PROJECT PROGRESS
(DIABETES)

นำเสนอโดย

นายกฤตณัฐ ขจรกุลญาณี 65102010190
นายบูรพา ยืนยง 65102010418

วิชา CP 462 INTRODUCTION TO DATA SCIENCE

PROBLEM STATEMENT – PART II.

ปัญหาที่ต้องการแก้ไขด้วยข้อมูล

การทำนายการเกิดโรคเบาหวานในผู้ป่วย โดยใช้ข้อมูลทางการแพทย์และลักษณะประชากรของผู้ป่วย เช่น อายุ , น้ำหนัก , ความดันโลหิต , ระดับน้ำตาลในเลือด ฯลฯ เพื่อสร้างแบบจำลองทางสถิติ หรือ Machine Learning ที่สามารถทำนายความเสี่ยงในการเกิดโรคเบาหวานในอนาคต

การวิเคราะห์ข้อมูลเกี่ยวกับโรคเบาหวานนี้ จะสามารถช่วยในการระบุปัจจัยเสี่ยงต่างๆ ที่มีผลต่อการเกิดโรคเพื่อส่งเสริมการดูแลรักษาสุขภาพและเพิ่มภูมิคุ้มกันให้กับตนเองและคนรอบข้าง

ข้อมูลเพิ่มเติม

> Model : Logistic Regression , GaussionDB , DecisionTree , KNN
> Type of Learning : Supervised Learning
> Task : Classification

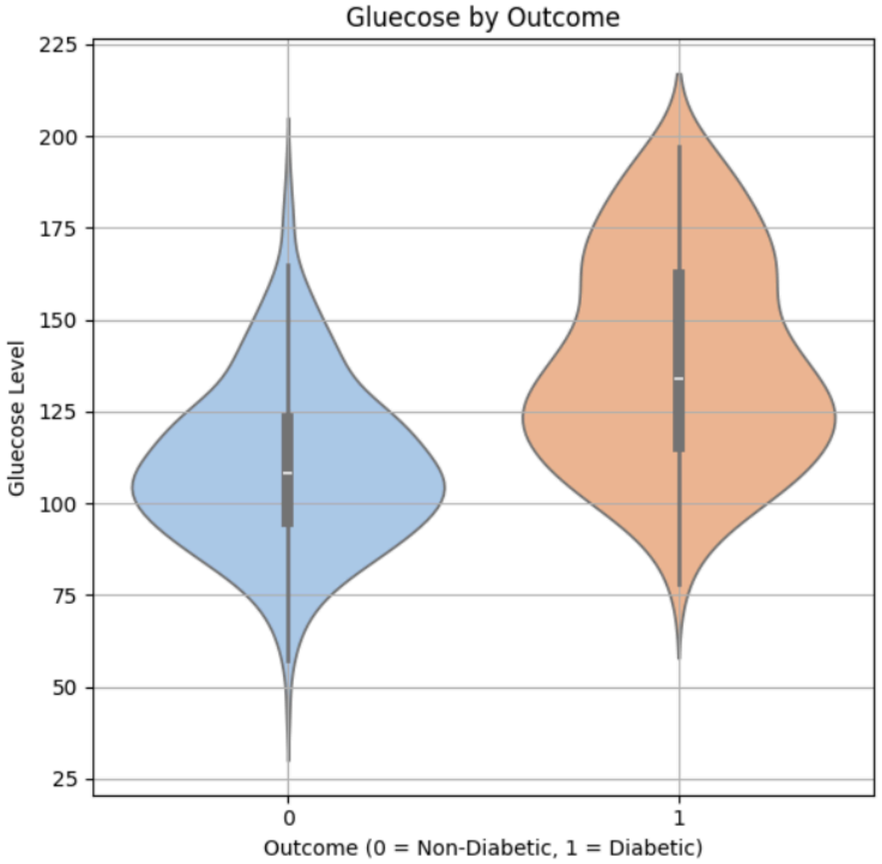
DATA ACQUISITION

```
df.info(10) # Show Column Heads
```

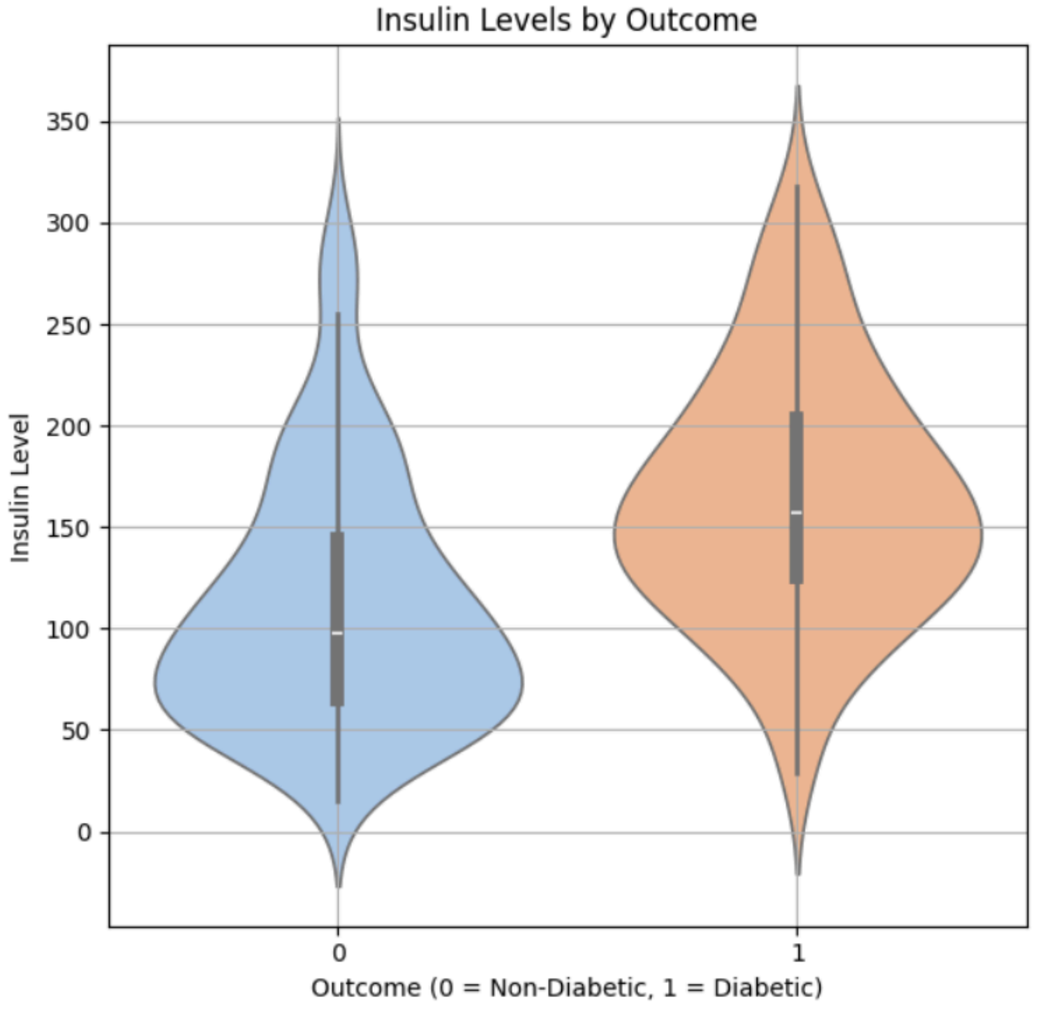
```
<class 'pandas.core.frame.DataFrame'>
Index: 332 entries, 3 to 765
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               332 non-null    int64
 1   Glucose                   332 non-null    int64
 2   BloodPressure             332 non-null    int64
 3   SkinThickness             332 non-null    int64
 4   Insulin                   332 non-null    int64
 5   BMI                       332 non-null    float64
 6   DiabetesPedigreeFunction  332 non-null    float64
 7   Age                       332 non-null    int64
 8   Outcome                   332 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 25.9 KB
```

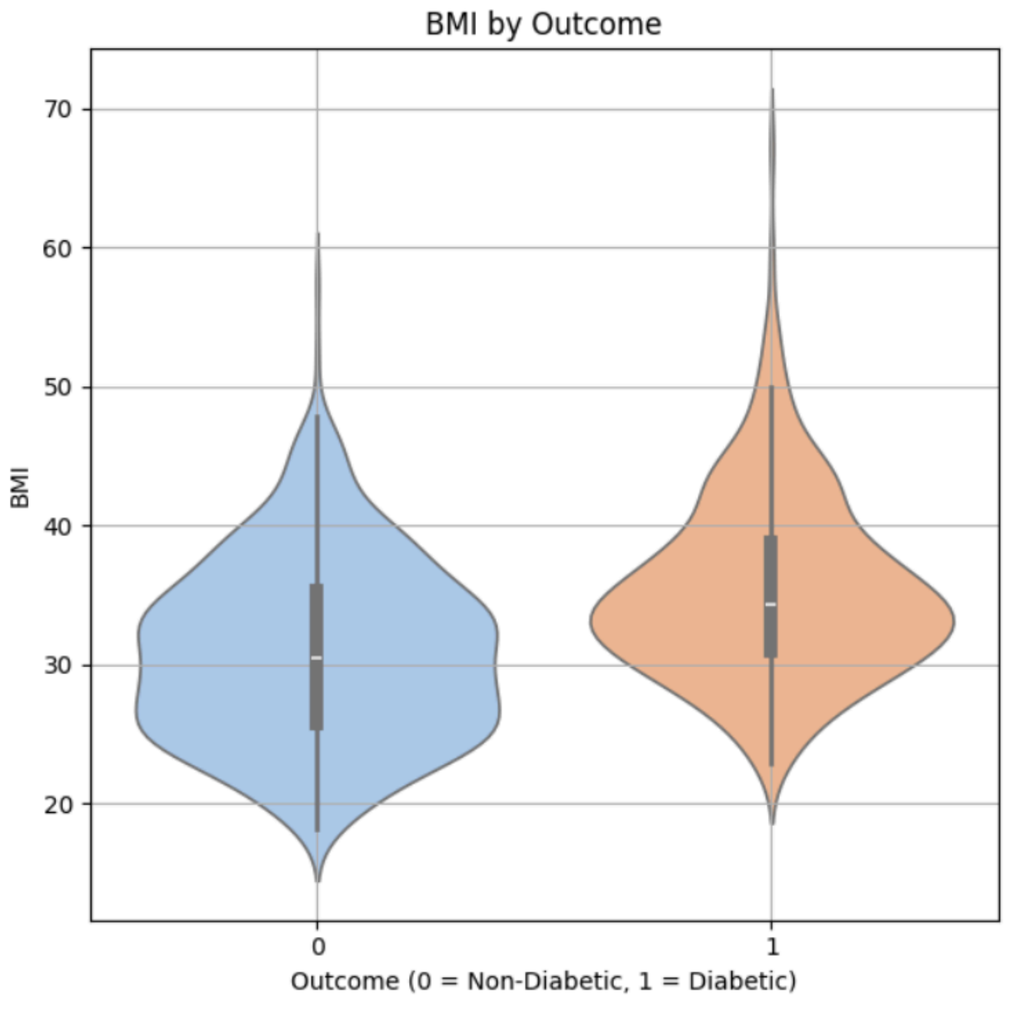
ชุดข้อมูลนี้เกี่ยวกับ ผู้ป่วยหญิง อายุ 21 ปีขึ้นไป จำนวน 769 Rows
โดยมีปัจจัยที่อาจส่งผลต่อการเป็นโรคเบาหวาน
มีจำนวนข้อมูลทั้งหมด 9 หลัก ดังนี้ ..

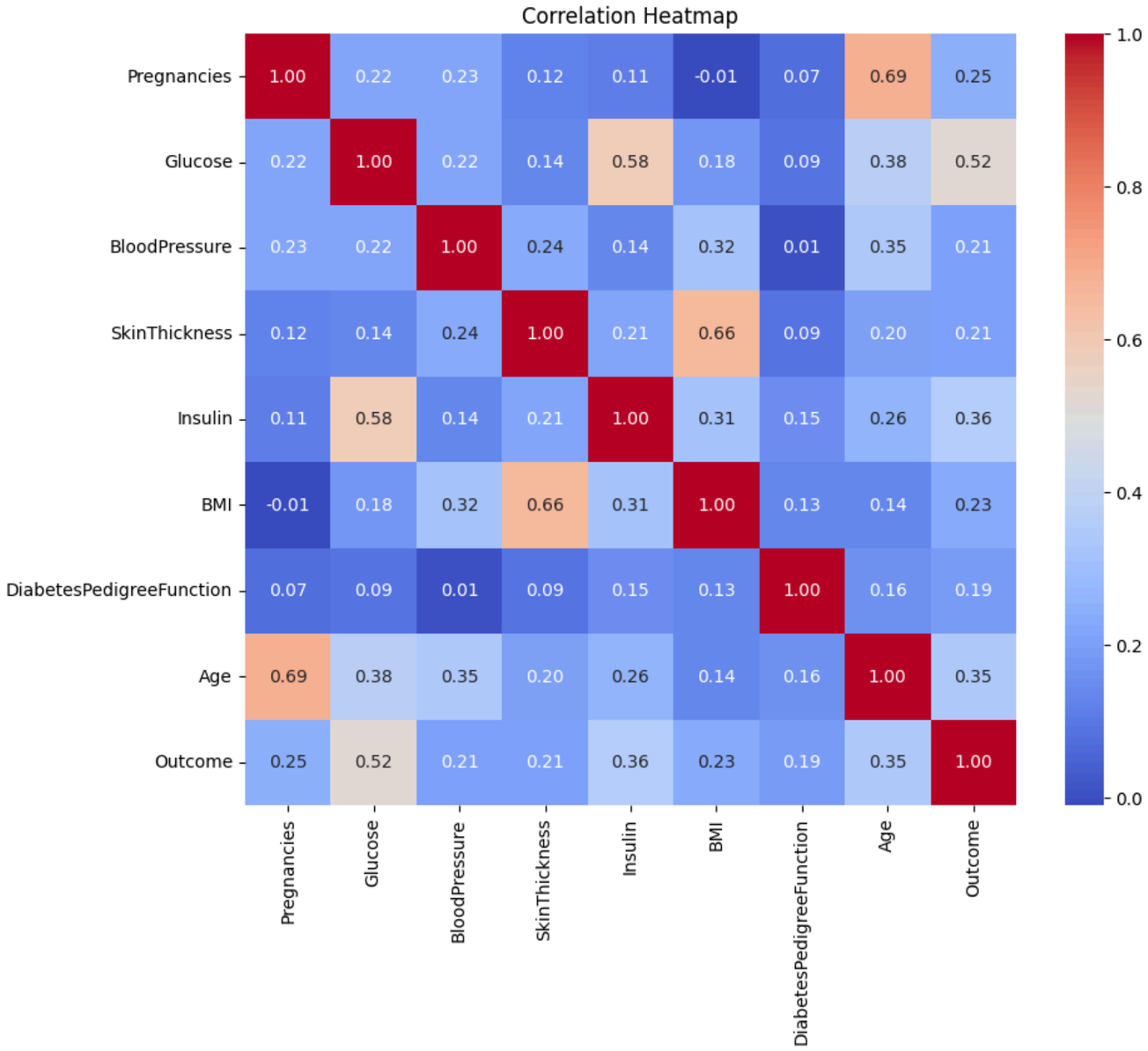
1) Pregnancies (Int) : จำนวนครั้งที่ตั้งครรภ์
2) Glucose (Int) : ระดับน้ำตาลกลูโคสในเลือด
3) BloodPressure (Int) : ระดับความดันโลหิต
4) SkinThickness (Int) : ความหนาของผิวหนัง
5) Insulin (Int) : ระดับฮอร์โมนอินซูลินในเลือด
6) BMI (Float) : ดัชนีมวลกาย
7) DiabetesPedigreeFunction (Float) : อัตราส่วนของการเป็นโรคเบาหวาน
8) Age (Int) : อายุ
9) Outcome (Int) : ผลลัพท์ที่ได้ (0 คือ ไม่เป็นโรคเบาหวาน และ 1 คือ เป็นโรคเบาหวาน)



YOU ARE LIKELY TO HAVE DIABETIES
IF YOU HAVE HIGH GLUECOSE


Insulin Levels by Outcome
Insulin Level
350
300
250
200
150
100
50
0
0
1
Outcome (0 = Non-Diabetic, 1 = Diabetic)


BMI by Outcome
BMI
70
60
50
40
30
20
0
1
Outcome (0 = Non-Diabetic, 1 = Diabetic)



SOME USE FUL INFO ON CORRELATION HEATMAP

- YOUR INSULIN INCREASEAS GLUECOSE INCREASE
  - INSULIN JOB IS TO MOVE GLUCOSE INTO BODY CELL
- HIGHER AGE MORE PREGNANCIES

DATA MANIPULATE / CLEANSING

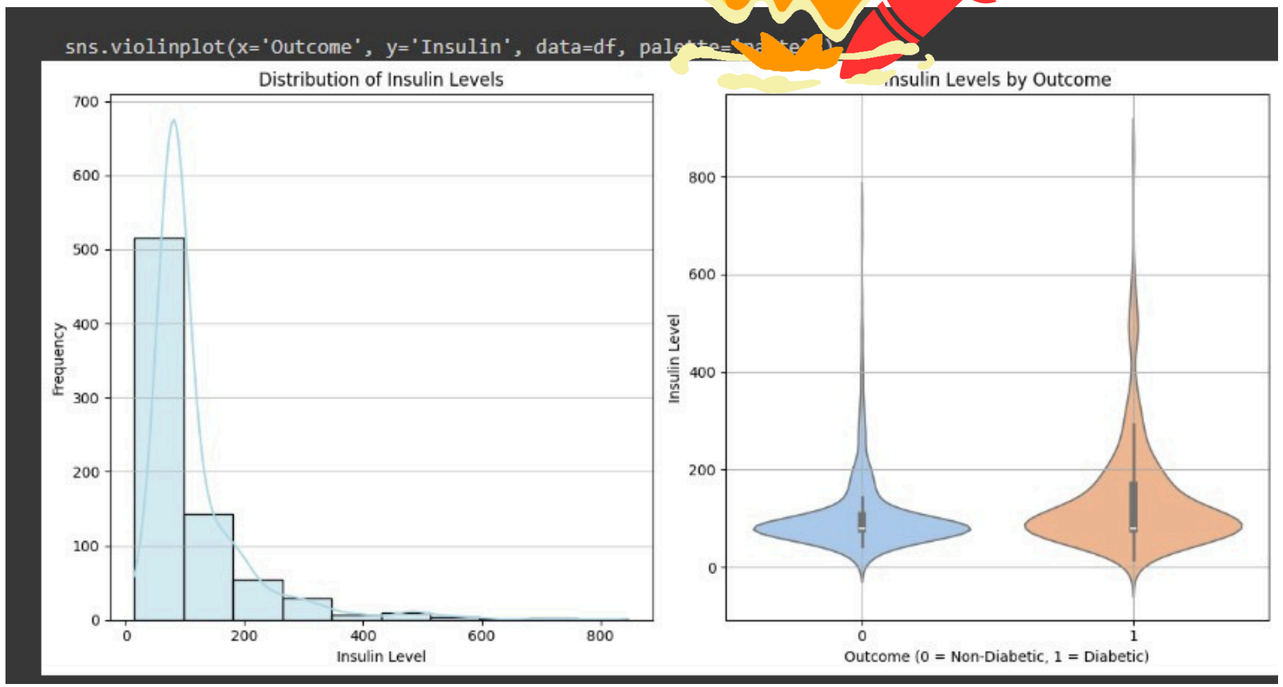
ZERO VALUE

```
1 (df
```

Pregna…	111
Glucose	5
BloodPressure	35
SkinThickness	227
Insulin	374
BMI	11
DiabetesPedigreeFunction	0
Age	0
Outcome	500

MASSIVE C…

```
sns.violinplot(x='Outcome', y='Insulin', data=df, palette=…
```



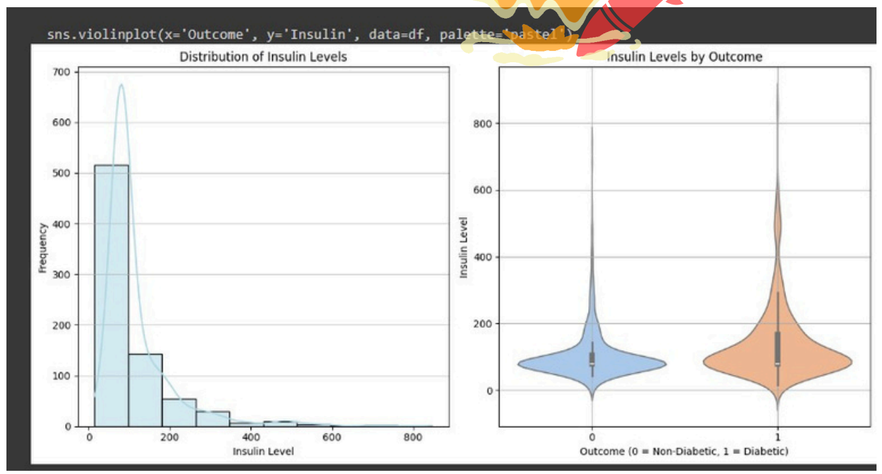
DATA MANIPULATE / CLEANSING

ZERO VALUE

```
1 (df == 0).sum()
```

Pregnancies	111
Glucose	5
BloodPressure	35
SkinThickness	227
Insulin	374
BMI	11
DiabetesPedigreeFunction	0
Age	0
Outcome	500

MASSIVE OUTLINER



- Outliner provide Information that has no statistical significance

- Fill zero with mean broke distribution
- Drop zero lost half of the data
  - easy for beginner (like us)
  - distribution still look nice
  - quality > quantity

DATA PREPROCESSSING

```python
X = df.drop('Outcome', axis=1)  # คอลัมน์ Features
y = df['Outcome']               # คอลัมน์ Target


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

column_transformer_scale = ColumnTransformer(
    transformers=[
        ('scaler', StandardScaler(), ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']),
    ],
    remainder='passthrough'  # Keep other columns unchanged
)

column_transformer_normalize = ColumnTransformer(
    transformers=[
        ('normalize', MinMaxScaler(), ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']),
    ],
    remainder='passthrough'  # Keep other columns unchanged
)
```

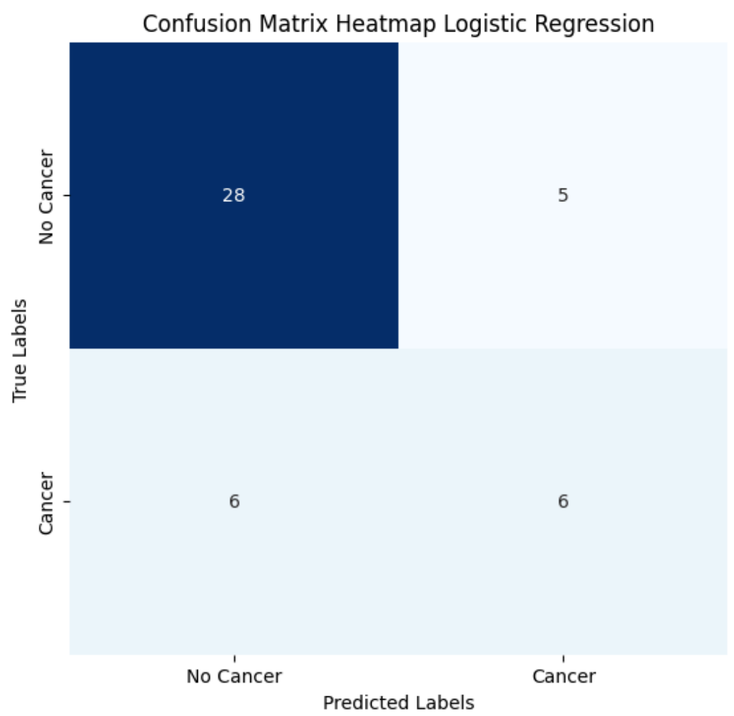
DATA PREPROCESSSING

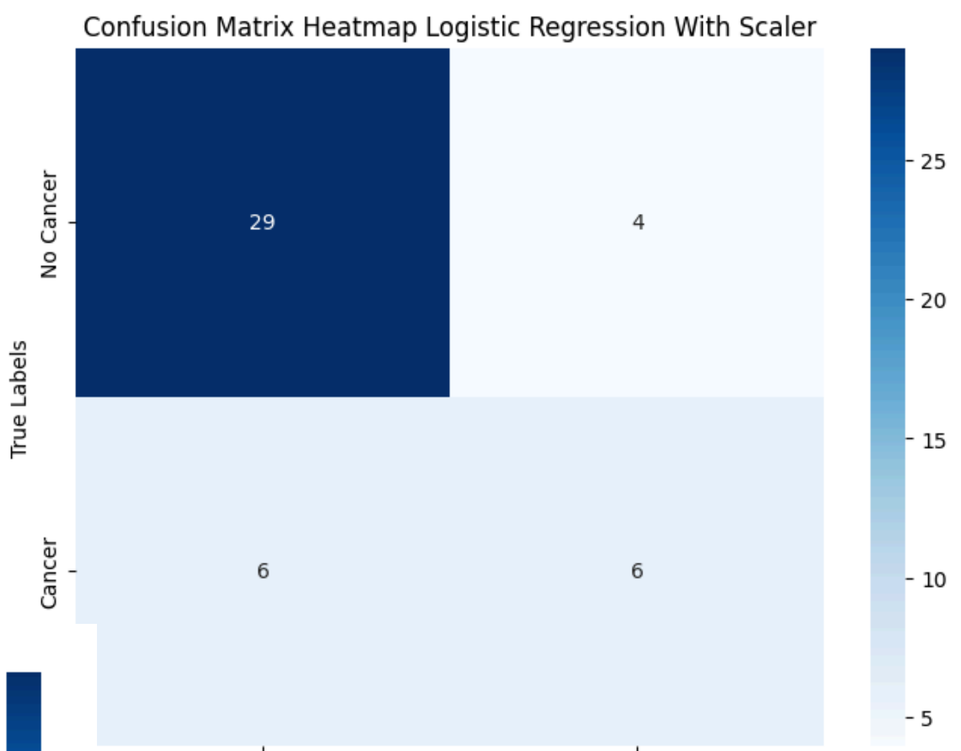
```python
### Logistic Regression
Logistic_pipeline = Pipeline([
    ('classifier', LogisticRegression())
])

Logistic_pipeline_Scaler = Pipeline([
    ('scaler', column_transformer_scale),
    ('classifier', LogisticRegression())
])

Logistic_pipeline_Normalize = Pipeline([
    ('normalize', column_transformer_normalize),
    ('classifier', LogisticRegression())
])


### GaussianNB
GaussianNB_pipeline = Pipeline([
    ('classifier', GaussianNB())
])
GaussianNB_pipeline_Scaler = Pipeline([
    ('scaler', column_transformer_scale),
    ('classifier', GaussianNB())
])
GaussianNB_pipeline_Normalize = Pipeline([
    ('normalize', column_transformer_normalize),
    ('classifier', GaussianNB())
])


### DecisionTreeClassifier
DT_pipeline = Pipeline([
    ('classifier', DecisionTreeClassifier())
])
```

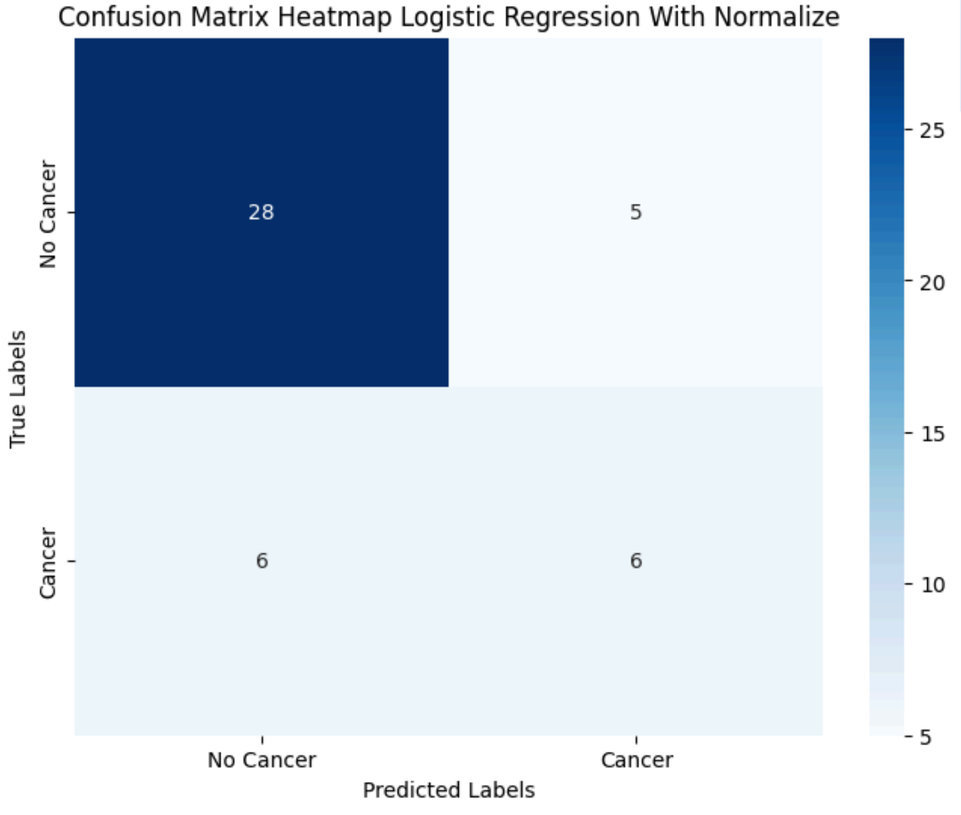
DATA PREPROCESSSING

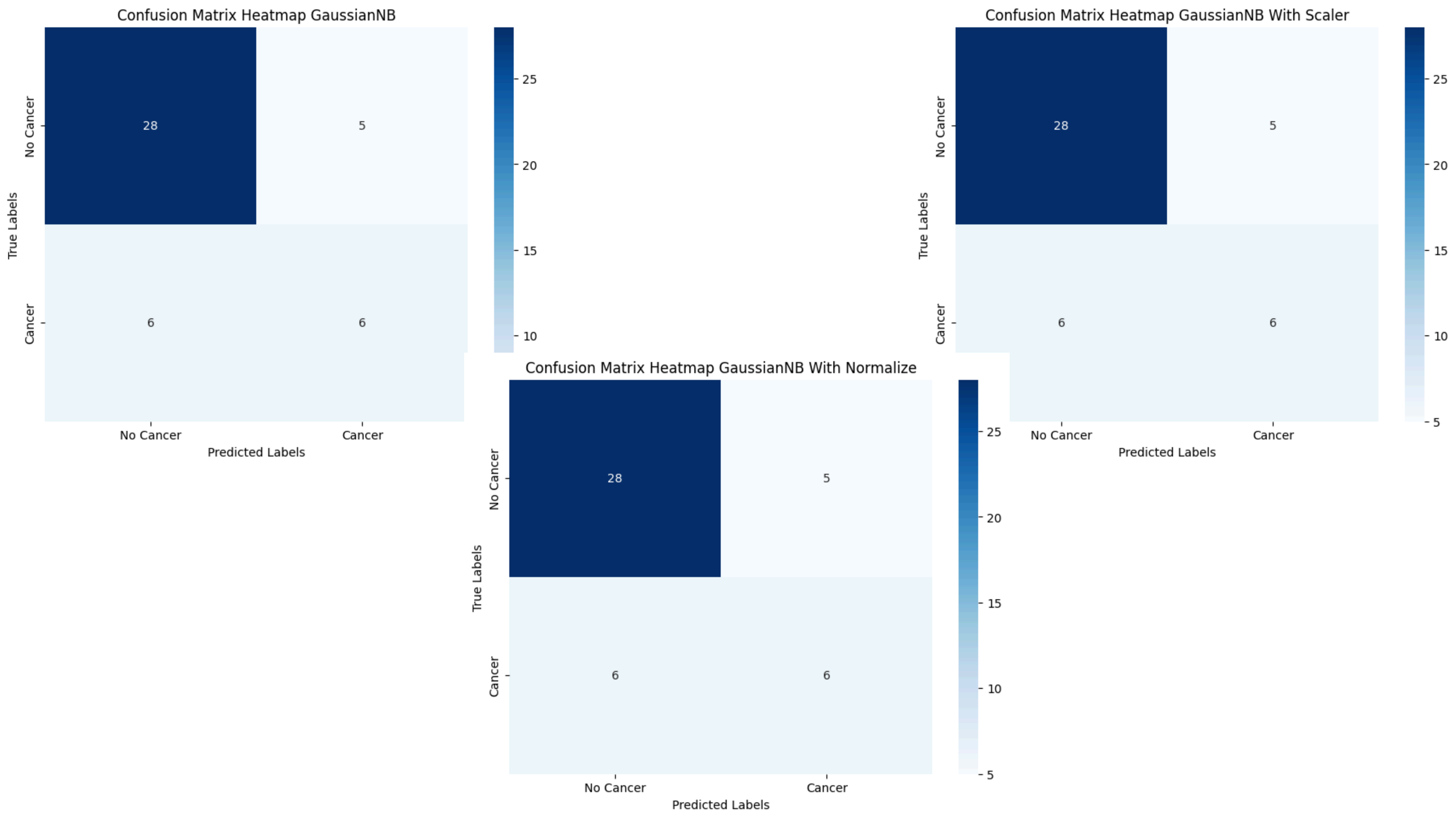
```python
param_grid_lr = {
    'classifier__C': [0.001, 0.01, 0.1, 1, 10, 100],
    'classifier__penalty': ['l1', 'l2'],
    'classifier__solver': ['liblinear', 'saga']
}

grid_search_lr = GridSearchCV(Logistic_pipeline, param_grid_lr, cv=5, scoring='accuracy')
grid_search_lr_sc = GridSearchCV(Logistic_pipeline_Scaler, param_grid_lr, cv=5, scoring='accuracy')
grid_search_lr_nr = GridSearchCV(Logistic_pipeline_Normalize, param_grid_lr, cv=5, scoring='accuracy')
# Fit the pipeline and find the best model

grid_search_lr.fit(X_train, y_train)
grid_search_lr_sc.fit(X_train, y_train)
grid_search_lr_nr.fit(X_train, y_train)

# Get the best score
best_score_ = grid_search_lr.best_score_
best_score_sc = grid_search_lr_sc.best_score_
best_score_nr = grid_search_lr_nr.best_score_

# Print the best score
print("Best cross-validation accuracy for Logistic Regression:", best_score_)
print("Best cross-validation accuracy for Logistic Regression With Scaler:", best_score_sc)
print("Best cross-validation accuracy for Logistic Regression With Normalize:", best_score_nr)

# Evaluate on the test set
y_pred_lr = grid_search_lr.predict(X_test)
accuracy_lr = accuracy_score(y_test, y_pred_lr)
y_pred_lr_sc = grid_search_lr_sc.predict(X_test)
accuracy_lr_sc = accuracy_score(y_test, y_pred_lr_sc)
y_pred_lr_nr = grid_search_lr_nr.predict(X_test)
accuracy_lr_nr = accuracy_score(y_test, y_pred_lr_nr)

print("Test set accuracy for Logistic Regression:", accuracy_lr)
print("Test set accuracy for Logistic Regression With Scaler:", accuracy_lr_sc)
print("Test set accuracy for Logistic Regression With Normalize:", accuracy_lr_nr)

```

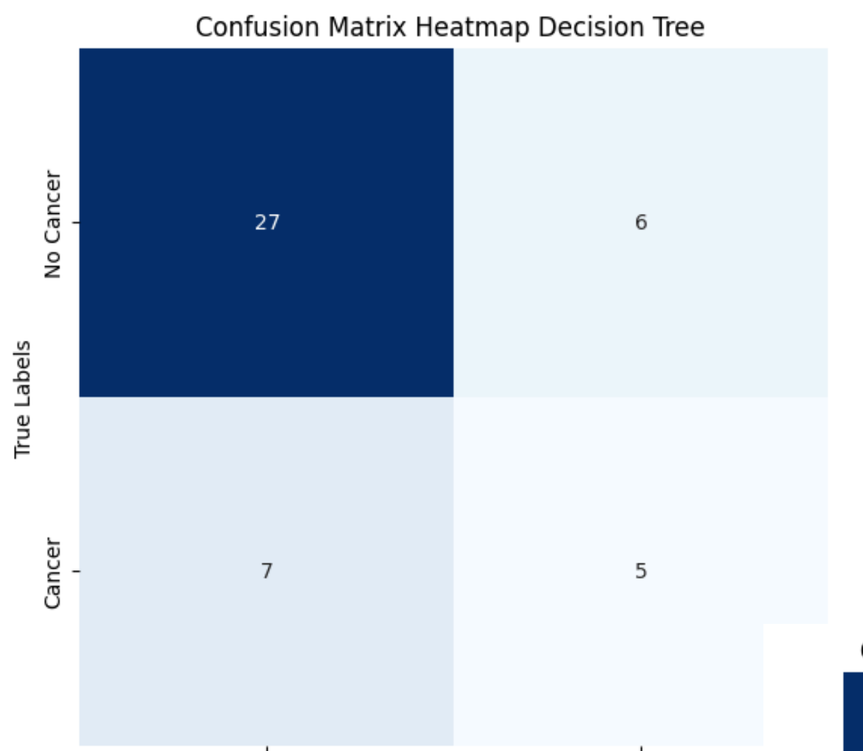
Result

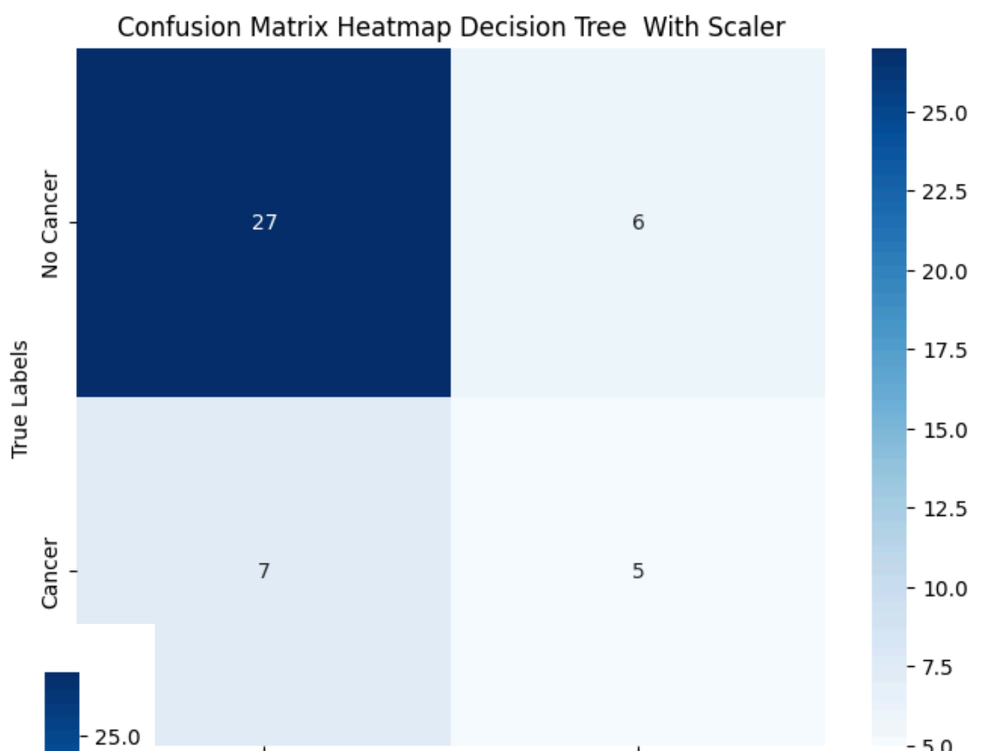

Confusion Matrix Heatmap Logistic Regression
No Cancer
Cancer
True Labels
28
5
6
6
No Cancer
Cancer
Predicted Labels
25
20
15
10

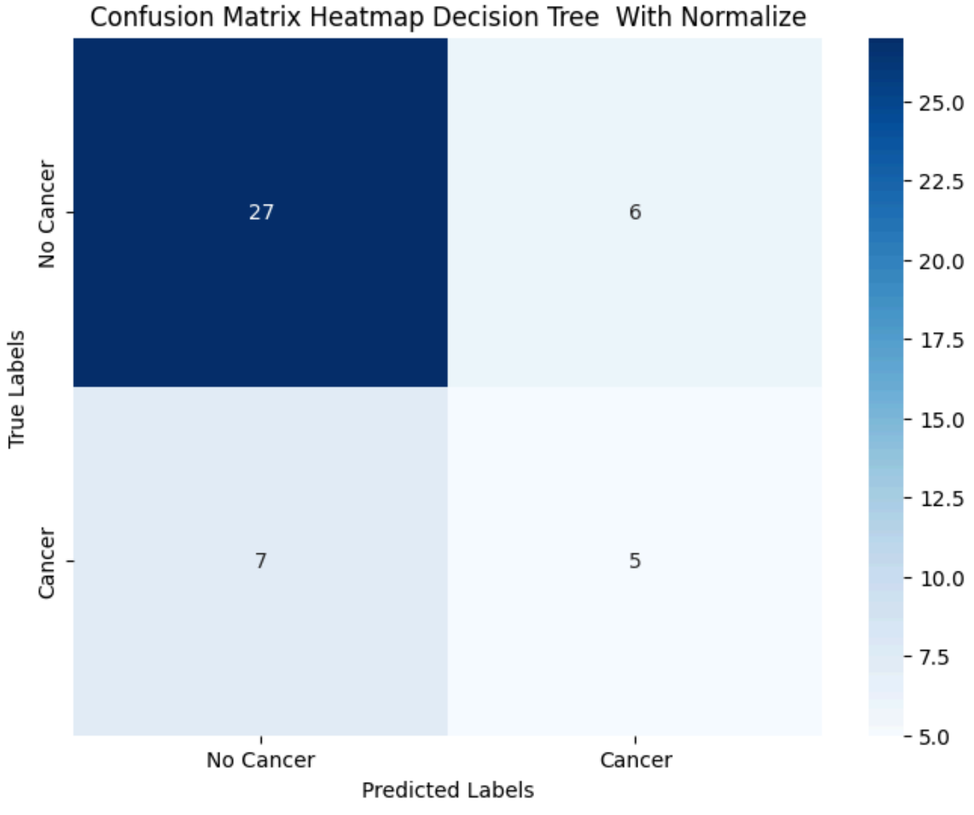

Confusion Matrix Heatmap Logistic Regression With Scaler
No Cancer
Cancer
True Labels
29
4
6
6
No Cancer
Cancer
Predicted Labels
25
20
15
10
5

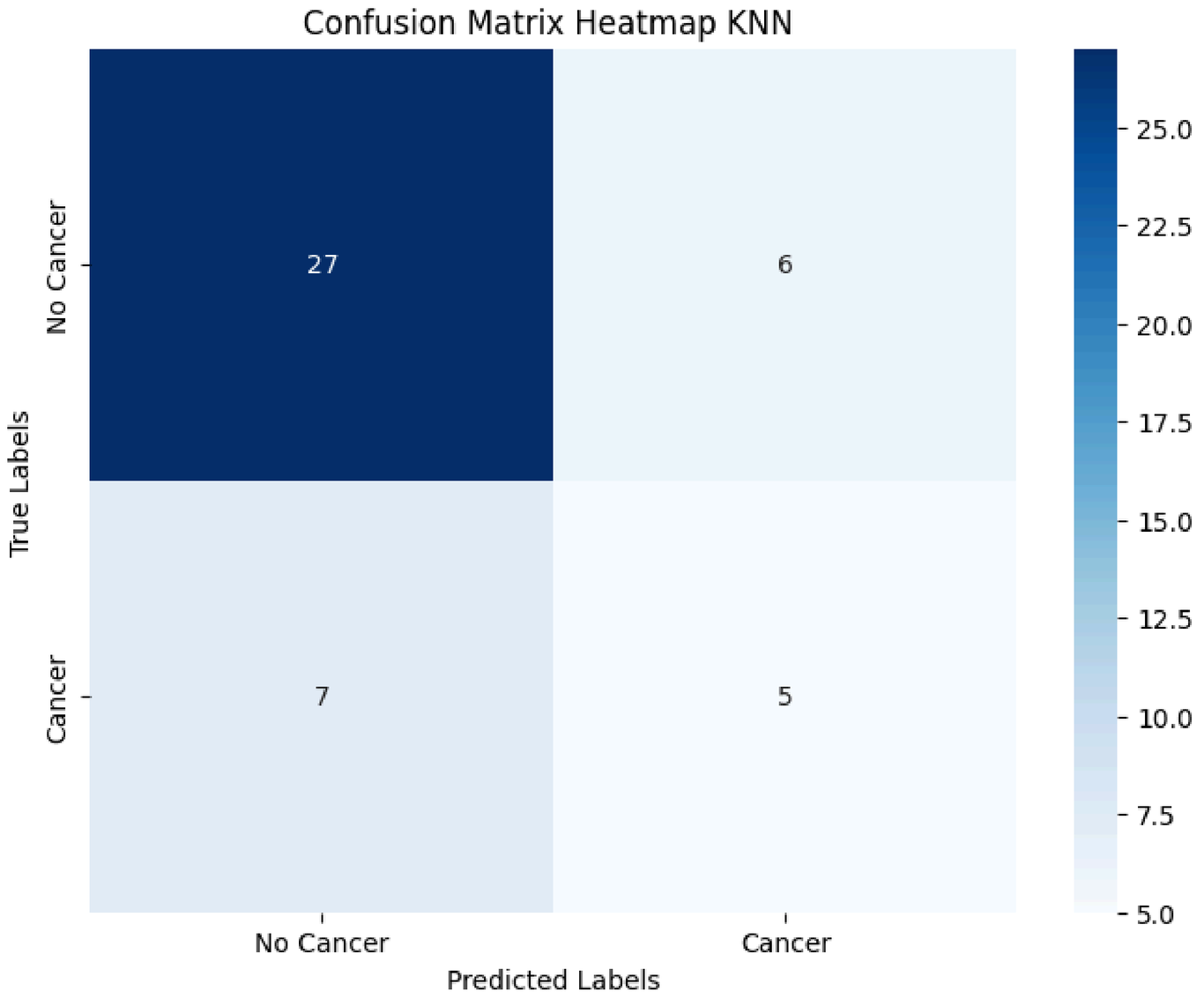

Confusion Matrix Heatmap Logistic Regression With Normalize
No Cancer
Cancer
True Labels
28
5
6
6
No Cancer
Cancer
Predicted Labels
25
20
15
10
5

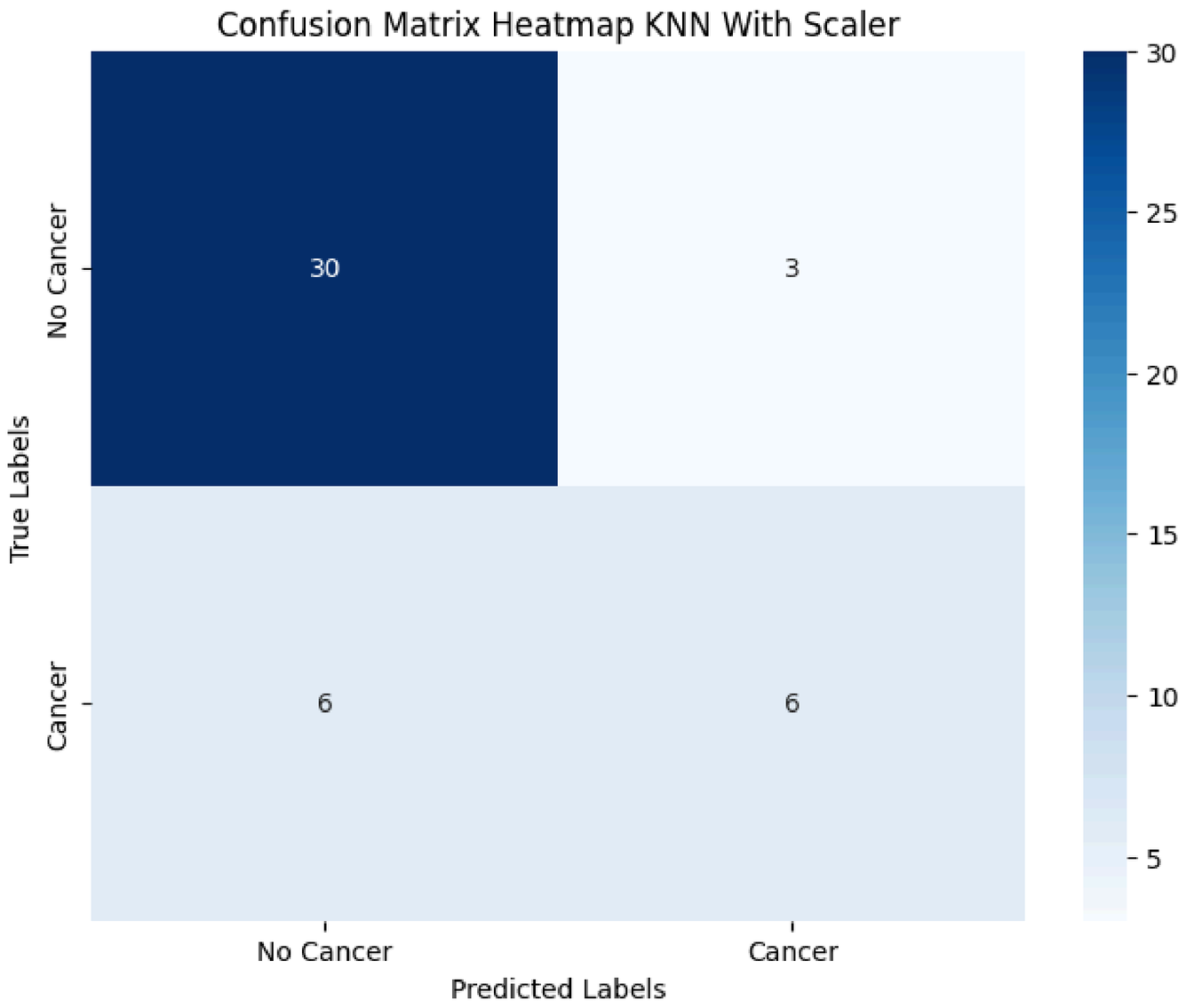

Confusion Matrix Heatmap GaussianNB
No Cancer
Cancer
True Labels
28
5
6
6
No Cancer
Cancer
Predicted Labels
25
20
15
10
Confusion Matrix Heatmap GaussianNB With Scaler
No Cancer
Cancer
True Labels
28
5
6
6
No Cancer
Cancer
Predicted Labels
25
20
15
10
5
Confusion Matrix Heatmap GaussianNB With Normalize
No Cancer
Cancer
True Labels
28
5
6
6
No Cancer
Cancer
Predicted Labels
25
20
15
10
5

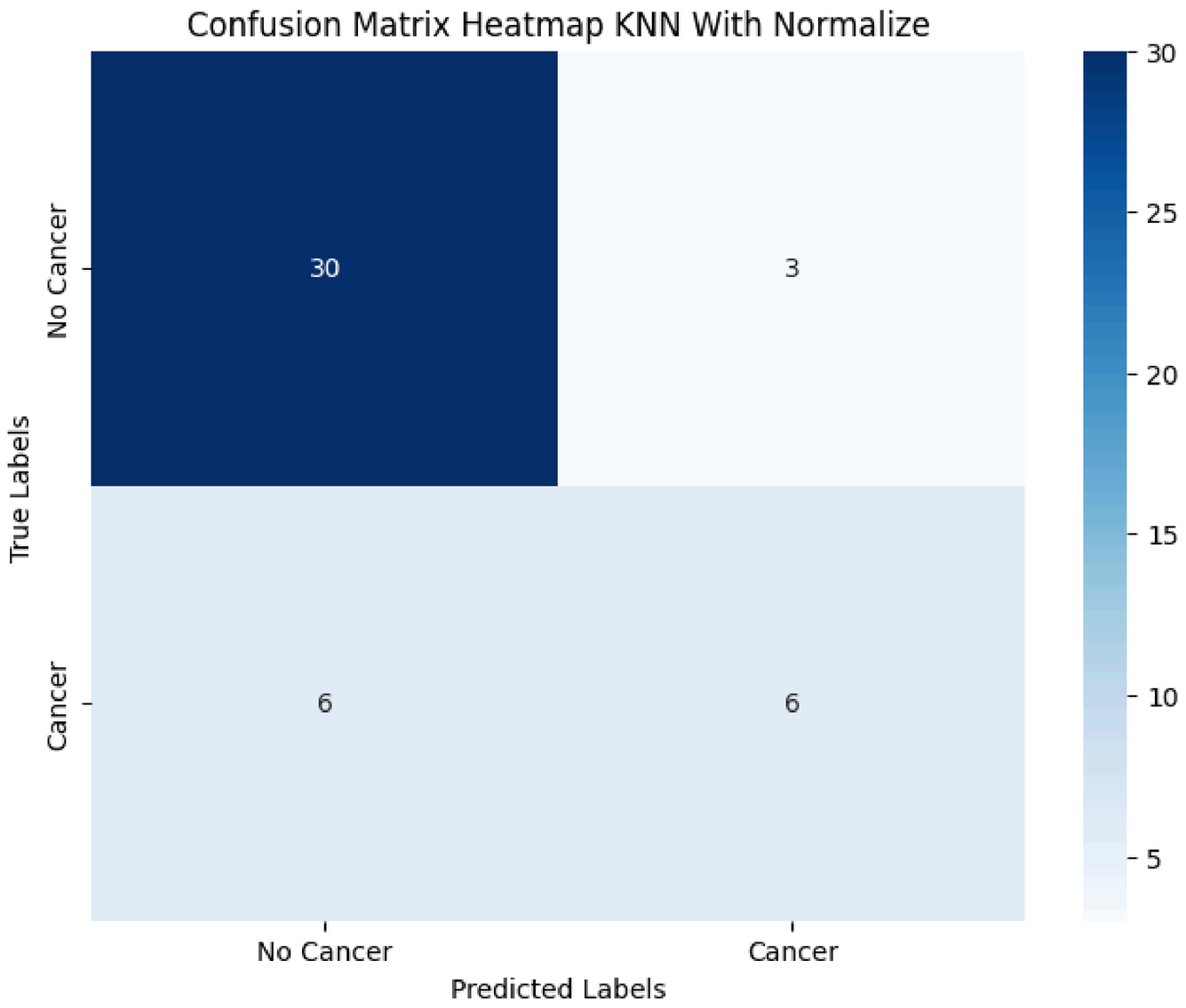

Confusion Matrix Heatmap Decision Tree
No Cancer
Cancer
True Labels
27
6
7
5
No Cancer
Cancer
Predicted Labels
25.0
22.5
20.0
17.5
15.0
12.5
10.0

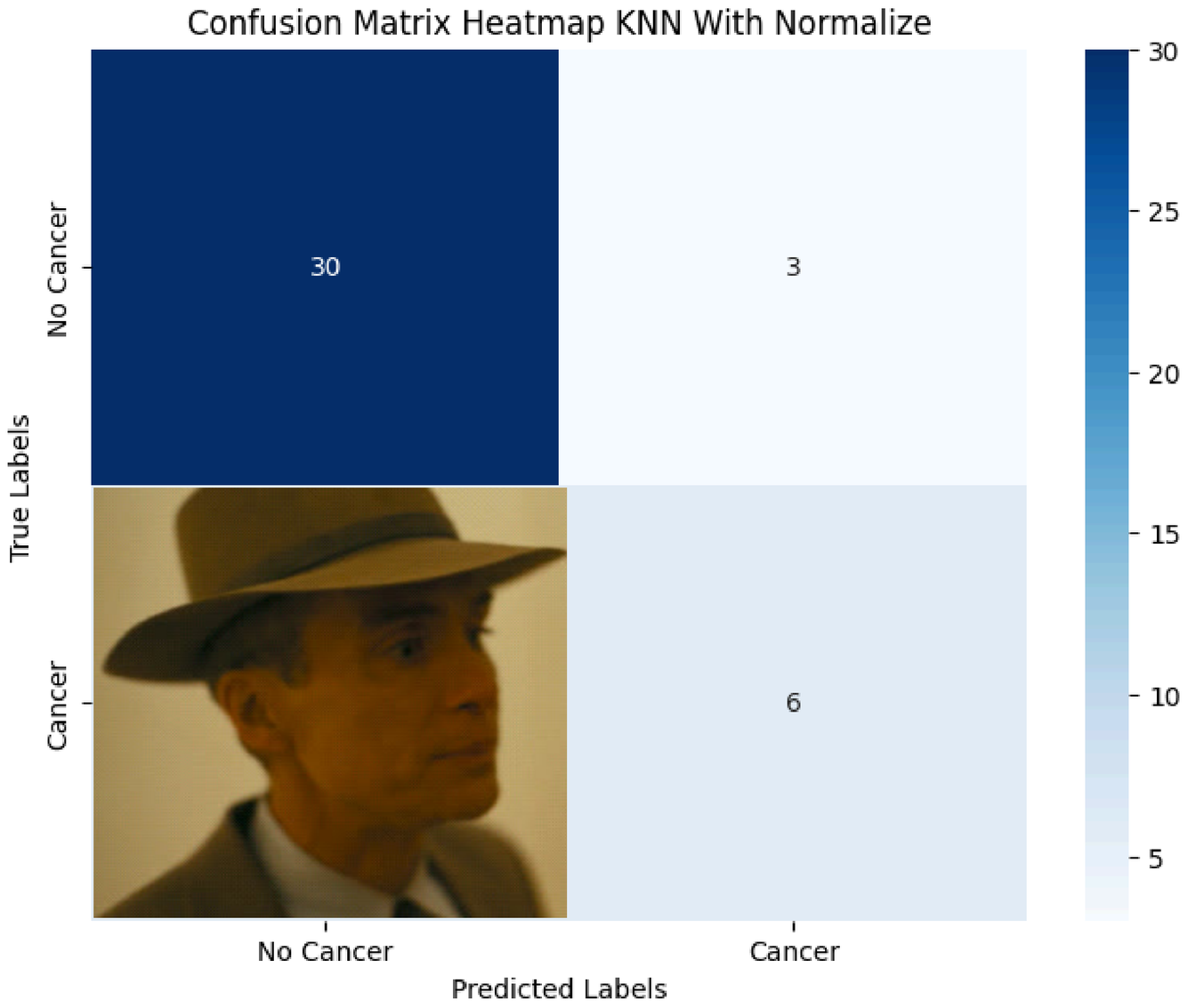

Confusion Matrix Heatmap Decision Tree With Scaler
No Cancer
Cancer
True Labels
27
6
7
5
No Cancer
Cancer
Predicted Labels
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0


Confusion Matrix Heatmap Decision Tree With Normalize
No Cancer
Cancer
True Labels
27
6
7
5
No Cancer
Cancer
Predicted Labels
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0


Confusion Matrix Heatmap KNN
No Cancer
Cancer
True Labels
27
6
7
5
No Cancer
Cancer
Predicted Labels
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0


Confusion Matrix Heatmap KNN With Scaler
True Labels
No Cancer
Cancer
30
3
6
6
No Cancer
Cancer
Predicted Labels
30
25
20
15
10
5


Confusion Matrix Heatmap KNN With Normalize
True Labels
No Cancer
Cancer
30
3
6
6
No Cancer
Cancer
Predicted Labels
30
25
20
15
10
5


Confusion Matrix Heatmap KNN With Normalize
30
3
6
No Cancer
Cancer
True Labels
No Cancer
Cancer
Predicted Labels
30
25
20
15
10
5

EVALUATING

1) หลังจากการทดลองทำนาย Classification Model จาก 3 Model ได้แก่ ..
   1.1) Logistic Regression
   1.2) GaussianNB
   1.3) DecisionTree
   1.3) KNN

   สามารถประเมินและผลลัพท์ที่ได้ว่า ..

" ผลการทำนายนั้นมี F1-Score ที่ไม่ได้ห่างกันมากอย่างมีนัยยะสำคัญ รวมถึงการ Normalize และ Scaler ก็มีการพัฒนาผลขึ้นจากเดิมเพียงเล็กน้อย หากข้อมูลใหญ่กว่านี้อาจจะสังเกตพฤตกิกรรมของโมเดลได้ชัดขึ้นมากกว่านี้ "

2) precision ใน true ต่ำคาดว่าเกิดจาก Imbalanced Dataset

SUMMARY

1) สรุปการทำงานของโมเดล :

> Accuracy : โมเดลมีการใช้ Scaler ที่มีความแม่นยำค่อนข้างสูงกว่าแบบปกติ และ With Normalize
.. ซึ่งอาจแสดงว่า Standard Scaling มีผลช่วยให้โมเดลแยกแยะข้อมูลได้ดีขึ้นเล็กน้อย
> Precision และ Recall สำหรับ " Diabetes " :
- การใช้ Scaler มี Precision และ Recall ที่ดีกว่าเล็กน้อยเมื่อเทียบกับแบบปกติและ With Normalize
- Recall ของ " Diabetes " ในทุกกรณียังคงต่ำ ซึ่งหมายความว่าโมเดลยังพลาดการตรวจจับ "Diabetes" อยู่ในบางกรณี ******
> ความสมดุลระหว่าง Precision และ Recall :
- ค่า F1-Score สำหรับ " Diabetes " ชี้ให้เห็นว่าโมเดลยังมีปัญหาในการแยกแยะตัวอย่างของคลาสที่ไม่สมดุล (Diabetes มีจำนวนตัวอย่างน้อยกว่า)
> ผลกระทบของ Preprocessing :
- การใช้ Scaler ช่วยเพิ่ม Accuracy และ Precision ของคลาส "Diabetes" เล็กน้อย ซึ่งสอดคล้องกับธรรมชาติของความไวต่อการกระจายตัวของข้อมูล
- การ Normalize อาจไม่ช่วยเพิ่มประสิทธิภาพในบางกรณี

2) สรุปการแก้ปัญหาที่ได้ตั้งไว้ :

> ข้อสรุป : ผลการวิเคราะห์แสดงให้เห็นว่า โมเดลที่สร้างสามารถทำนายความเสี่ยงในการ ”ไม่” เป็นโรคเบาหวานได้อย่างมีประสิทธิภาพ
.. โดยมีตัวแปรสำคัญที่มีผลต่อการทำนายคือ น้ำตาลกลูโคส , BMI , และอายุ
> ประโยชน์ของการวิเคราะห์ : โครงงานนี้สามารถนำไปต่อยอดและอาจใช้งานได้จริงในระบบสาธารณสุขเพื่อช่วยในการคัดกรองผู้ป่วยที่อาจมีความเสี่ยงในการเป็นโรคเบาหวาน ซึ่งนอกจากจะช่วยให้สามารถดำเนินการป้องกันและรักษาได้อย่างถูกต้องและทันการรักษาแล้ว ยังสามารถช่วยลดความเสี่ยงต่อการเกิดโรคในระยะยาวได้อีกด้วย

REFERENCE

1) Framingham Heart Study (USA) : found that Insulin and Bloodpressure are both good predictors for Diabetes
> https://www.sciencedirect.com/science/article/pii/S2211816013000021

2) Finnish Diabetes Prevention Study (Finland) : found of controlling Bloodpressure and Insulin can prevent Diabetes Type 2 significantly
> https://watermark.silverchair.com/dc1203003230.pdf?t

2) NHANES Study (USA) : found of high bloodpressure with the heavily body can cause an Insulin Resistance and cause Diabetes Type 2
> https://pubmed.ncbi.nlm.nih.gov/35354619/