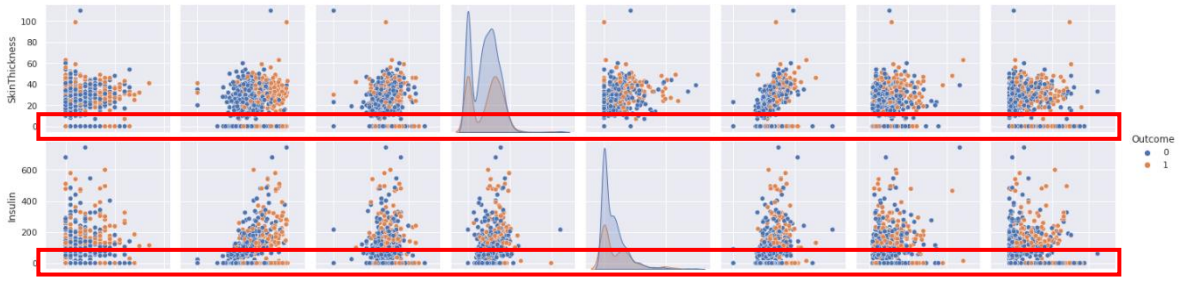
ค่าทางสถิติก่อนการจัดการข้อมูล

- Bar plot ส่วนของ SkinThickness และ Insulin มีค่าที่เป็น 0 ค่อนข้างมาก



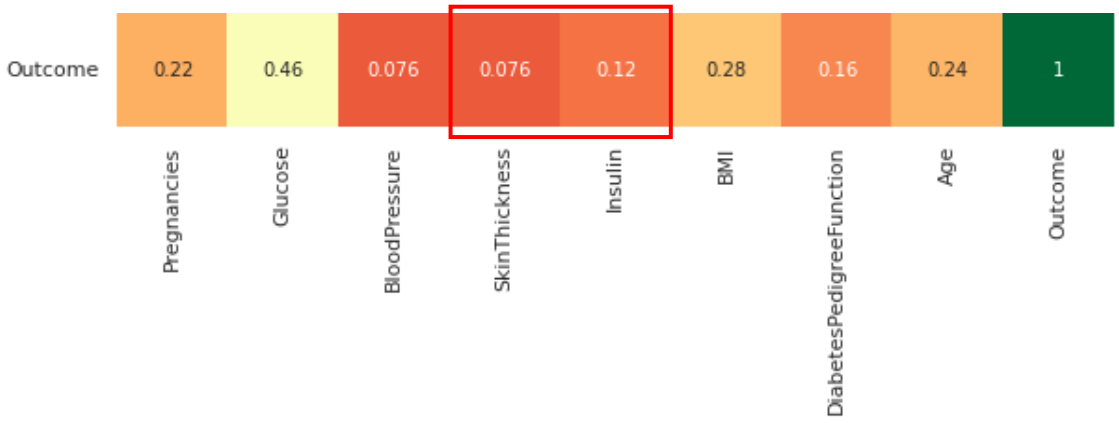
Bar plot ของข้อมูล SkinThickness และ Insulin

- จาก pair plot พบข้อมูลที่ใช้แกน SkinThickness และ Insulin มีข้อมูลที่เกาะกลุ่มตรงที่ค่าเป็น 0 อย่างชัดเจน



Pair plot ของข้อมูล SkinThickness และ Insulin

- จาก Heatmap พบว่า SkinThickness และ Insulin แทบไม่มีผลต่อ Outcome จึงสามารถตัดทิ้งได้



Heatmap เมื่อสังเกต Correlation ระหว่างข้อมูล SkinThickness และ Insulin เทียบกับ Outcome

- เมื่อเทรนโมเดลด้วยข้อมูลข้างต้น พบว่า Accuracy ของโมเดล ทำได้เพียง 75.5%

The model correctly classifies with 75.50% accuracy.

	0	1
0	232	28
1	70	70

Accuracy ของข้อมูลก่อน clearing data

The model correctly classifies with 80.95% accuracy.

	0	1
0	239	25
1	47	67

Accuracy ของข้อมูลหลัง clearing data

แปลว่าการจัดการข้อมูล ทำให้การเทรนโมเดลมีประสิทธิภาพมากขึ้น

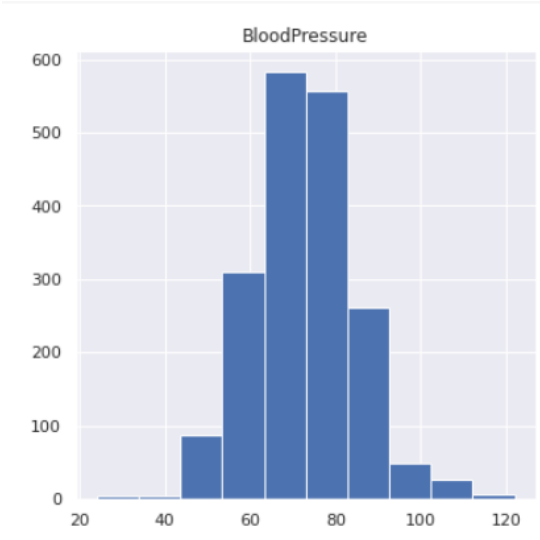
วิเคราะห์ผลหลังจากทำ Data Visualization

- ข้อมูลทางสถิติเบื้องต้น

	count	mean	std	min	25%	50%	75%	max
Pregnancies	1888.0	3.742055	3.304971	0.000	1.000	3.00	6.000	17.00
Glucose	1888.0	122.163665	30.784603	44.000	99.000	117.00	142.000	199.00
BloodPressure	1888.0	72.423729	12.255992	24.000	64.000	72.00	80.000	122.00
BMI	1888.0	32.655508	7.196732	18.200	27.500	32.40	36.800	80.60
DiabetesPedigreeFunction	1888.0	0.472931	0.323601	0.078	0.245	0.38	0.624	2.42
Age	1888.0	33.217691	11.780350	21.000	24.000	29.00	40.250	81.00
Outcome	1888.0	0.338453	0.473309	0.000	0.000	0.00	1.000	1.00

ข้อมูลมีการวัดค่า mean, std, min, 25th percentile, 50th percentile, 75th percentile, max ตามลำดับ

- Bar plot

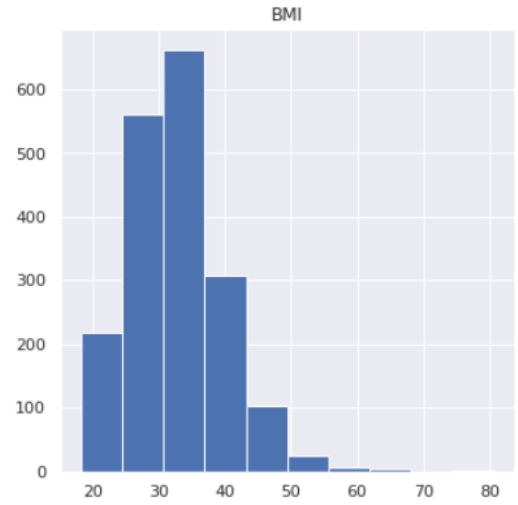


Bar plot แสดงความถี่ของค่า BloodPressure

ช่วงอายุ	ค่าความดันตัวบน/ค่าความดันตัวล่าง (โดยเฉลี่ย)
วัยทารก	ไม่ควรเกิน 90/60
วัยเด็กเล็ก (3 – 6 ปี)	ไม่ควรเกิน 110/70
วัยเด็กโต (7 – 17 ปี)	ไม่ควรเกิน 120/80
วัยทำงาน (18 ปีขึ้นไป)	ไม่ควรเกิน 140/90
วัยสูงอายุ (60 ปีขึ้นไป)	ไม่ควรเกิน 160/90

รูปภาพเกณฑ์ความดันเลือดปกติจาก
https://allwellhealthcare.com/normal-blood-pressure-range/

จาก Bar Plot เทียบ BloodPressure ของข้อมูล เทียบกับเกณฑ์พบว่า BloodPressure ของข้อมูล ส่วนใหญ่อยู่ในเกณฑ์ปกติ

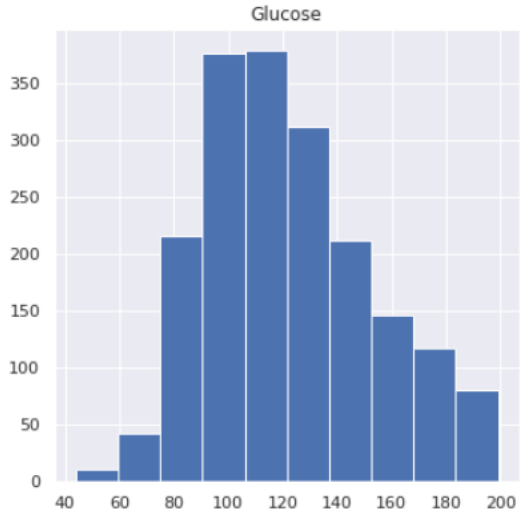


Bar plot แสดงความถี่ของค่า BMI

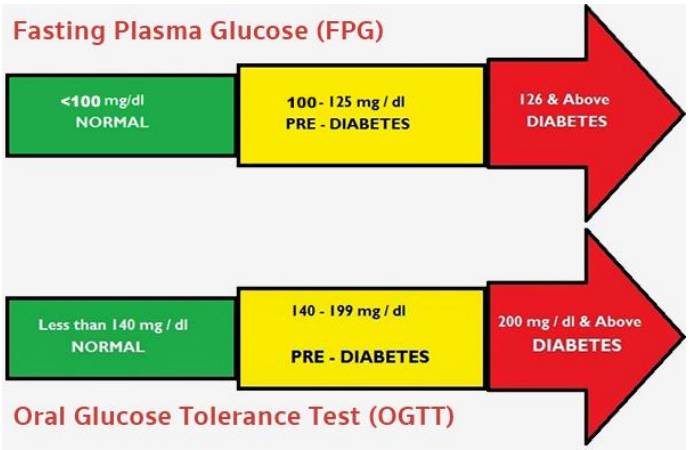
BMI	Nutritional status
Below 18.5	Underweight
18.5–24.9	Normal weight
25.0–29.9	Pre-obesity
30.0–34.9	Obesity class I
35.0–39.9	Obesity class II
Above 40	Obesity class III

รูปภาพเกณฑ์ BMI ปกติจาก shorturl.at/lFIW4

จาก Bar Plot เทียบ BMI ของข้อมูล เทียบกับเกณฑ์พบว่า BMI ของข้อมูล ส่วนใหญ่อยู่ในเกณฑ์ Pre-obesity กับ Obesity class I



Bar plot แสดงความถี่ของค่า Glucose

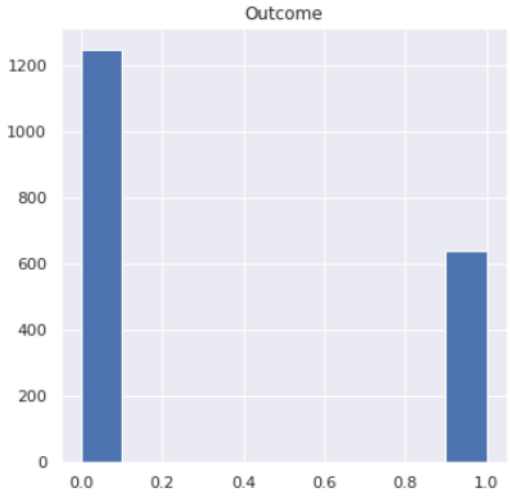


รูปภาพเกณฑ์ BMI ปกติจาก shorturl.at/dfCL7

จาก Bar plot เทียบ Glucose ของข้อมูล เทียบกับเกณฑ์พบว่า Glucose ของข้อมูล ส่วนใหญ่อยู่ในเกณฑ์ Pre-diabetes

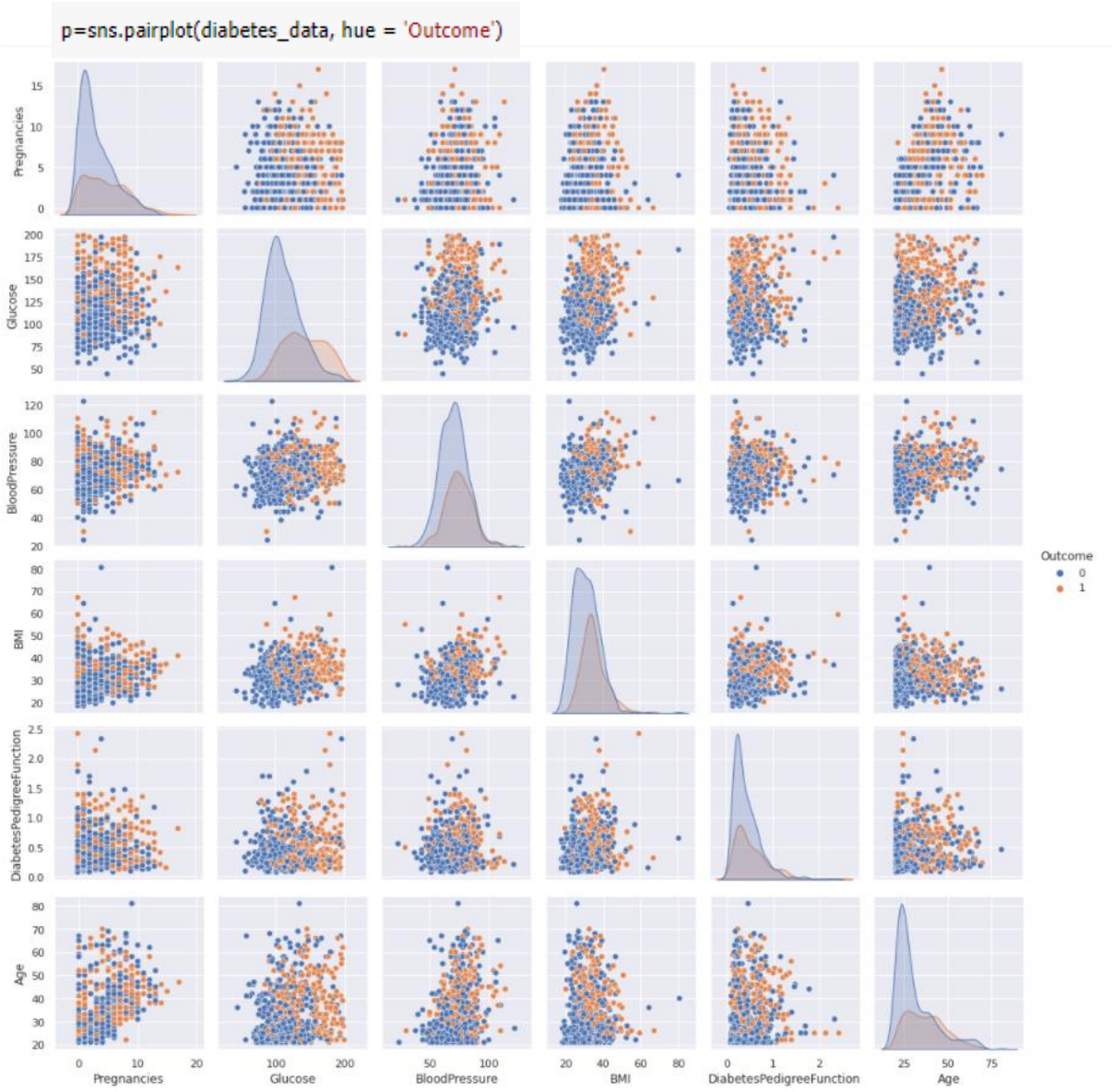
นอกจากนั้น Bar plot ของ Age, Pregnancies ขึ้นอยู่กับแต่ละบุคคล ไม่มีเกณฑ์ปกติ

จากที่กล่าวมาข้างต้น สอดคล้องกับ Bar plot ของ Outcome ที่ข้อมูลส่วนใหญ่ Outcome เป็น 0 ซึ่งเป็นข้อมูลของคนที่ไม่เป็นเบาหวาน ซึ่งข้อมูลทั้งหมดสอดคล้องกัน แปลว่าโดยภาพรวมข้อมูลปกติ



Bar plot แสดงความถี่ของค่า Outcome

- Pair plot



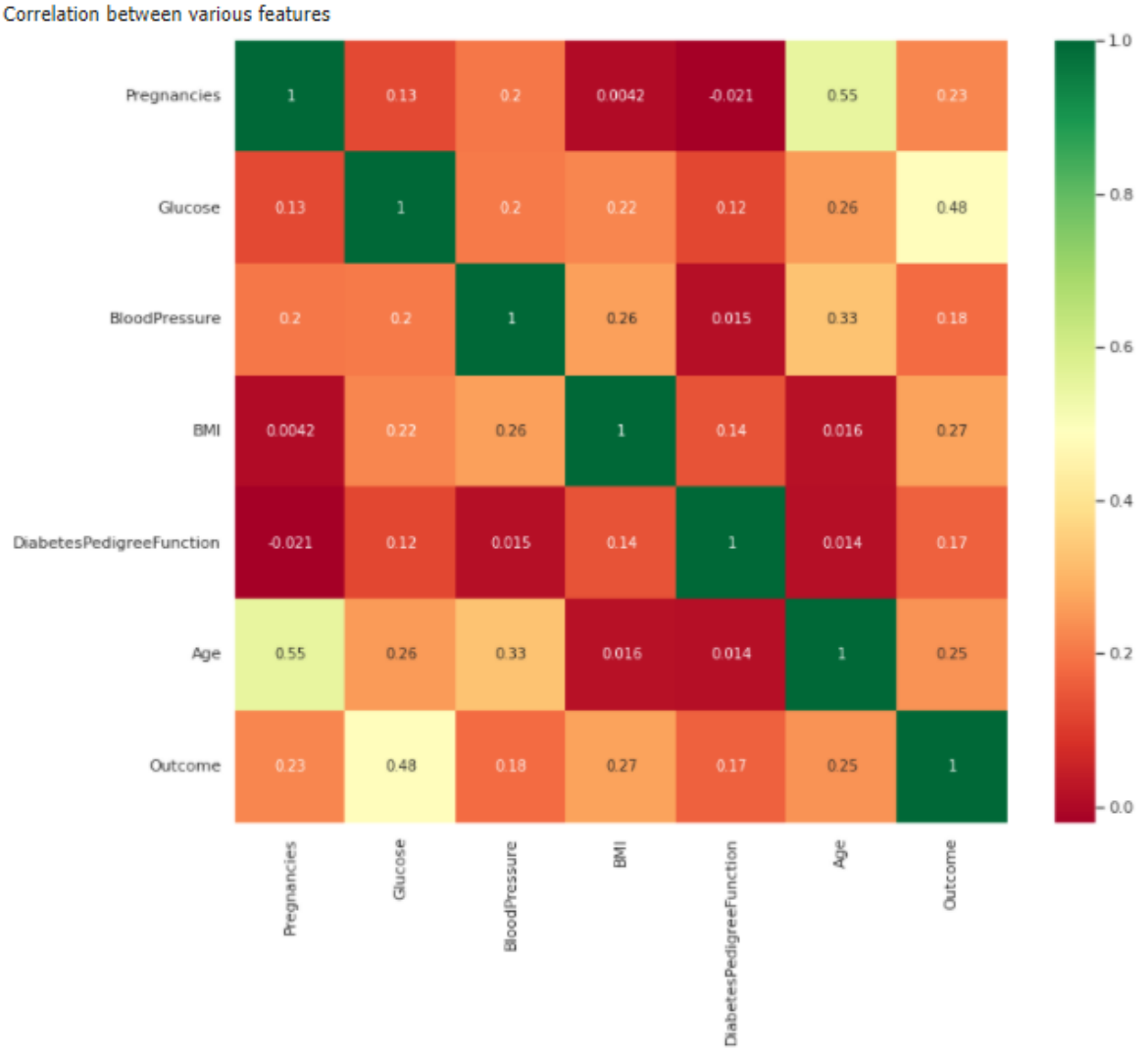
Pair plot ของโปรเจค Diabetes

จาก Pair plot จะเห็นข้อมูลที่ใช้แกน Glucose พบว่าข้อมูลของคนที่เป็นเบาหวานเกาะกลุ่มอยู่อย่างชัดเจน

แปลว่า Glucose เป็นปัจจัยสำคัญ ที่ทำให้เป็นเบาหวาน เพราะว่ายิ่งค่า Glucose สูง ก็มีโอกาสเป็นเบาหวานมากขึ้น ยิ่งค่า Glucose ต่ำ ก็มีโอกาสเป็นเบาหวานน้อยลง

- Heatmap

```
plt.figure(figsize=(12,10))
print('Correlation between various features')
p=sns.heatmap(diabetes_data.corr(), annot=True,cmap ='RdYlGn')
```



Heatmap ของโปรเจค Diabetes

จาก Heatmap เมื่อสังเกตค่า Correlation ระหว่างตัวแปรต่าง ๆ กับ Outcome พบว่า Correlation เรียงจากมากไปน้อย Glucose, BMI, Age, Pregnancies, BloodPressure, DiabetesPedigreeFunction ดังนี้ ซึ่งสอดคล้องกับ Pair plot ที่ผ่านมา ที่กลุ่มผู้เป็นเบาหวานที่ใช้แกนของ Glucose จะสังเกตได้ชัดเจนที่สุด

วิเคราะห์ผลของโมเดลที่ได้

```
score = model.score(x_test, y_test)
print(f"The model correctly classifies with {score*100:.2f}% accuracy.")
y_pred = model.predict(x_test)
cm = pd.DataFrame(confusion_matrix(y_test, y_pred))
cm
```

```
The model correctly classifies with 80.95% accuracy.
```

	0	1
0	239	25
1	47	67

จากโมเดลมีประสิทธิภาพ 80.95% ซึ่งมากกว่าวัตถุประสงค์ที่ตั้งไว้คือมากกว่า 60%

เมื่อวิเคราะห์และตีความ Confusion matrix ที่ได้

```
print(f"The model predicts disease with the following accuracy:\n\n\
Healthy people - classified as\n\t\
No disease = {cm[0][0]}/{cm.loc[0].sum()}\n\tDisease = {cm[1][0]}/{cm.loc[0].sum()}\n\n\
\
Sick people - diabetic - classified as\n\t\
No disease = {cm[0][1]}/{cm.loc[1].sum()}\n\tDisease = {cm[1][1]}/{cm.loc[1].sum()}")
```

```
The model predicts disease with the following accuracy:

Healthy people - classified as
        No disease = 239/264
        Disease = 25/264

Sick people - diabetic - classified as
        No disease = 47/114
        Disease = 67/114
```

จะได้ว่า

โมเดลทำการทำนายคนที่ไม่เป็นเบาหวาน และผลคือไม่เป็นเบาหวาน	239	ครั้ง
โมเดลทำการทำนายคนที่ไม่เป็นเบาหวาน แต่ว่าผลคือเป็นเบาหวาน	25	ครั้ง
โมเดลทำการทำนายคนที่เป็นเบาหวาน แต่ว่าผลคือไม่เป็นเบาหวาน	47	ครั้ง
โมเดลทำการทำนายคนที่เป็นเบาหวาน และผลคือเป็นเบาหวาน	67	ครั้ง

เมื่อคำนวณตามสูตรหา Accuracy จาก

$$\text{Accuracy}(\%) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%$$

จะได้

$$\text{Accuracy}(\%) = \frac{239 + 67}{239 + 67 + 25 + 47} \times 100\% = 80.95\%$$

แปลว่าโปรแกรมทำงานได้ถูกต้อง เพราะข้อมูลจากการคำนวณ ตรงกับข้อมูลที่แสดงผล

Sample Case #1

ผู้หญิงอายุ 33 ปี ท้องมาแล้ว 3 คน มีค่า Glucose เมื่อทำ oral glucose tolerance test (OGTT) เป็น 122 มีความดันเลือด 72 mm/Hg มี BMI 32.6555 และ Diabetes Pedigree Function เป็น 0.47293 ผลจากโมเดลออกมาเป็น

Outcome: 0 - (0 if non-diabetic, 1 if diabetic)
The probability of no disease 71.85%
The probability of disease 28.15%

ผลจากการทำ Euclidean Distance เป็น

vector ข้อมูลเฉลี่ยของคนที่ไม่เป็นเบาหวาน	3.208967	111.5452	70.85829	31.27174	0.434562	31.15132		
vector ข้อมูลเฉลี่ยของคนเป็นเบาหวาน	4.784038	142.9186	75.48357	35.36025	0.547926	37.25665		
vector ที่นำมาทดสอบ	3	122	72	32.6555	0.47293	33		
Euclidean เทียบ vector ทดสอบ กับ vector เฉลี่ยของคนที่ไม่เป็นเบาหวาน	0.043667	109.3021	1.303509	1.914799	0.001472	3.417614	=	10.76955
Euclidean เทียบ vector ทดสอบ กับ vector เฉลี่ยของคนที่เป็นเบาหวาน	3.18279	437.5888	12.13525	7.315675	0.005624	18.11908	=	21.87115

พบว่าไม่เป็นเบาหวานเหมือนกัน

Sample Case #2

ผู้ชายอายุ 19 ปี มีค่า Glucose เมื่อทำ oral glucose tolerance test (OGTT) เป็น 201 มีความดันเลือด 87 mm/Hg มี BMI 37.6555 และ Diabetes Pedigree Function เป็น 0.01274 ผลจากโมเดลออกมาเป็น

Outcome: 1 - (0 if non-diabetic, 1 if diabetic)
The probability of no disease 28.56%
The probability of disease 71.44%

ผลจากการทำ Euclidean Distance เป็น

vector ข้อมูลเฉลี่ยของคนที่ไม่เป็นเบาหวาน	3.208967	111.5452	70.85829	31.27174	0.434562	31.15132		
vector ข้อมูลเฉลี่ยของคนเป็นเบาหวาน	4.784038	142.9186	75.48357	35.36025	0.547926	37.25665		
vector ที่นำมาทดสอบ	0	201	87	37.6555	0.01274	19		
Euclidean เทียบ vector ทดสอบ กับ vector เฉลี่ยของคนที่ไม่เป็นเบาหวาน	10.29747	8002.155	260.5549	40.75243	0.177934	147.6546	=	91.98691
Euclidean เทียบ vector ทดสอบ กับ vector เฉลี่ยของคนที่เป็นเบาหวาน	22.88702	3373.446	132.6282	5.268171	0.286425	333.3053	=	62.19181

พบว่าเป็นเบาหวานเหมือนกัน

บทที่ 5
สรุปผลและข้อเสนอแนะ

สรุปผลการทดลอง

จากผลการทดลองสรุปได้ว่า ผลลัพธ์ที่ได้ค่อนข้างตรงกับที่คณะผู้จัดทำคาดหวังไว้ตามจุดประสงค์ของการทำโครงงานครั้งนี้ ซึ่งทำให้โครงงานนี้มีโอกาสขยับขยายและประยุกต์ใช้ได้ในอนาคต เพราะปัจจุบันยังมีผู้คนจำนวนมากที่ยังไม่ได้รับการตรวจเลือดอย่างสม่ำเสมอ ทำให้ไม่อาจทราบได้ว่าตนเองเป็นผู้ที่เสี่ยงจะมีโอกาสเป็นโรคเบาหวานหรือไม่ โครงงานนี้จะสามารถต่อยอดเพื่อนำไปช่วยเหลือกลุ่มคนเหล่านี้ได้ต่อไป

ข้อเสนอแนะ

- เพิ่มประเภทของข้อมูลและจำนวนข้อมูลที่นำมาทดสอบเพื่อที่จะสามารถนำมาช่วยวิเคราะห์และเทรนโมเดล เพื่อให้ได้ความแม่นยำของโมเดลมากขึ้น
- ข้อมูลที่เก็บเพื่อตรวจสอบ เป็นข้อมูลที่ไม่สามารถตอบได้ทันที เช่น ความดันเลือด, หรือค่า Glucose จากการทำ oral glucose tolerance test (OGTT) ถ้าอยากให้สามารถเข้าถึงประชาชนทั่วไปได้ควรจะเก็บข้อมูลประเภทที่สามารถตอบได้ง่ายกว่านี้
- ควรนำข้อมูลนี้ไปต่อยอดนำมาแสดงผลให้เห็นชัดเจนมากขึ้น

เอกสารอ้างอิง

- https://numpy.org/
- https://pandas.pydata.org/
- https://matplotlib.org/
- https://seaborn.pydata.org/
- https://scikit-learn.org/stable/
- https://allwellhealthcare.com/normal-blood-pressure-range/
- https://www.euro.who.int/en/health-topics/disease-prevention/nutrition/a-healthy-lifestyle/body-mass-index-bmi
- https://medthai.com/ทดสอบความทนต่อน้ำตาล/
- https://www.kaggle.com/johndasilva/diabetes
- https://web.facebook.com/AiByNeto/posts/119059709855995/
- https://medium.com/@cheng3374/%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B8%AA%E0%B8%B4%E0%B8%97%E0%B8%98%E0%B8%B4%E0%B8%A0%E0%B8%B2%E0%B8%9E-model-%E0%B8%88%E0%B8%B2%E0%B8%81-confusion-matrix-69d391bcd48
- https://datarockie.com/blog/euclidean-distance-kmeans-knn/

ภาคผนวก

ภาคผนวก ก
ข้อมูลโครงงาน

[1] ข้อมูลที่ใช้

https://drive.google.com/file/d/1AW7UKr9mvj9Eq6xbU-HXuutgU2cKgMhj/view?usp=sharing

[2] Source code หรือ File ที่ใช้ในการคำนวณ

https://colab.research.google.com/drive/1fnsR-N_Sqk3YKslCqW9s9z4776X8kMx0?usp=sharing

https://github.com/Suratan63011017/LinearAlgebraProject

https://docs.google.com/spreadsheets/d/1rYlRF_3z-0rMc18n6NPBGoaezdy8yvTJ/edit?usp=sharing&ouid=104765072424806059580&rtpof=true&sd=true

ภาคผนวก ข

วิดีโอและสไลด์นำเสนอโครงงาน

ลิงค์เก็บวิดีโอและสไลด์

https://drive.google.com/drive/folders/18qJpMO05C7wBe2rS-ZywYf1q7e4FThpq?usp=sharing

รูปภาพผู้จัดทำโครงงาน

63010791 ยงยุทธ์ แก้วดวงน้อย

63010852 วรวิชญ์ ธรรมารักษ์วัฒนะ

63011017 สุรธันย์ บุญผ่อง

63011018 สุรพัศ วงศ์ประไพพักตร์