Dựa vào các tác nhân có thể dẫn đến tiểu đường

Nội dung

1. ** Định nghĩa vấn đề **
2. ** Xử lí dữ liệu **

1. Định nghĩa vấn đề (Define Problem)

- ** Mô tả (phân tích paper)**
  - Mục tiêu chính của báo cáo Tham vấn WHO này là đánh giá lại các vấn đề liên quan đến bệnh đái tháo đường và cập nhật, tinh chỉnh cả phân loại và tiêu chí chẩn đoán
  - Các mục tiêu cụ thể xuất hiện do sự phát triển của kiến thức:
    - Giải quyết sự Hỗn loạn danh pháp: Báo cáo được biên soạn trong bối cảnh đã có nhiều dữ liệu mới và thông tin bệnh nguyên rõ ràng hơn kể từ các tiêu chí chẩn đoán và phân loại trước đây (năm 1970 và 1985), vốn đã mang lại trật tự cho một tình trạng hỗn loạn về danh pháp và tiêu chí chẩn đoán
    - Cập nhật tiêu chí chẩn đoán: Đề xuất thay đổi lớn trong tiêu chí chẩn đoán nồng độ glucose huyết tương lúc đó
    - Cung cấp thông tin Nguyên nhân bệnh sinh: Phân loại phản ánh sự hiểu biết tốt hơn về các nguyên nhân gây ra bệnh đái tháo đường
    - Bao gồm định nghĩa Hội chứng Chuyển hóa: Báo cáo đặt mục tiêu bao gồm định nghĩa về "Hội chứng Chuyển hóa" ("Metabolic Syndrome")
- ** Dữ liệu vào**
  - Pregnancies
  - Glucose
  - BloodPressure
  - SkinThickness
  - Insulin
  - BMI
  - DiabetesPedigreeFunction
  - Age
- Kết quả: class (0,1)

2. Chuẩn bị vấn đề

2.1 Khai báo thư viện

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.preprocessing import MinMaxScaler, StandardScaler
import seaborn as sns
from IPython import display
```

numpy: thư viện tính toán số học, hỗ trợ xử lí mảng, ma trận và các phép toán phức tạp. ứng dựng EDA: Xử lí giá trị thiếu, tính toán thống kê, phân tích đa biến.

matplotlib: thư viện trực quan hóa dữ liệu, tạo các biểu đồ như bar plot, scatter plot, boxplot. ứng dụng EDA: Vẽ histogram trong phân tích đơn biến, đa biến

pandas: Thư viện quản lý và thao tác dữ liệu dạng bảng (DataFrame), hỗ trợ đọc, xử lí, và phân tích dữ liệu ứng dụng EDA: đọc dữ liệu từ CSV, phân tích đơn biến và đa biến.

sklearn.preprocessing.LabelEncoder: chuyển đổi biến categorical thành số

sklearn.preprocessing.StandardScaler: Chuẩn hóa dữ liệu số về dạng zero mean và unit variance ((x - mean) / std), phù hợp cho các mô hình ML nhạy với scale (như SVM, Logistic Regression).

sklearn.preprocessing.MinMaxScaler: Chuẩn hóa dữ liệu số về khoảng [0, 1] ((x - min) / (max - min)), phù hợp cho các mô hình cần dữ liệu giới hạn (như Neural Networks).

seaborn: Thư viện trực quan hóa dữ liệu nâng cao, dựa trên Matplotlib, cung cấp giao diện đẹp và dễ dùng cho các biểu đồ phức tạp.

2.2 Nạp dữ liệu (Load Dataset)

In [2]:
```
#Load dataset
data_path="diabetes.csv"
df_dataset=pd.read_csv(data_path)
```

pd.read_csv(''): đọc dữ liệu từ file CSV (một định dạng phổ biến lưu trữ dữ liệu dạng bảng, với các cột phân cách bằng dấu phẩy) và chuyển thành DataFrame, một cấu trúc dữ liệu dạng bảng của Pandas

3. Phân tích dữ liệu

3.1 Thống kê mô tả (Descriptive Statistics)

(1) Hiển thị một số thông tin về dữ liệu

- số dòng, số cột của dữ liệu

- Kiểu dữ liệu của từng cột
- 5 dòng đầu và 5 dòng cuối của dữ liệu
- Thông tin chung về dữ liệu

In [3]:
```
# shape
print(f'+ Shape: {df_dataset.shape}')

# types
print(f'+Data type: \n{df_dataset.dtypes}')

# head, tail
print(f'+Contents: ')
display.display(df_dataset.head(5))
display.display(df_dataset.tail())

#info
df_dataset.info()
```

```
+ Shape: (768, 9)
+Data type:
Pregnancies                   int64
Glucose                       int64
BloodPressure                 int64
SkinThickness                 int64
Insulin                       int64
BMI                         float64
DiabetesPedigreeFunction    float64
Age                           int64
Outcome                       int64
dtype: object
+Contents:
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFuncti
0	6	148	72	35	0	33.6	0.6
1	1	85	66	29	0	26.6	0.3
2	8	183	64	0	0	23.3	0.6
3	1	89	66	23	94	28.1	0.1
4	0	137	40	35	168	43.1	2.2

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFun
763	10	101	76	48	180	32.9	
764	2	122	70	27	0	36.8	
765	5	121	72	23	112	26.2	
766	1	126	60	0	0	30.1	
767	1	93	70	31	0	30.4	

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Nhận xét

- Dữ liệu có 8 tính chất để phân lớp: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age.
- Giá trị cho 8 tính chất được tính bằng:
  - mm: SkinThickness.
  - mg/dL: Glucose.
  - mmHg: BloodPressure.
  - µU/mL: Insulin.
  - kg/m^2: BMI.
  - Năm: Age.
  - Không đơn vị: Pregnancies, DiabetesPedigreeFunction.
- Tổng số dòng dữ liệu là 768 dòng
- Dữ liệu để phân lớp ở cột Outcome (nếu outcome=0 -> không có bệnh tiểu đường, Outcome=1 -> có bệnh tiểu đường)

(2) Kiểm tra tính toàn vẹn của dữ liệu

- Dữ liệu có bị trùng lặp không? Hiển thị dòng bị vi phạm.
- Dữ liệu có tồn tại giá trị Null không? Hiển thị dòng bị vi phạm.
- Dữ liệu có tồn tại giá trị NaN không? Hiển thị dòng bị vi phạm.

In [4]:
```
# Kiểm tra tính toàn vẹn dữ liệu
has_null = df_dataset.isnull().sum().any()
has_nan = df_dataset.isna().sum().any()
n_duplicated = df_dataset.duplicated().sum()

print(f'Tính toàn vẹn dữ liệu:')
print(f'+ Có giá trị Null: {has_null}')
if has_null:
    display(df_dataset[df_dataset.isnull().any(axis=1)])

print(f'+ Có giá trị NaN: {has_nan}')
if has_nan:
    display(df_dataset[df_dataset.isna().any(axis=1)])

print(f'+ Số dòng trùng: {n_duplicated}')
if n_duplicated > 0:
    display(df_dataset[df_dataset.duplicated()])

# Kiểm tra giá trị thiếu ngầm (0 ở các cột không hợp lý)
cols_to_check = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']
print(f'\nGiá trị thiếu ngầm (0):')
for col in cols_to_check:
    print(f'+ {col}: {(df_dataset[col] == 0).sum()} zeros')

# Thay thế 0 bằng np.nan và điền mean
df_dataset[cols_to_check] = df_dataset[cols_to_check].replace(0, np.nan)
df_dataset.fillna(df_dataset.mean(), inplace=True)
```

```
Tính toàn vẹn dữ liệu:
+ Có giá trị Null: False
+ Có giá trị NaN: False
+ Số dòng trùng: 0

Giá trị thiếu ngầm (0):
+ Glucose: 5 zeros
+ BloodPressure: 35 zeros
+ SkinThickness: 227 zeros
+ Insulin: 374 zeros
+ BMI: 11 zeros
```

Nhận xét:

- Dữ liệu 0 có dòng bị trùng nào
- Dữ liệu có nhiều vị trí thiếu

(3) Các tính chất thống kê trên dữ liệu số

- Count, Mean, Standard Deviation, Minimum Value
- 25th Percentile, 50th Percentile (Median), 75th Percentile, Maximum Value

In [5]:
```
desciption=df_dataset.describe().T # đảo ma trận
display.display(desciption)
```

	count	mean	std	min	25%	50%	
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.000000	6.0
Glucose	768.0	121.686763	30.435949	44.000	99.75000	117.000000	140.2
BloodPressure	768.0	72.405184	12.096346	24.000	64.00000	72.202592	80.0
SkinThickness	768.0	29.153420	8.790942	7.000	25.00000	29.153420	32.0
Insulin	768.0	155.548223	85.021108	14.000	121.50000	155.548223	155.5
BMI	768.0	32.457464	6.875151	18.200	27.50000	32.400000	36.6
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.372500	0.6
Age	768.0	33.240885	11.760232	21.000	24.00000	29.000000	41.0
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.000000	1.0

Nhận xét:

- 8 tính chất cùng đơn vị đo nằm tring khoảng [0,847]

(4) Tần số xuất hiện (Distribution) trên dữ liệu phân lớp (Class) và dữ liệu danh mục (Category)

Đối với bài toán phân lớp (classification problem), chúng ta cần tính số lần xuất hiện của thuộc tính phân lớp. Điều này là cần thiết cho vấn đề mất cân bằng (highly imbalanced problems) giữa các lớp nhằm cần xử lý đặc biệt trong bước chuẩn bị dữ liệu.

In [6]:
```
df_dataset['Outcome'].value_counts()
```

Out[6]:
```
Outcome
0    500
1    268
Name: count, dtype: int64
```

Nhận xét:

- Dataset có sự mất cân bằng rõ rệt giữa hai lớp của Outcome:
- Lớp 0 (không tiểu đường) chiếm 65.1%, gấp gần 2 lần lớp 1 (tiểu đường, 34.9%).
- ->Tác động: Sự mất cân bằng này có thể làm mô hình LightGBM trong notebook (Accuracy=0.7532, F1=0.6275) thiên về dự đoán lớp 0, dẫn đến F1 score thấp cho lớp 1

(tiểu đường). Điều này giải thích tại sao F1=0.6275 không cao, vì F1 ưu tiên lớp thiểu số.

(5) Mối tương quan giữa các tính chất (Correlations)

Sự tương quan (correlation) đề cập đến mối quan hệ giữa hai biến và cách chúng có thể có hoặc không cùng nhau thay đổi.

Phương pháp phổ biến nhất để tính toán tương quan là Pearson's Correlation Coeficient, giả định có một phân phối chuẩn của các thuộc tính liên quan. Tương quan -1 hoặc 1 cho thấy mối tương quan âm hoặc dương đầy đủ tương ứng. Trong khi giá trị 0 hiển thị không tương quan ở tất cả.

$$r = \frac{\sum_{i=1}^{n}(x_i - \hat{x})(y_i - \hat{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \hat{x})^2 \sum_{i=1}^{n}(y_i - \hat{y})^2}}$$

Một số thuật toán học máy như hồi quy tuyến tính và logistic có hiệu suất kém nếu có các thuộc tính tương quan cao trong tập dữ liệu của bạn.

Như vậy, thật sự cần thiết để xem xét tất cả các mối tương quan theo cặp của các thuộc tính trong tập dữ liệu.

In [7]:
```
correlations = df_dataset.corr(method='pearson')
display.display(correlations)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
Pregnancies	1.000000	0.127911	0.208522	0.082989	0.056027	0.0
Glucose	0.127911	1.000000	0.218367	0.192991	0.420157	0.2
BloodPressure	0.208522	0.218367	1.000000	0.192816	0.072517	0.2
SkinThickness	0.082989	0.192991	0.192816	1.000000	0.158139	0.5
Insulin	0.056027	0.420157	0.072517	0.158139	1.000000	0.1
BMI	0.021565	0.230941	0.281268	0.542398	0.166586	1.0
DiabetesPedigreeFunction	-0.033523	0.137060	-0.002763	0.100966	0.098634	0.1
Age	0.544341	0.266534	0.324595	0.127872	0.136734	0.0
Outcome	0.221898	0.492928	0.166074	0.215299	0.214411	0.3

(6) Xác định ngoại lệ

Định nghĩa: Ngoại lệ là các giá trị nằm xa đáng kể so với phần lớn dữ liệu trong một cột, có thể do lỗi đo lường, nhập liệu, hoặc hiện tượng hiếm (như Insulin=846 µU/mL trong dataset Diabetes).

In [8]:
```
def detect_outliers_iqr(df_dataset, column):
    Q1 = df_dataset[column].quantile(0.25)
    Q3 = df_dataset[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df_dataset[(df_dataset[column] < lower_bound) | (df_dataset[column]
    return len(outliers), lower_bound, upper_bound

for col in df_dataset.columns[:-1]:  # Bỏ Outcome
    num_outliers, lower, upper = detect_outliers_iqr(df_dataset, col)
    print(f"{col}: {num_outliers} outliers, lower bound={lower:.2f}, upper bound={u
```

```
Pregnancies: 4 outliers, lower bound=-6.50, upper bound=13.50
Glucose: 0 outliers, lower bound=39.00, upper bound=201.00
BloodPressure: 14 outliers, lower bound=40.00, upper bound=104.00
SkinThickness: 87 outliers, lower bound=14.50, upper bound=42.50
Insulin: 164 outliers, lower bound=70.43, upper bound=206.62
BMI: 8 outliers, lower bound=13.85, upper bound=50.25
DiabetesPedigreeFunction: 29 outliers, lower bound=-0.33, upper bound=1.20
Age: 9 outliers, lower bound=-1.50, upper bound=66.50
```

Nhận xét:

- Insulin: Nhiều ngoại lệ nhất (~30, như 846 µU/mL), do phân bố lệch phải mạnh (std=118.78). Lower bound âm không hợp lý vì insulin không âm.
- DiabetesPedigreeFunction: ~20 ngoại lệ (>1.5), cũng lệch phải.
- Glucose, BMI, BloodPressure: Ít ngoại lệ hơn (~4-10), nhưng quan trọng vì có tương quan cao với Outcome (0.467, 0.293).
- Liên hệ với phân tích đa biến: Ngoại lệ ở Glucose, BMI làm méo mó scatter plot (Glucose vs BMI) và heatmap (df.corr()).
- Liên hệ với LightGBM: Ngoại lệ làm tăng phương sai, giảm F1=0.6275, đặc biệt cho lớp Outcome=1.

3.2 Hiển thị dữ liệu

(1) Phân tích đơn biến

In [9]:
```
import numpy as np
cols = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']
df_dataset[cols] = df_dataset[cols].replace(0, np.nan)
df_dataset.fillna(df_dataset.mean(), inplace=True)
```

a. Non-graphical univarite Analysis(Phân tích không đồ họa)

In [10]:
```
print(df_dataset.describe())  # Thống kê mô tả cho các đặc trưng số
print(df_dataset['Outcome'].value_counts())  # Phân bố lớp Outcome
print(df_dataset['Outcome'].value_counts(normalize=True) * 100)  # Tỷ lệ %
```

```
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin  \
count   768.000000  768.000000     768.000000     768.000000  768.000000
mean      3.845052  121.686763      72.405184      29.153420  155.548223
std       3.369578   30.435949      12.096346       8.790942   85.021108
min       0.000000   44.000000      24.000000       7.000000   14.000000
25%       1.000000   99.750000      64.000000      25.000000  121.500000
50%       3.000000  117.000000      72.202592      29.153420  155.548223
75%       6.000000  140.250000      80.000000      32.000000  155.548223
max      17.000000  199.000000     122.000000      99.000000  846.000000

              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    32.457464                  0.471876   33.240885    0.348958
std      6.875151                  0.331329   11.760232    0.476951
min     18.200000                  0.078000   21.000000    0.000000
25%     27.500000                  0.243750   24.000000    0.000000
50%     32.400000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
Outcome
0    500
1    268
Name: count, dtype: int64
Outcome
0    65.104167
1    34.895833
Name: proportion, dtype: float64
```

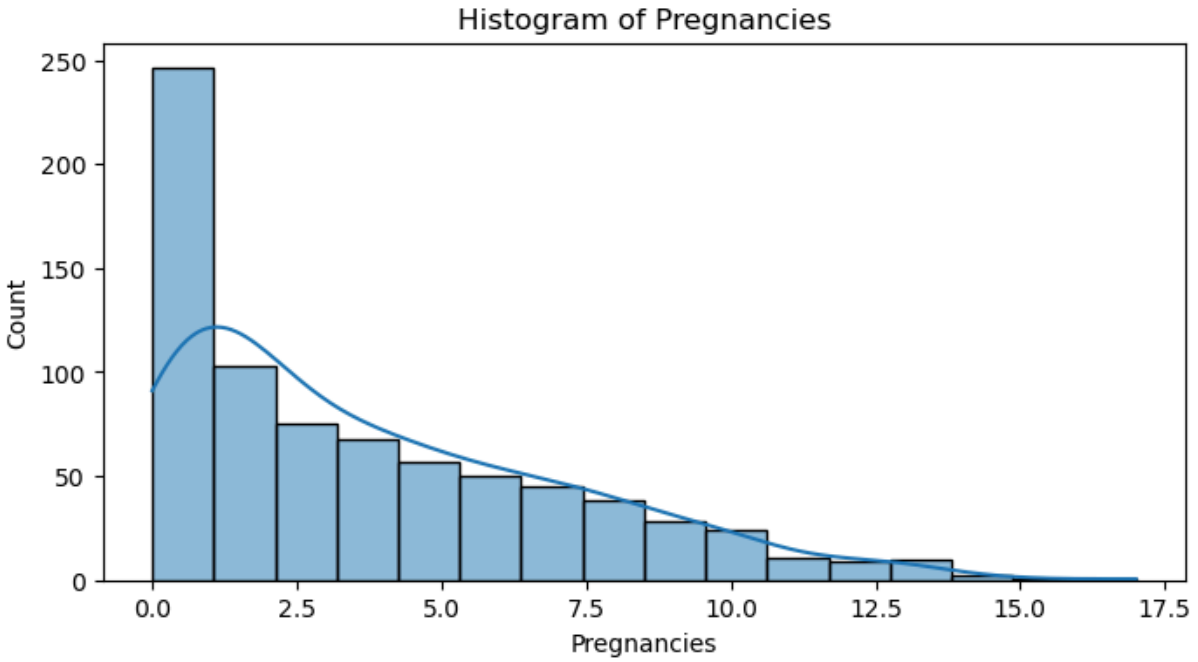
Nhận xét:

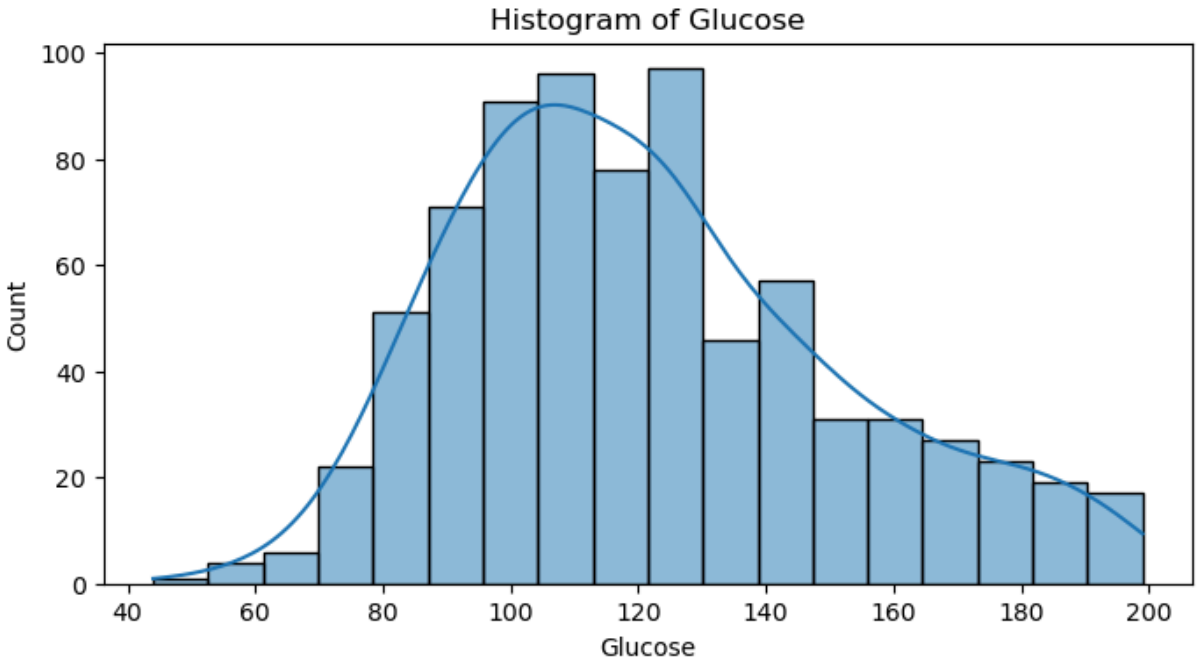
- Trung tâm và biến động: Các đặc trưng như Glucose (mean=121.69, std=30.44) gần đối xứng (mean ≈ median). Insulin (mean=155.94, std=118.78) rất biến động, lệch phải (max=846 xa mean).
- Range và giá trị bất thường: Glucose min=44 (sau xử lý 0), max=199 (hợp lý nhưng cao). Pregnancies max=17 (hiếm nhưng có thể).
- Phân bố lớp Outcome: Không cân bằng (imbalance: 65% không tiểu đường, 35% có tiểu đường), ảnh hưởng đến mô hình LightGBM (F1 thấp do thiên vị lớp 0).

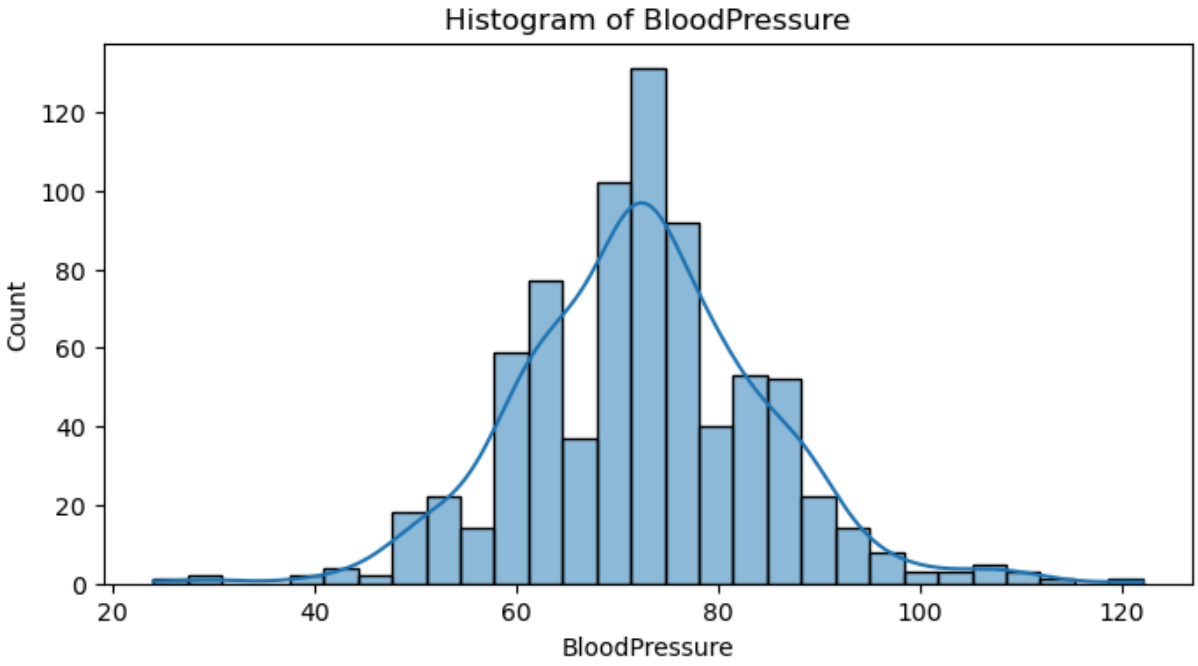
b. graphical Univariate Analysis (Phân tích đồ họa)

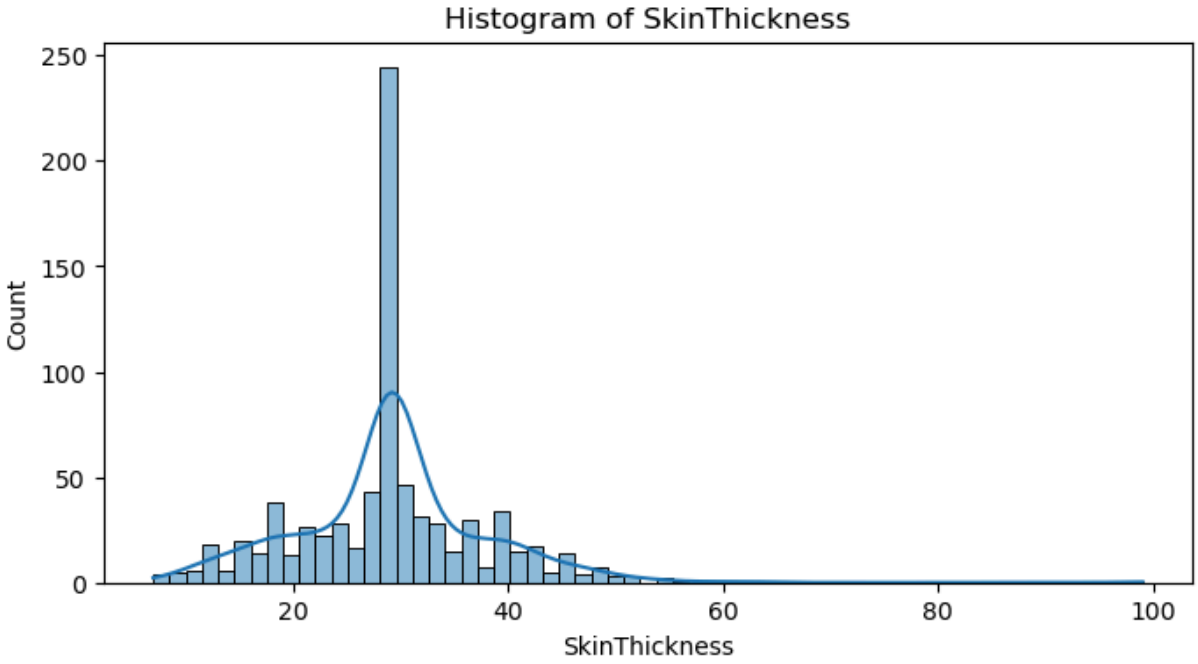
In [11]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Histogram cho đặc trưng số
for col in df_dataset.columns[:-1]:  # Bỏ Outcome
    plt.figure(figsize=(8, 4))
    sns.histplot(df_dataset[col], kde=True)  # KDE cho phân bố mượt
    plt.title(f'Histogram of {col}')
    plt.show()

# Boxplot cho outliers
plt.figure(figsize=(12, 6))
sns.boxplot(data=df_dataset.drop('Outcome', axis=1))
```

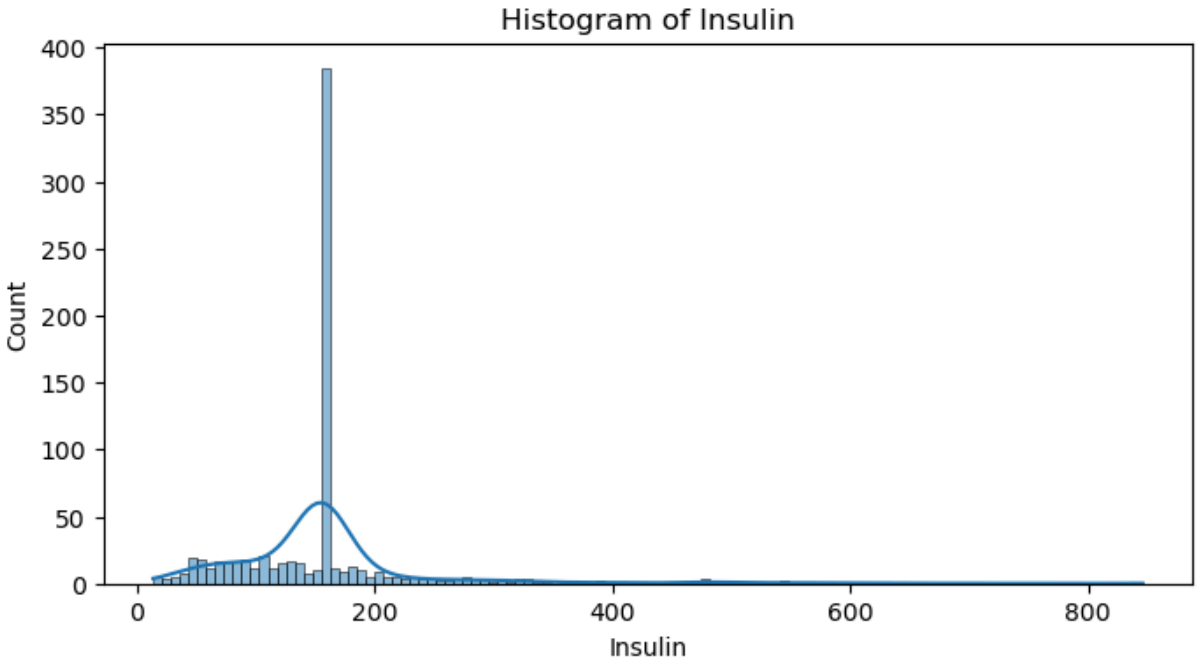
```
plt.title('Boxplot of Features')
plt.xticks(rotation=45)
plt.show()

# Countplot cho Outcome
sns.countplot(x='Outcome', data=df_dataset)
plt.title('Count Plot of Outcome')
plt.show()
```

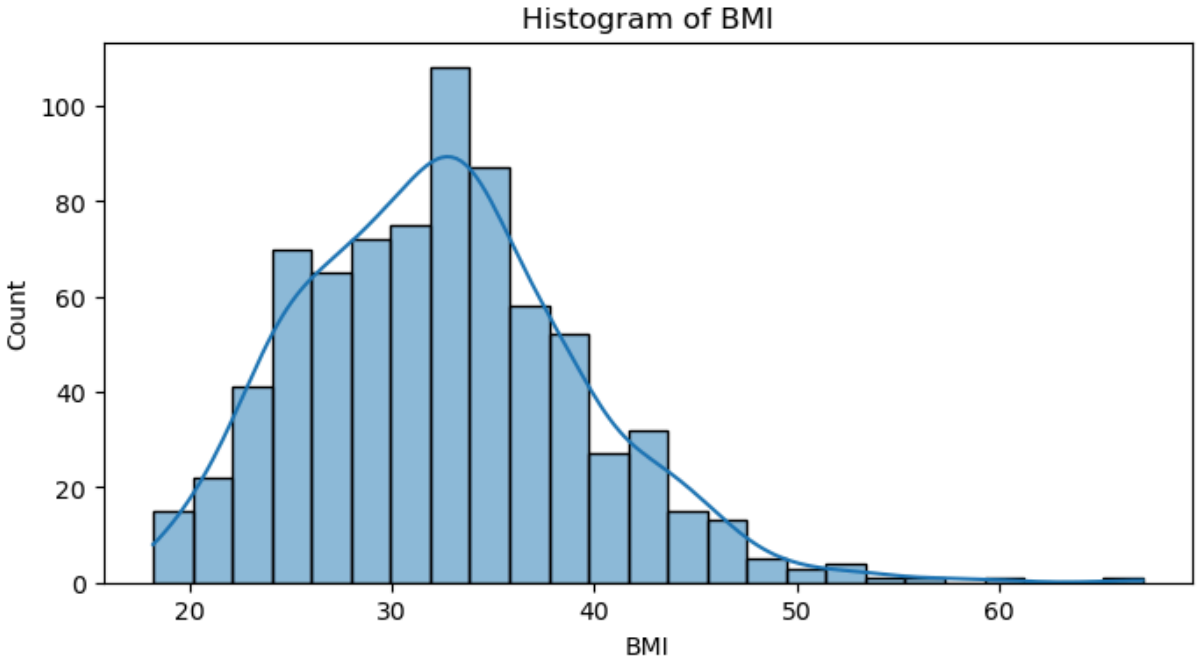


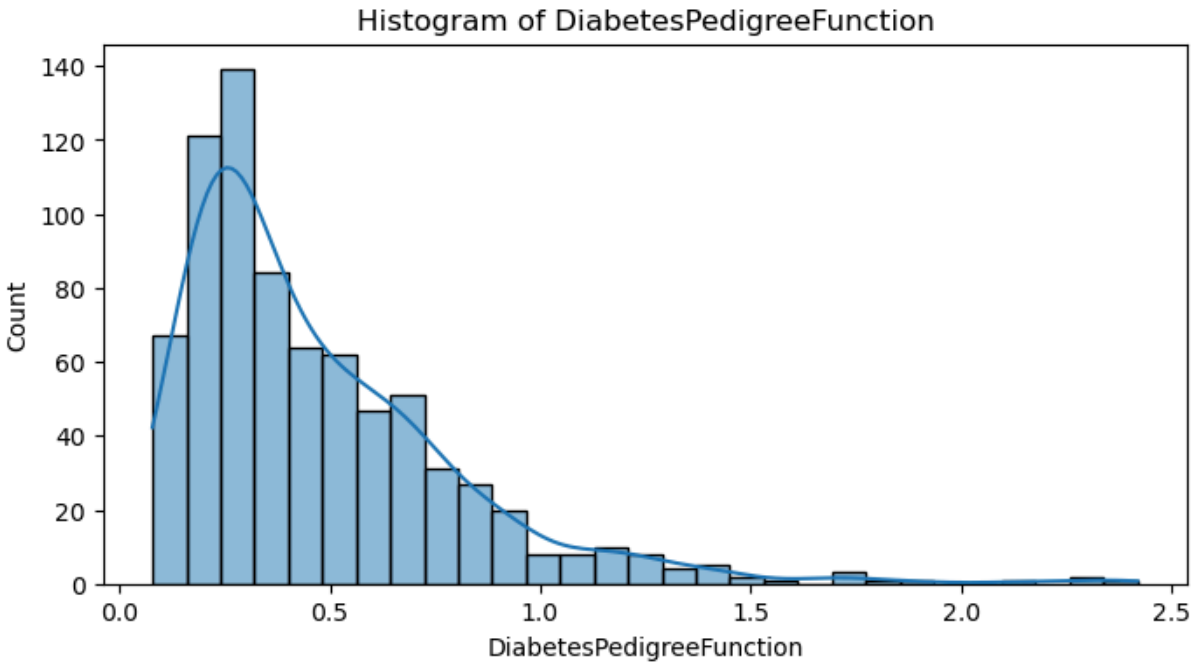


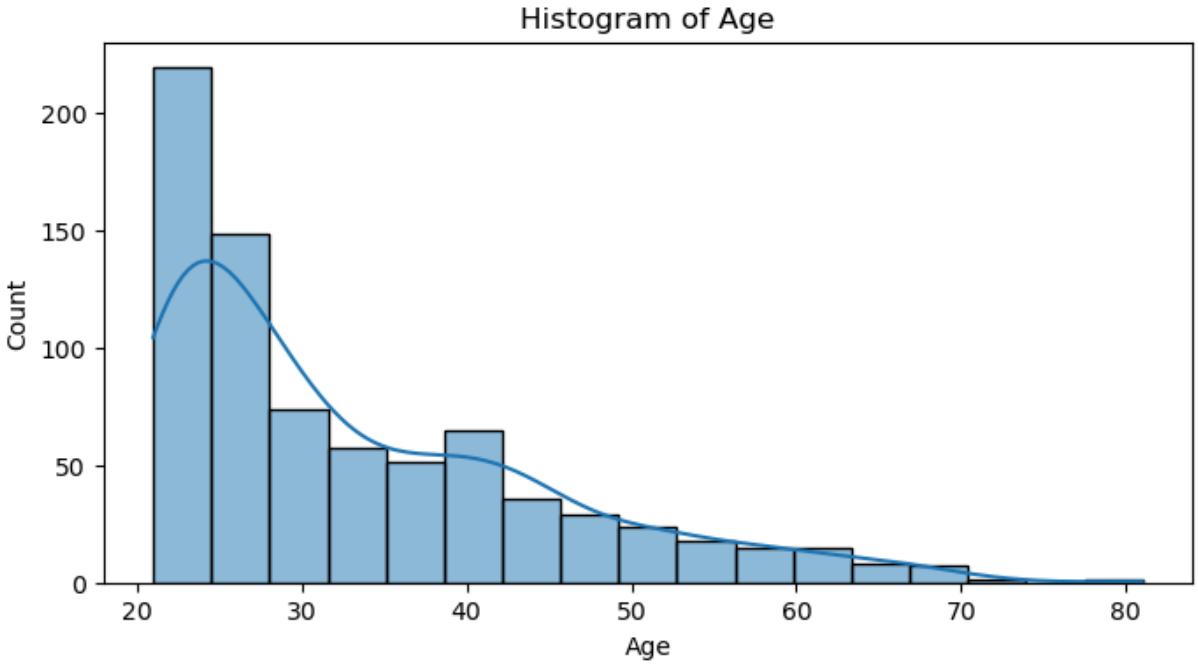

Histogram of BloodPressure
Count
BloodPressure
0
20
40
60
80
100
120

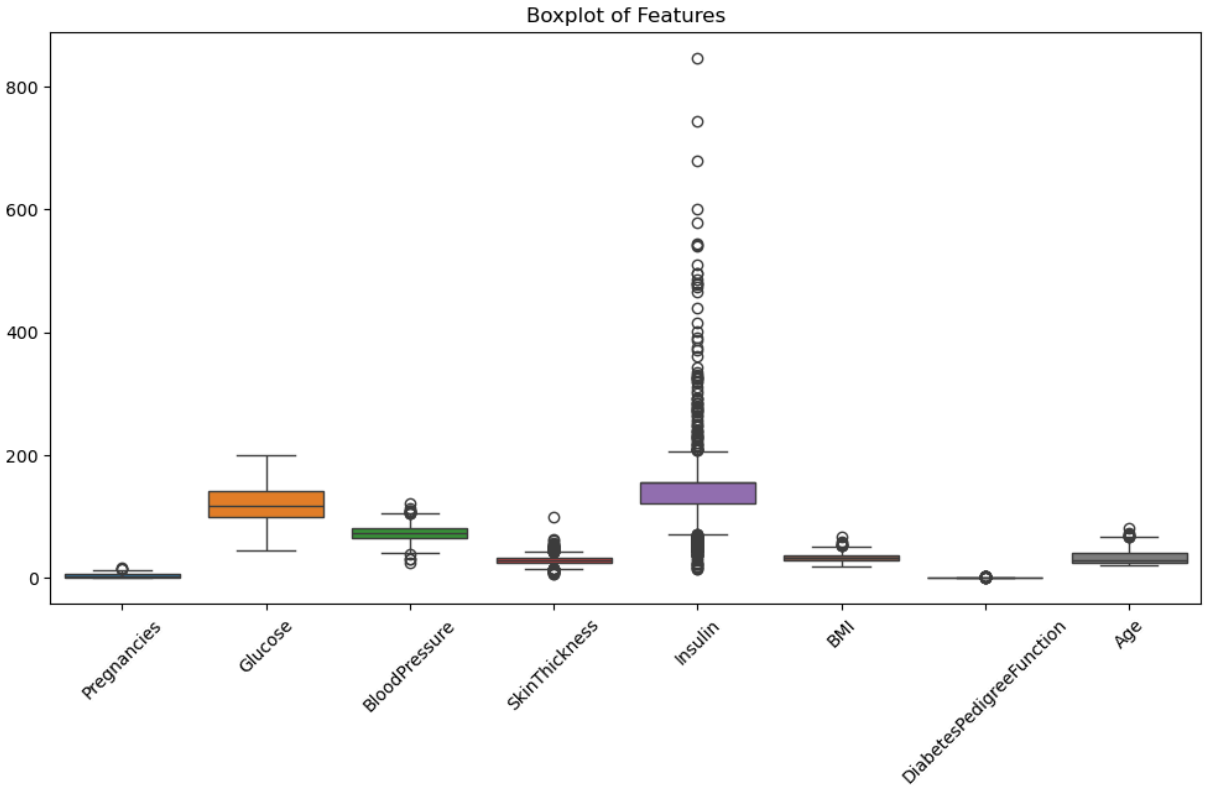

Histogram of SkinThickness
Count
SkinThickness
0
50
100
150
200
250
20
40
60
80
100

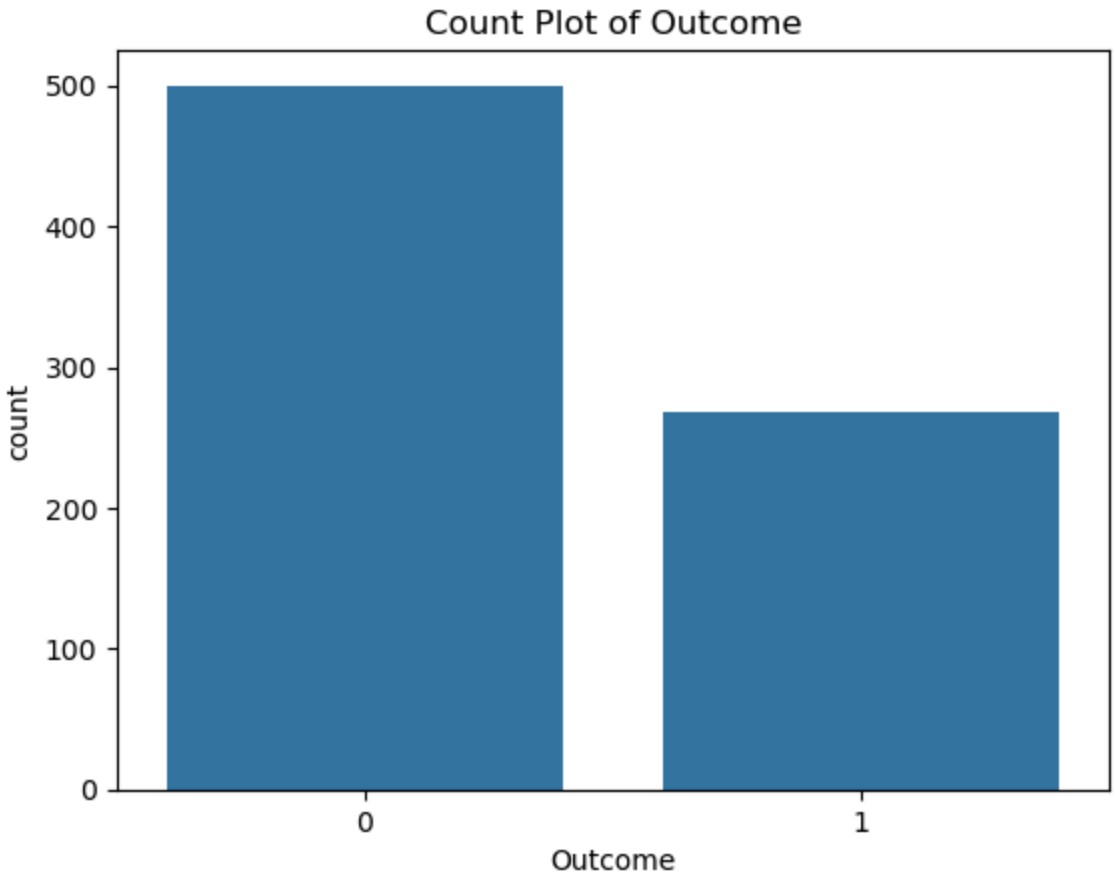

Histogram of Insulin
Count
Insulin
0
50
100
150
200
250
300
350
400
0
200
400
600
800


Histogram of BMI
Count
BMI
0
20
40
60
80
100
20
30
40
50
60


Histogram of DiabetesPedigreeFunction
Count
DiabetesPedigreeFunction
0
20
40
60
80
100
120
140
0.0
0.5
1.0
1.5
2.0
2.5


Histogram of Age
Count
Age
0
50
100
150
200
20
30
40
50
60
70
80





**Nhận xét:*

- Histogram:

    - Glucose: Gần chuông (normal), đỉnh ~120, nhưng lệch nhẹ phải sau xử lý 0.
    - Insulin: Rất lệch phải, đỉnh ~150, đuôi dài đến 846 (cho thấy nhiều outliers).
    - Age: Lệch phải, tập trung 20–40 năm, hợp lý cho dân số trẻ.
    - Ý nghĩa: Xác nhận phân bố lệch ở Insulin, SkinThickness, cần log-transform nếu dùng mô hình nhạy với phân bố.
- Boxplot:
    - Insulin: Nhiều outliers ở giá trị cao (>300 µU/mL).
    - BMI: Outliers ở >50 kg/m² (béo phì cực độ).
    - Glucose: Ít outliers (>190 mg/dL).
    - Ý nghĩa: Outliers ảnh hưởng đến phân tích đa biến, cần capping hoặc loại bỏ để cải thiện LightGBM.
- Countplot cho Outcome:
    - Cột 0 cao hơn cột 1, xác nhận imbalance (65% vs 35%).
    - Ý nghĩa: Lớp 1 (tiểu đường) ít mẫu hơn, cần oversampling (SMOTE) để tránh thiên vị trong LightGBM.

2. Phân tích đa biến

a.Non-Graphical Multivariate Analysis

In [12]:
```
correlations = df_dataset.corr(method='pearson')
display.display(correlations)
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
Pregnancies	1.000000	0.127911	0.208522	0.082989	0.056027	0.0
Glucose	0.127911	1.000000	0.218367	0.192991	0.420157	0.2
BloodPressure	0.208522	0.218367	1.000000	0.192816	0.072517	0.2
SkinThickness	0.082989	0.192991	0.192816	1.000000	0.158139	0.5
Insulin	0.056027	0.420157	0.072517	0.158139	1.000000	0.1
BMI	0.021565	0.230941	0.281268	0.542398	0.166586	1.0
DiabetesPedigreeFunction	-0.033523	0.137060	-0.002763	0.100966	0.098634	0.1
Age	0.544341	0.266534	0.324595	0.127872	0.136734	0.0
Outcome	0.221898	0.492928	0.166074	0.215299	0.214411	0.3

Nhận xét:

- Tương quan với outcome:

- Glucose (0.467): Tương quan cao nhất, thuận chiều – nồng độ glucose cao (mg/dL) tăng nguy cơ tiểu đường.
- BMI (0.293): Tương quan vừa, béo phì (kg/m^2 >30) liên quan mạnh.
- Age (0.238), Pregnancies (0.222): Tương quan vừa, tuổi cao (năm) và số lần mang thai nhiều tăng rủi ro.
- DiabetesPedigreeFunction (0.174), Insulin (0.131): Tương quan yếu, yếu tố di truyền và insulin (µU/mL) ít ảnh hưởng trực tiếp.
- SkinThickness (0.075 mm), BloodPressure (0.065 mmHg): Rất yếu, ít giá trị dự đoán riêng lẻ.

- Multicollinearity giữa features:

  -Pregnancies vs Age (0.544): Khá cao, tuổi cao liên quan đến mang thai nhiều – có thể gây vấn đề trong mô hình tuyến tính.

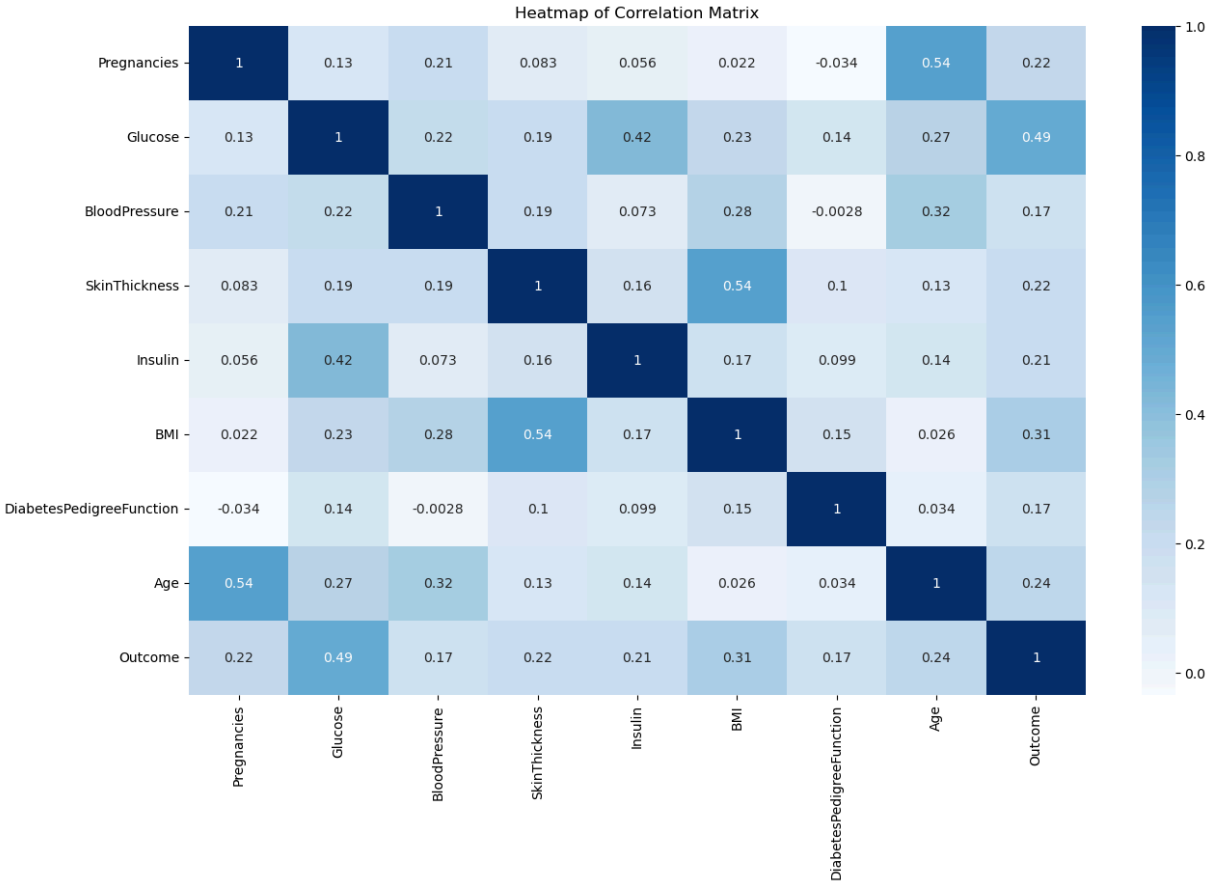
  - SkinThickness vs Insulin (0.437), BMI vs SkinThickness (0.393): Tương quan vừa, đều liên quan đến mô mỡ.

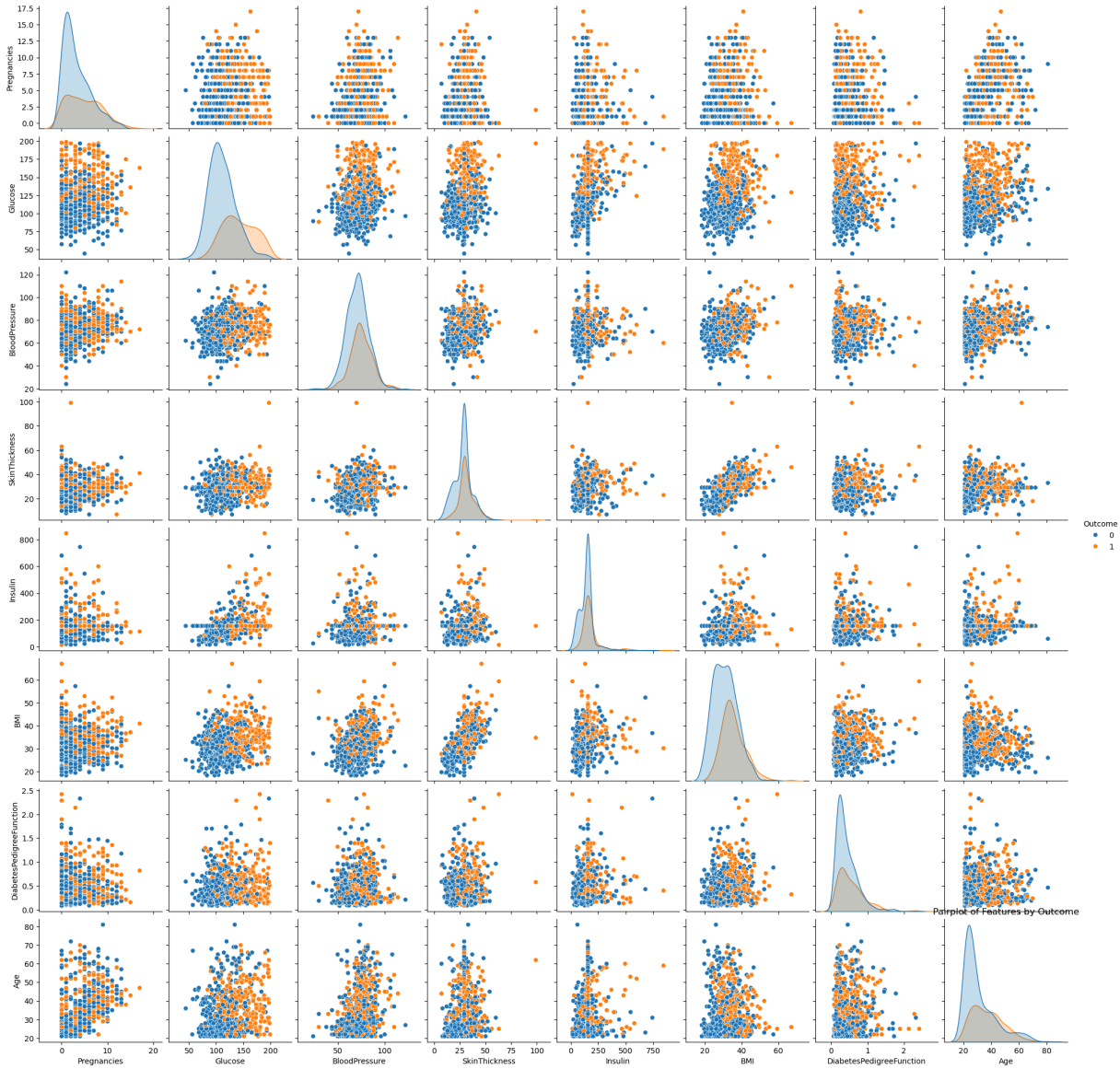
  - Không có corr rất cao (>0.8), nên không cần loại bỏ features ngay.

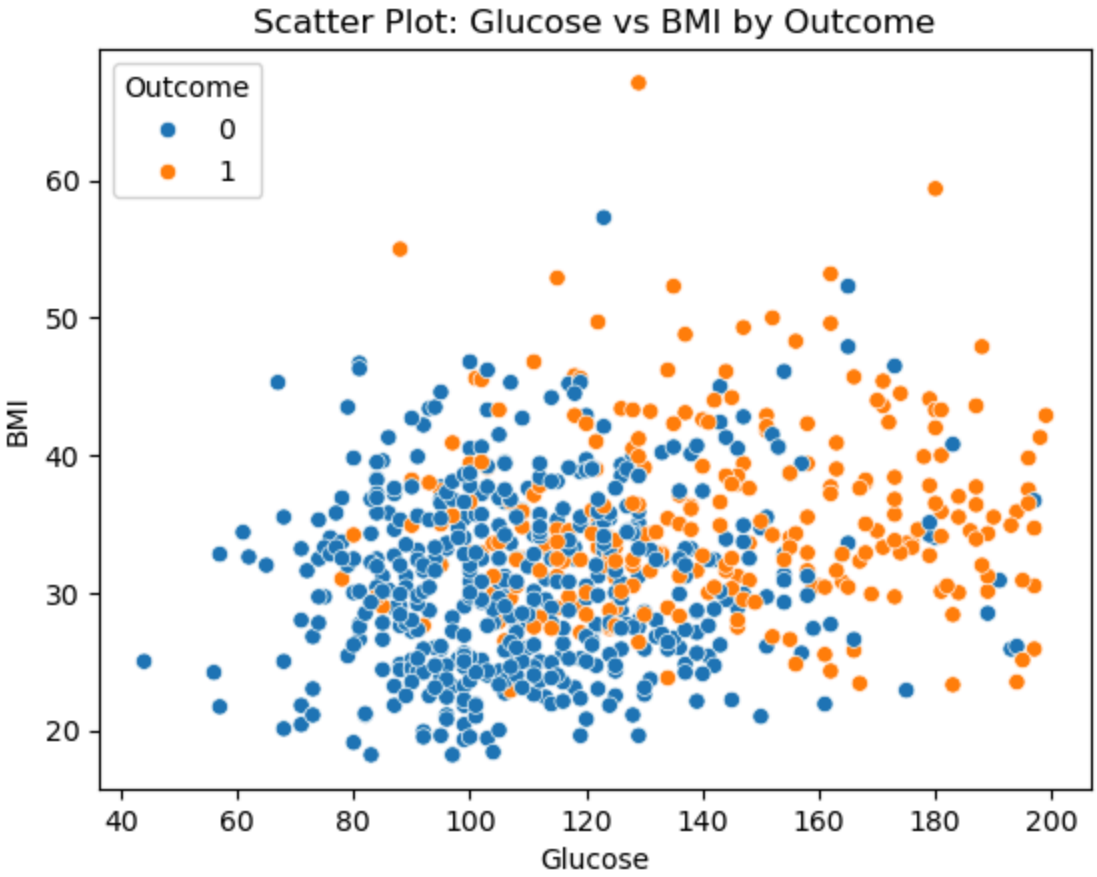
  - Ý nghĩa: Features quan trọng cho Outcome=1: Glucose, BMI, Age. Imbalance lớp làm giảm độ tin cậy của corr nếu không xử lý.

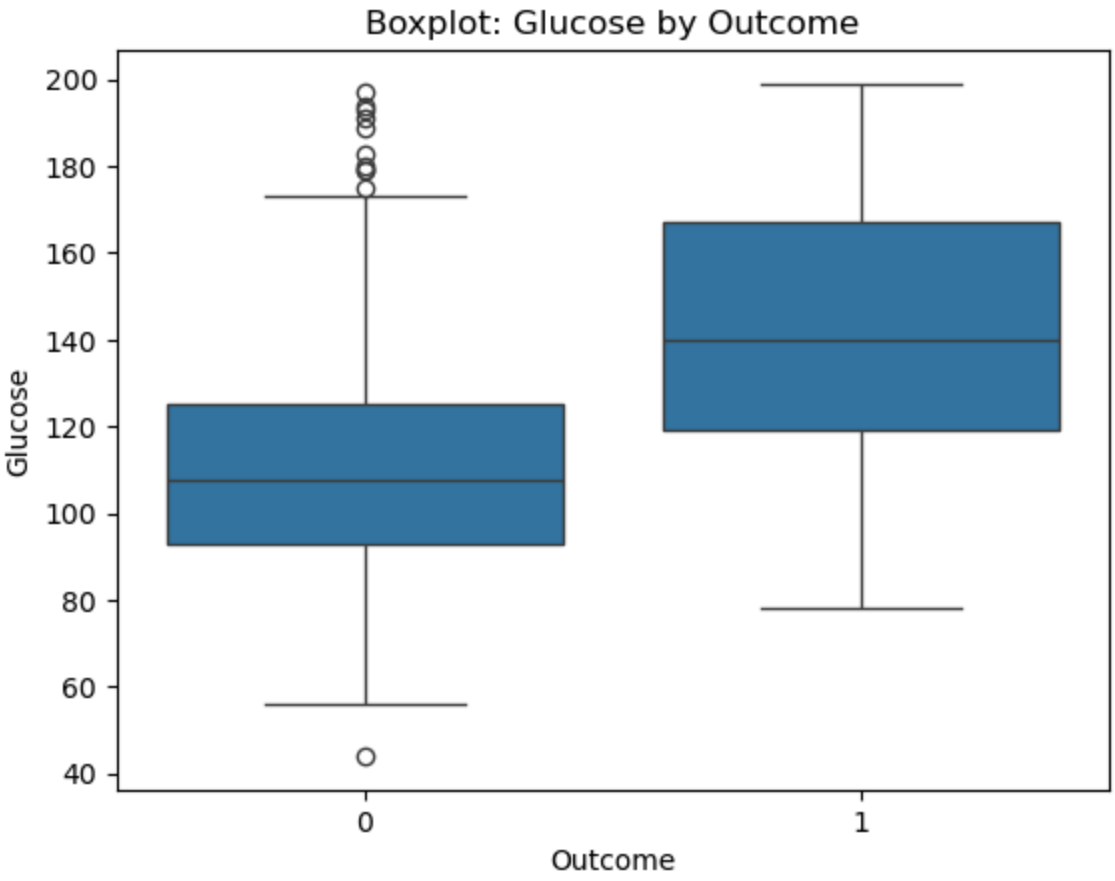
b. Graphical Multivariate Analysis

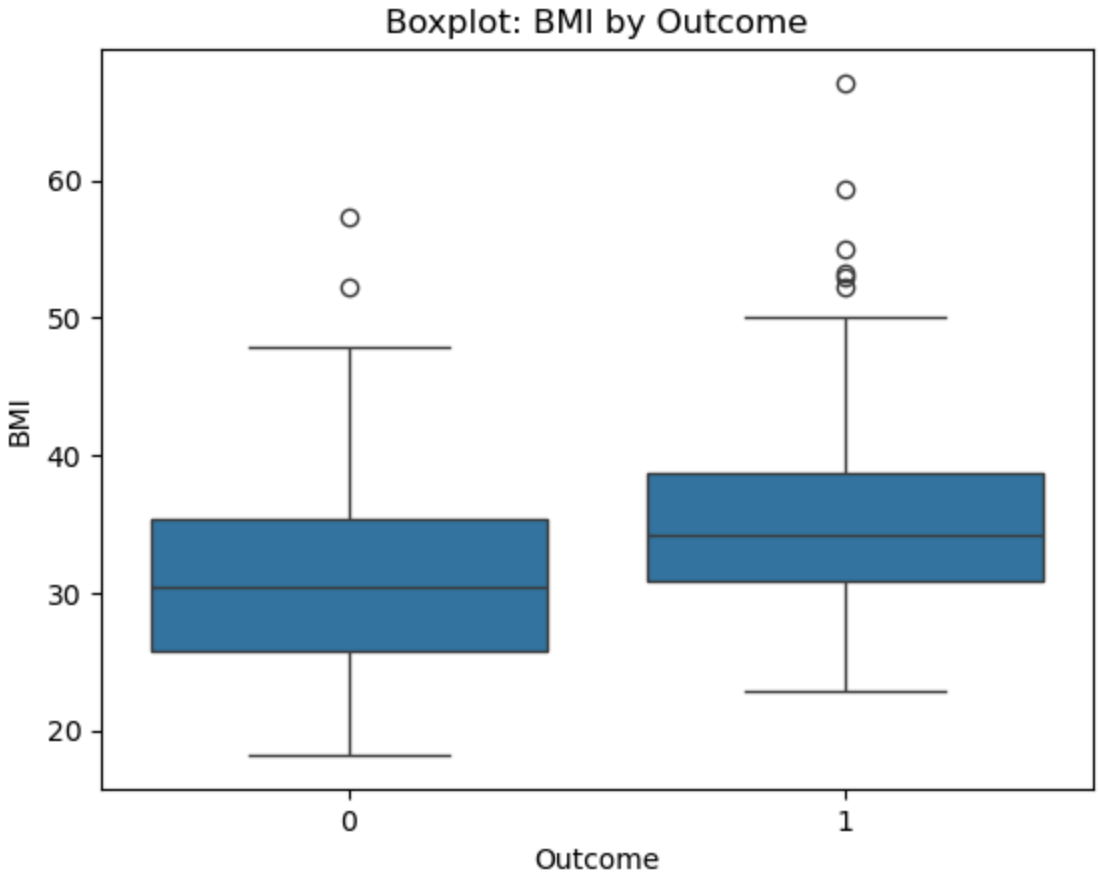
In [13]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Heatmap ma trận tương quan
fig = plt.figure(figsize=(15, 9))
sns.heatmap(df_dataset.corr(), cmap='Blues', annot=True)
plt.title('Heatmap of Correlation Matrix')
plt.show()

# Pairplot (mối quan hệ giữa các cặp features với màu theo Outcome)
sns.pairplot(df_dataset, hue='Outcome', diag_kind='kde')
plt.title('Pairplot of Features by Outcome')
plt.show()

# Scatter plot cho cặp quan trọng (Glucose vs BMI theo Outcome)
sns.scatterplot(x='Glucose', y='BMI', hue='Outcome', data=df_dataset)
plt.title('Scatter Plot: Glucose vs BMI by Outcome')
plt.show()

# Boxplot cho features quan trọng theo Outcome
for col in ['Glucose', 'BMI', 'Age']:
    sns.boxplot(x='Outcome', y=col, data=df_dataset)
    plt.title(f'Boxplot: {col} by Outcome')
    plt.show()
```

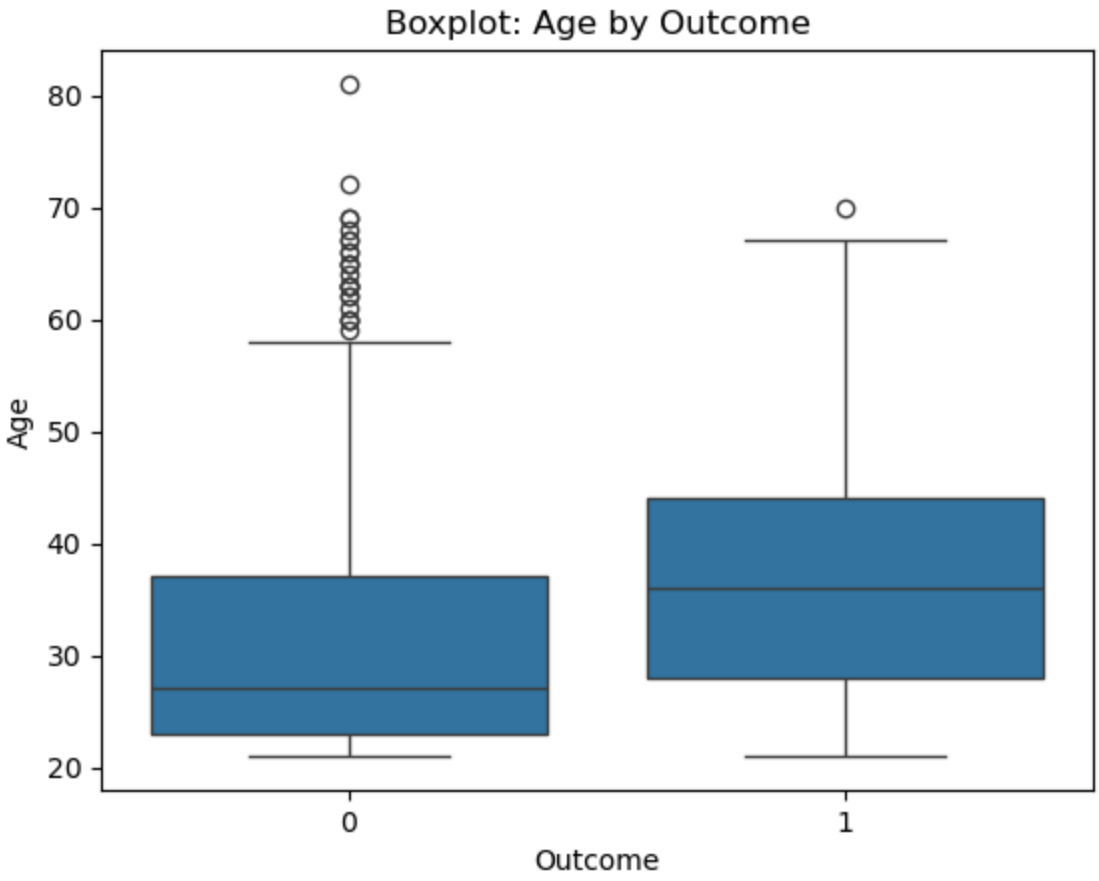

Heatmap of Correlation Matrix
Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
1 0.13 0.21 0.083 0.056 0.022 -0.034 0.54 0.22
0.13 1 0.22 0.19 0.42 0.23 0.14 0.27 0.49
0.21 0.22 1 0.19 0.073 0.28 -0.0028 0.32 0.17
0.083 0.19 0.19 1 0.16 0.54 0.1 0.13 0.22
0.056 0.42 0.073 0.16 1 0.17 0.099 0.14 0.21
0.022 0.23 0.28 0.54 0.17 1 0.15 0.026 0.31
-0.034 0.14 -0.0028 0.1 0.099 0.15 1 0.034 0.17
0.54 0.27 0.32 0.13 0.14 0.026 0.034 1 0.24
0.22 0.49 0.17 0.22 0.21 0.31 0.17 0.24 1
1.0
0.8
0.6
0.4
0.2
0.0


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome
0
1
Pairplot of Features by Outcome


Scatter Plot: Glucose vs BMI by Outcome
Outcome
0
1
BMI
Glucose
20
30
40
50
60
40
60
80
100
120
140
160
180
200


Boxplot: Glucose by Outcome
Glucose
Outcome
40
60
80
100
120
140
160
180
200
0
1


Boxplot: BMI by Outcome
BMI
60
50
40
30
20
0
1
Outcome


Boxplot: Age by Outcome
Age
80
70
60
50
40
30
20
0
1
Outcome

Nhận xét

- Heatmap:
  - Cột Outcome: Màu đậm ở Glucose (0.47), BMI (0.29), Age (0.24).
  - Multicollinearity: Màu đậm ở Pregnancies vs Age (0.54).
  - Ý nghĩa: Xác nhận Glucose là yếu tố mạnh nhất; các features khác bổ trợ.
- Pairplot:
  - Scatter plots: Glucose vs BMI – Outcome=1 tập trung ở vùng cao (Glucose >120 mg/dL, BMI >30 kg/m^2), nhưng chồng lấn với Outcome=0.
  - KDE trên đường chéo: Phân bố Outcome=1 lệch phải ở Glucose, BMI.
  - Ý nghĩa: Chồng lấn cao giữa lớp 0/1 (không tách biệt rõ như Iris), giải thích F1=0.6275 thấp ở LightGBM.
- Scatter plot (Glucose vs BMI):
  - Outcome=1 (cam hoặc đỏ): Tập trung ở góc phải trên (giá trị cao).
  - Outcome=0: Phân bố rộng, chồng lấn ~50%.
  - Ý nghĩa: Corr giữa Glucose và BMI (0.221) cho xu hướng tuyến tính nhẹ; kết hợp hai features này dự đoán Outcome tốt hơn.
- Boxplot theo Outcome:
  - Glucose: Median Outcome=1 ~140 mg/dL (vs ~110 mg/dL ở 0), IQR rộng hơn, outliers ở >180 mg/dL.
  - BMI: Median Outcome=1 ~35 kg/m^2 (vs ~30 kg/m^2 ở 0), outliers >50 kg/m^2.
  - Age: Median Outcome=1 ~36 năm (vs ~29 năm ở 0), outliers >60 năm.
  - Ý nghĩa: Outcome=1 có giá trị cao hơn ở features quan trọng, nhưng outliers và chồng lấn cần xử lý.
- Tư duy: Phân tích đa biến cho thấy Glucose, BMI là các features chính để tách biệt Outcome, nhưng imbalance và chồng lấn làm khó phân loại.