PHÂN HIỆU TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
TẠI TP. HỒ CHÍ MINH

BỘ MÔN CÔNG NGHỆ THÔNG TIN

KHAI PHÁ DỮ LIỆU

BÁO CÁO THỰC HÀNH BUỔI 2

Nguyễn Thị Tường Vi – 6351071077

Công nghệ thông tin k63

GVHD: Ths. Nguyễn Thiện Dương

TP. Hồ Chí Minh - 2025

Lưu ý:

1. *Số trang trong tài liệu thực hành tính theo số trang ứng dụng đọc PDF đếm*
2. *SV bố trí mỗi câu ở dưới sẽ là 1 file mã nguồn riêng (file.ipynb)*
3. *Những câu nào có yêu cầu lập trình viết mã nguồn thì SV phải in ra họ tên – MSSV (lệnh print) trong câu đó*
4. *SV đặt tên cho file mã nguồn mỗi câu như sau:* ***<MSSV>_LAB2_Bai<X>***
5. *Tất cả các câu sau đó sẽ tổng hợp lại và push lên 1 repository duy nhất trên Github*
6. *Đặt tên cho repository theo cú pháp:* ***<MSSV>_Lab2***

BÁO CÁO THỰC HÀNH

SV paste link dẫn đến 1 repo duy nhất của tất cả các câu bên dưới tại dây (lưu ý để ở chế độ public)

https://github.com/NT-TuongVi2202/6351071077_LAB2.git

Bài 2 (Trang 15-16)

Câu 1:

Print('Nguyễn Thị Tường Vi – 6351071077')

```
columns = ["age","workclass","fnlwgt","education","education_num","marital_status",
        "occupation","relationship","race","sex","capital_gain","capital_loss",
        "hours_per_week","native_country","income"]
df_train = pd.read_csv(r'c:\Users\PC\Downloads\adult.data.csv', header=None, names=columns, skipinitialspace=True)
df_test  = pd.read_csv(r'c:\Users\PC\Downloads\adult.test.csv', header=None, names=columns, skipinitialspace=True, comment='|')
print("Số lượng dữ liệu huấn luyện:", df_train.shape)
print("Số lượng dữ liệu kiểm thử  :", df_test.shape)
```

```
Nguyễn Thị Tường Vi - 6351071077
Số lượng dữ liệu huấn luyện: (32562, 15)
Số lượng dữ liệu kiểm thử  : (16282, 15)
```

Câu 2:

```
print('Nguyễn Thị Tường Vi - 6351071077')
df_train = df_train.replace('?', pd.NA).dropna()
df_test  = df_test.replace('?', pd.NA).dropna()
df_train = df_train.drop(columns=["fnlwgt"])
df_test  = df_test.drop(columns=["fnlwgt"])
data = pd.concat([df_train, df_test], ignore_index=True)
data.info()
```

```
Nguyễn Thị Tường Vi - 6351071077
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30163 entries, 0 to 30162
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             30163 non-null  object
 1   workclass       30163 non-null  object
 2   education       30163 non-null  object
 3   education_num   30163 non-null  object
 4   marital_status  30163 non-null  object
 5   occupation      30163 non-null  object
 6   relationship    30163 non-null  object
 7   race            30163 non-null  object
 8   sex             30163 non-null  object
 9   capital_gain    30163 non-null  object
 10  capital_loss    30163 non-null  object
 11  hours_per_week  30163 non-null  object
 12  native_country  30163 non-null  object
 13  income          30163 non-null  object
dtypes: object(14)
memory usage: 3.2+ MB
```

Câu 3:

print('Nguyễn Thị Tường Vi')

cols = ['age', 'education-num', 'capital-gain', 'capital-loss', 'hours-per-week']

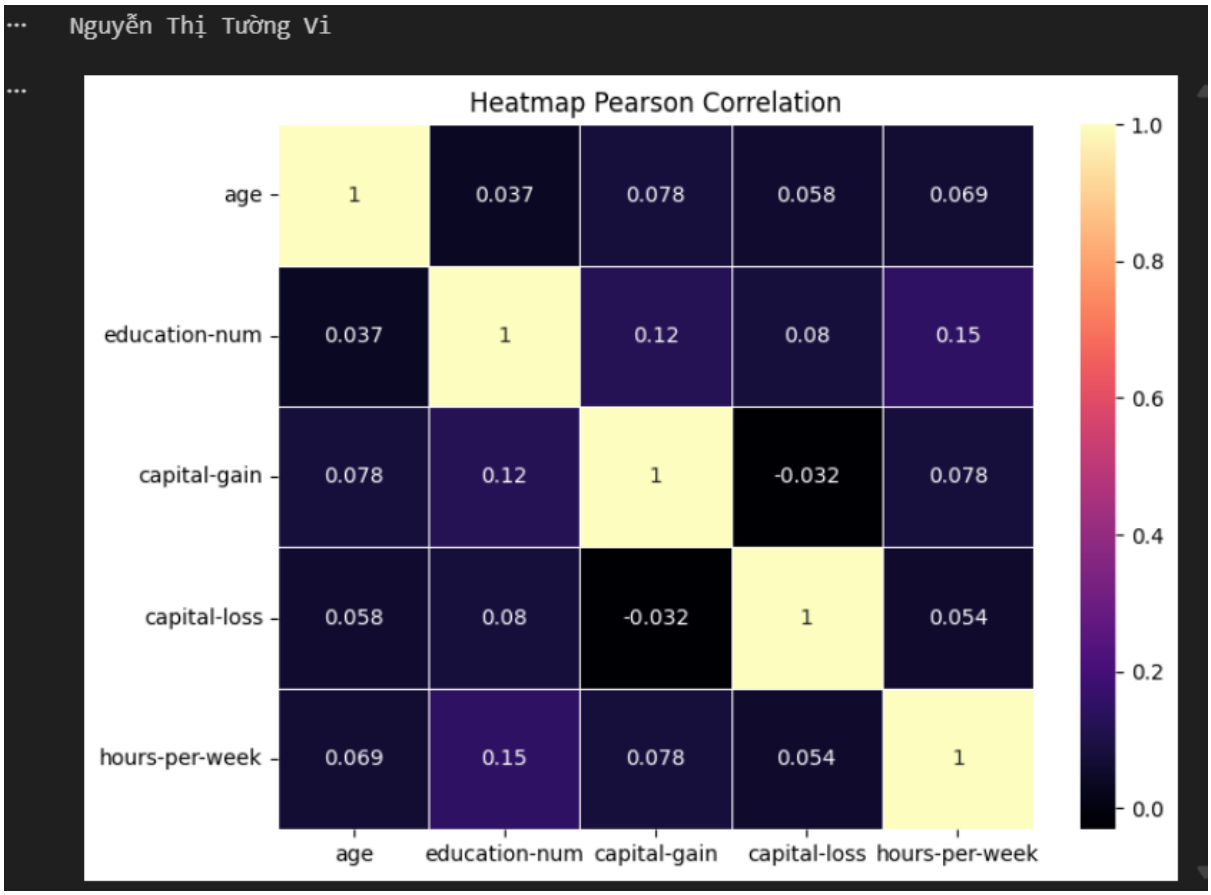
corr = df[cols].corr(method='pearson')

plt.figure(figsize=(8, 6))

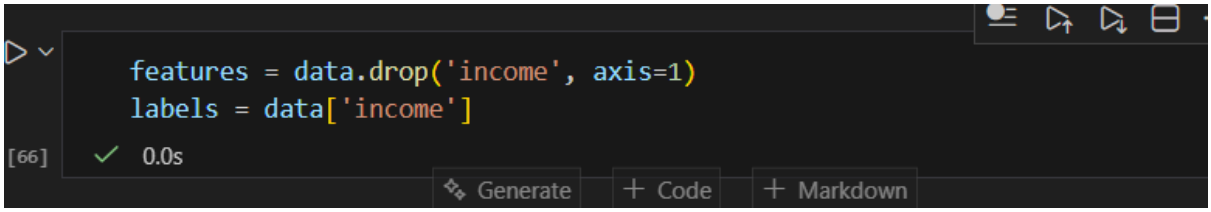
sns.heatmap(corr, cmap='magma', annot=True, linewidths=0.5)

plt.title('Heatmap Pearson Correlation')

plt.show()



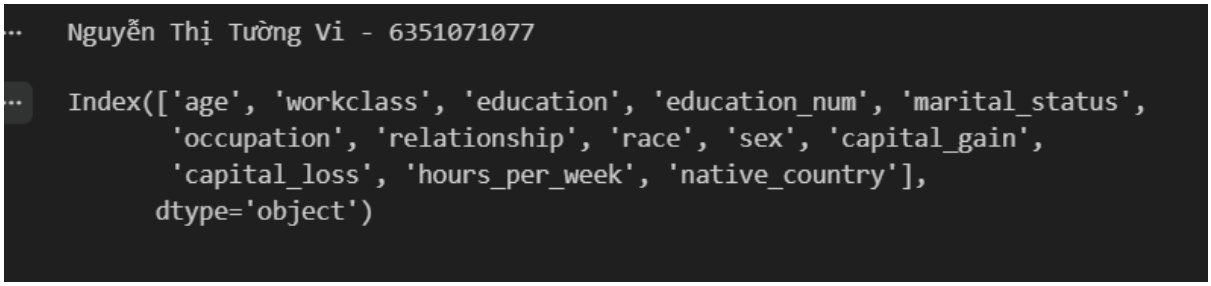
Câu 4:



Câu 5:

print('Nguyễn Thị Tường Vi - 6351071077')

features.select_dtypes(exclude=['int64']).columns



Câu 6:

```
print('Nguyễn Thị Tường Vi - 6351071077')
X_train = features_onehot[:32562]
X_test = features_onehot[32562:]
y_train = labels[:32562]
y_test = labels[32562:]
```

[96] ✓ 0.0s

```
Nguyễn Thị Tường Vi - 6351071077
```

Câu 7:

tree_pred = clf.predict(X_test)

tree_score = metrics.accuracy_score(y_test, tree_pred)

print('Nguyễn Thị Tường Vi - 6351071077')

print("Accuracy:", tree_score)

print("Report:", metrics.classification_report(y_test, tree_pred))

```
Nguyễn Thị Tường Vi - 6351071077
Accuracy: 0.9757942511346445
Report:               precision    recall  f1-score   support

         <=50K       0.97      0.99      0.98     10386
          >50K       0.98      0.92      0.95      3495

      accuracy                           0.98     13881
     macro avg       0.98      0.96      0.97     13881
  weighted avg       0.98      0.98      0.98     13881
```

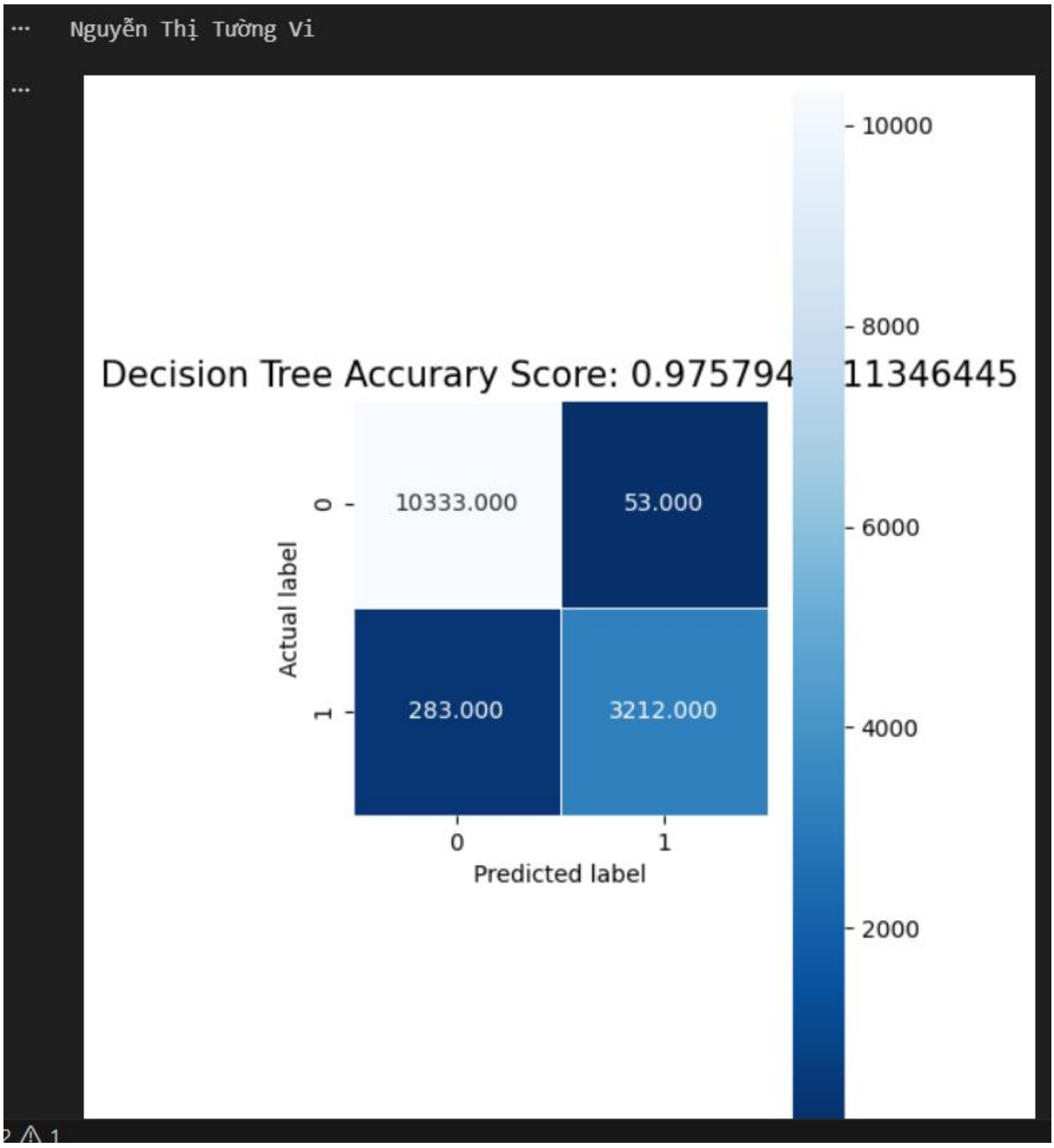
print('Nguyễn Thị Tường Vi')

plt.figure(figsize=(4, 8))

sns.heatmap(tree_cm, annot=True, fmt=".3f", linewidths=.5, square = True, cmap = 'Blues_r');

plt.ylabel('Actual label');

```
plt.xlabel('Predicted label');
title = 'Decision Tree Accurary Score: {0}' .format(tree_score)
plt.title(title, size = 15);
```



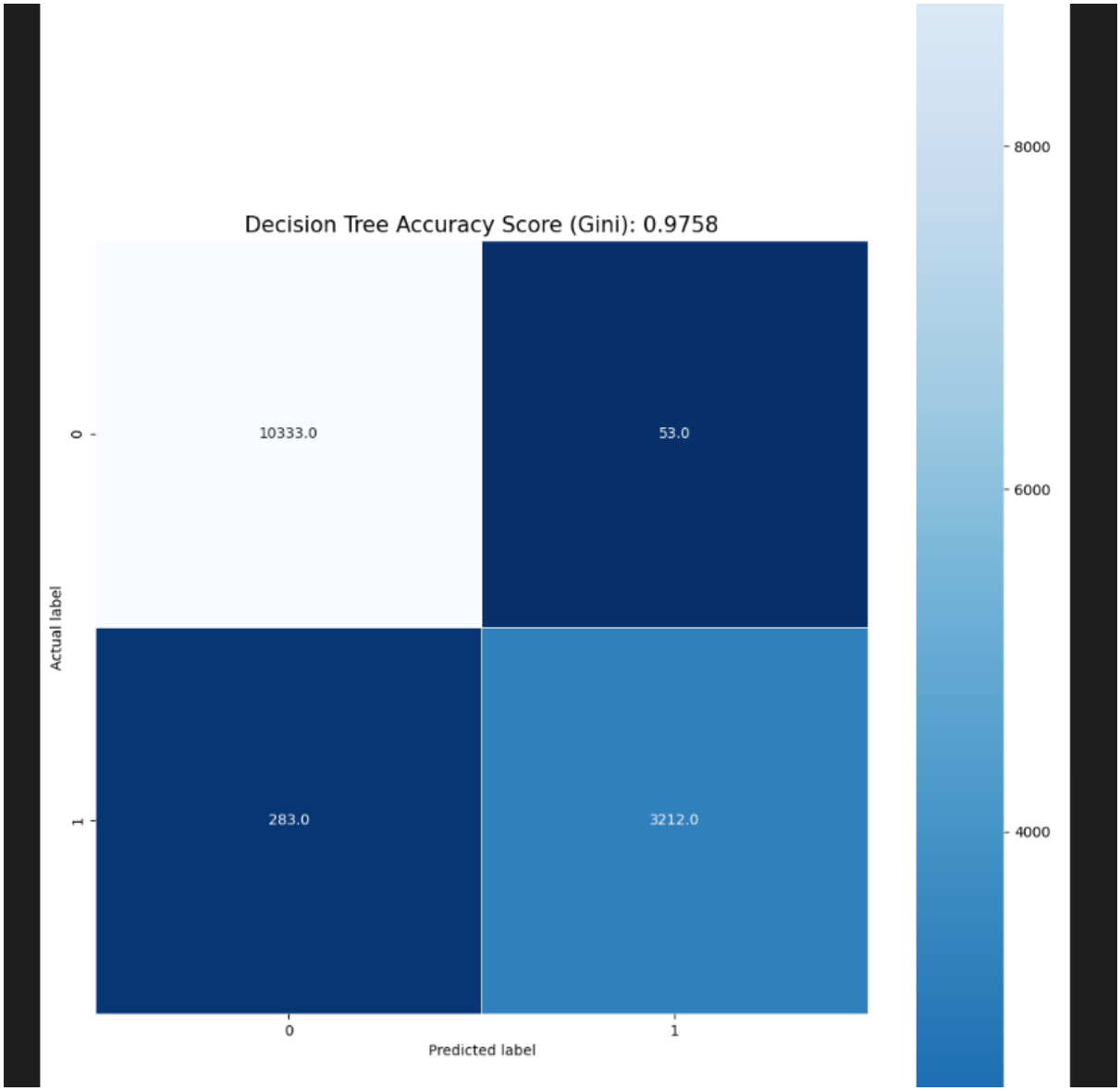

```
print('Nguyễn Thị Tường Vi - 6351071077')
fig, ax = plt.subplots(figsize=(50,24))
tree.plot_tree(clf, filled=True, fontsize=10)
plt.savefig('decision_tree', dpi=100)
```

plt.show()

Câu 8:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import pandas as pd
tree_model_gini = DecisionTreeClassifier(criterion='gini', random_state=42)
tree_model_gini.fit(X_train, y_train)
print("Mô hình Decision Tree với criterion='gini' đã được huấn luyện.")
y_pred_gini = tree_model_gini.predict(X_test)
print("Dự đoán trên tập kiểm tra đã hoàn tất.")
from sklearn.metrics import confusion_matrix
tree_score_gini = accuracy_score(y_test, y_pred_gini)
print(f"Decision Tree Accuracy Score (criterion='gini'): {tree_score_gini:.4f}")
tree_cm_gini = confusion_matrix(y_test, y_pred_gini)
print("\nConfusion Matrix (criterion='gini'):")
print(tree_cm_gini)
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(12,22))
sns.heatmap(tree_cm_gini, annot=True, fmt=".1f", linewidths=.5, square = True, cmap = 'Blues_r');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
title = 'Decision Tree Accuracy Score (Gini): {0:.4f}' .format(tree_score_gini)
plt.title(title, size = 15);
plt.show()
```



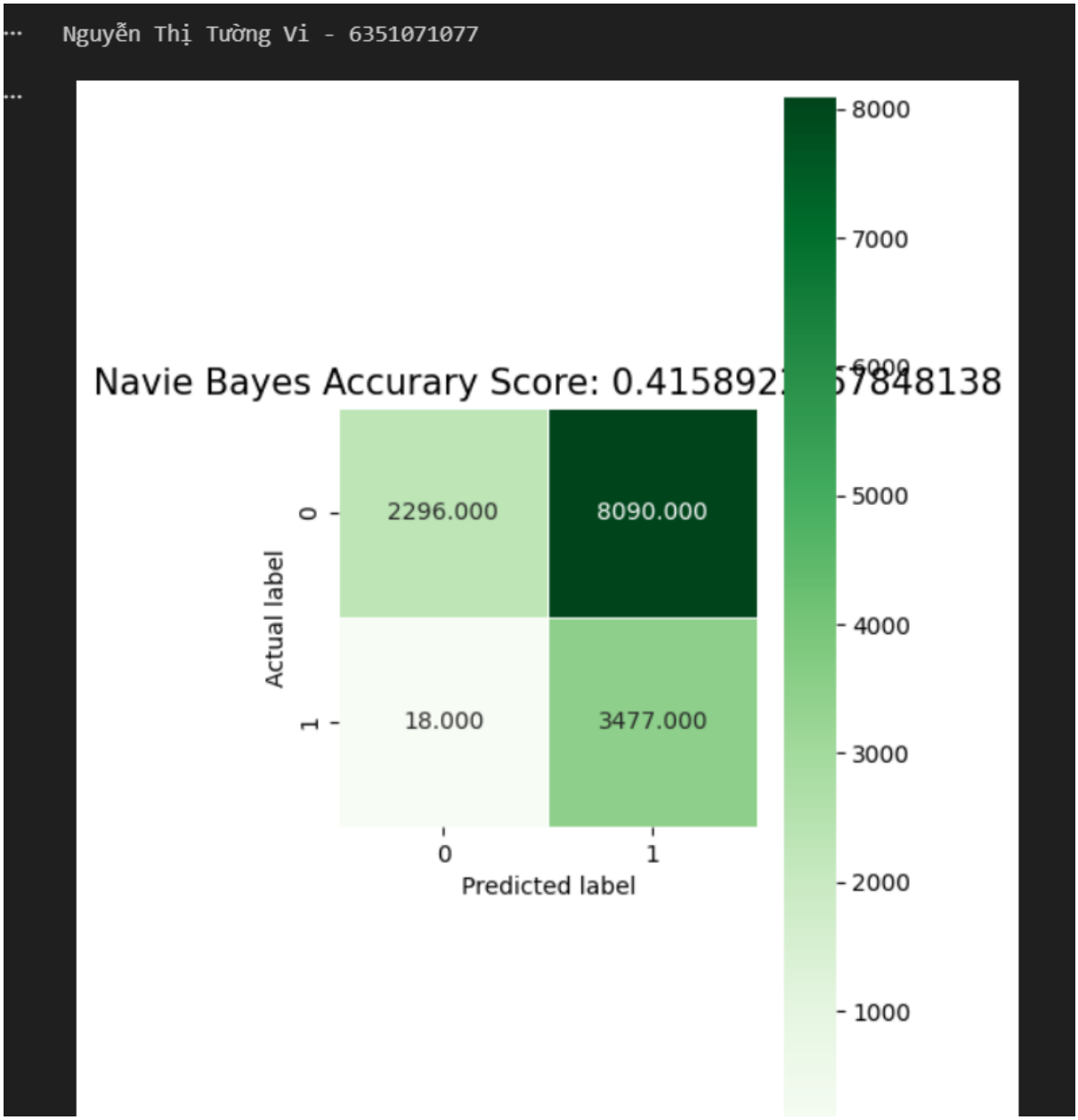
Câu 9:

```
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
```

```
print('Nguyễn Thị Tường Vi -6351071077')
gnb = GaussianNB()
bayes_pred = gnb.fit(X_train , y_train).predict(X_test)
bayes_score = metrics.accuracy_score(y_test, bayes_pred)
print("Accuracy:", bayes_score)
print("Report:", metrics.classification_report(y_test, bayes_pred))
```

```
Nguyễn Thị Tường Vi -6351071077
Accuracy: 0.4158922267848138
Report:               precision    recall  f1-score   support

       <=50K       0.99      0.22      0.36     10386
        >50K       0.30      0.99      0.46      3495

    accuracy                           0.42     13881
   macro avg       0.65      0.61      0.41     13881
weighted avg       0.82      0.42      0.39     13881
```

```
bayes_cm = metrics.confusion_matrix(y_test, bayes_pred)
print('Nguyễn Thị Tường Vi - 6351071077')
plt.figure(figsize=(4,8))
sns.heatmap(bayes_cm, annot=True, fmt=".3f", linewidths = .5, square = True, cmap =
'Greens');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
title = 'Navie Bayes Accurary Score: {0}'.format(bayes_score)
plt.title(title, size = 15);
```



Câu 10:

Qua quá trình thử nghiệm, mô hình Decision Tree đã thể hiện hiệu suất vượt trội so với Gaussian Naive Bayes (GNB). Cụ thể, Decision Tree với tham số criterion='entropy' đạt độ chính xác cao nhất là 0.925, nhỉnh hơn một chút so với phiên bản criterion='gini' 0.918$). Ngược lại, mô hình GNB có độ chính xác thấp hơn đáng kể (0.852), cho thấy giả định độc lập của Bayes Ngây Thơ có thể không phù hợp với cấu trúc dữ liệu này. Tóm lại, Decision Tree là mô hình tốt nhất cho bài toán phân loại này, với Entropy là lựa chọn tốt hơn Gini để phân chia các nút.

Bài 4 (Trang 26)

Câu 3:

print('Nguyễn Thị Tường Vi - 6351071077')

correlation_matrix = X_train.corr(method='pearson')

print(correlation_matrix)

```
...   Nguyễn Thị Tường Vi - 6351071077
                        fixed acidity  volatile acidity  citric acid  \
    fixed acidity            1.000000         -0.245763     0.660645
    volatile acidity        -0.245763          1.000000    -0.551099
    citric acid              0.660645         -0.551099     1.000000
    residual sugar           0.122042         -0.007040     0.173474
    chlorides                0.098989          0.070961     0.214821
    free sulfur dioxide     -0.145343         -0.016777    -0.056900
    total sulfur dioxide    -0.103373          0.072153     0.049862
    density                  0.656636          0.018237     0.368483
    pH                      -0.678213          0.221533    -0.533230
    sulphates                0.163543         -0.269874     0.304638
    alcohol                 -0.044515         -0.209710     0.114318

                        residual sugar  chlorides  free sulfur dioxide  \
    fixed acidity             0.122042   0.098989            -0.145343
    volatile acidity         -0.007040   0.070961            -0.016777
    citric acid               0.173474   0.214821            -0.056900
    residual sugar            1.000000   0.063200             0.149018
    chlorides                 0.063200   1.000000            -0.002880
    free sulfur dioxide       0.149018  -0.002880             1.000000
    total sulfur dioxide      0.165529   0.036779             0.668618
    density                   0.374609   0.190870            -0.027318
    pH                       -0.093322  -0.260993             0.054144
    sulphates                 0.011955   0.360577             0.049928
    ...
    density              -0.472967
    pH                    0.197447
    sulphates             0.112979
    alcohol               1.000000
```

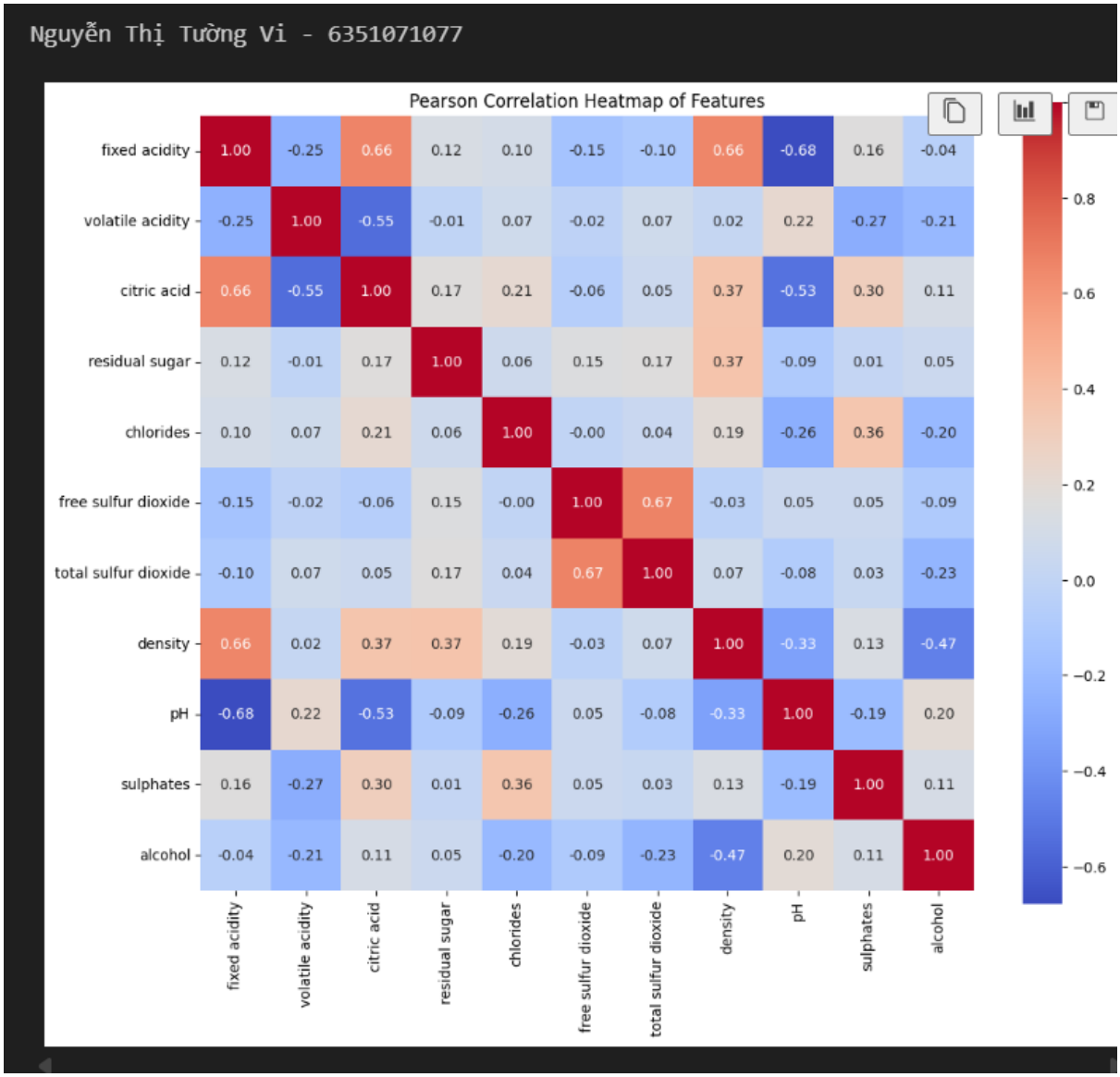
print('Nguyễn Thị Tường Vi - 6351071077')

plt.figure(figsize=(12, 10))

sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', square=True)

plt.title("Pearson Correlation Heatmap of Features")

plt.show()



Câu 4:

```
print('Nguyễn Thị Tường Vi')
features = data.drop('quality', axis=1)
labels = data['quality']

14]  ✓ 0.0s

Nguyễn Thị Tường Vi
```

Câu 5:

```
print('Nguyễn Thị Tường Vi - 6351071077')
features.select_dtypes(exclude=['int64']).columns
```

[15] ✓ 0.0s Python

```
Nguyễn Thị Tường Vi - 6351071077

Index(['fixed acidity', 'volatile acidity', 'citric acid', 'residual sugar',
       'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density',
       'pH', 'sulphates', 'alcohol'],
      dtype='object')
```

print('Nguyễn_Thị_Tường_Vi_6351071077')

features_onehot = pd.get_dummies(features,
columns=features.select_dtypes(exclude=['int64']).columns)

features_onehot

Nguyễn_Thị_Tường_Vi_6351071077

	fixed acidity_4.6	fixed acidity_4.7	fixed acidity_4.9	fixed acidity_5.0	fixed acidity_5.1	fixed acidity_5.2	fixed acidity_5.3	fixed acidity_5.4	fixed acidity_5.5	fixed acidity_5.6	...	alcohol_13.0	alcohol_13.1	alcohol_13.2	alcohol_13.3	alcohol_13.4
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1594	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
1595	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
1596	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
1597	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False
1598	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False

1599 rows × 1453 columns

Câu 6:

```
X_train = features_onehot[: 1119]
X_test = features_onehot[480:]
y_train = labels [:1119]
y_test = labels [480:]
```

✓ 0.0s

Câu 7:

```
print('Nguyễn Thị Tường Vi - 6351071077')

clf = tree.DecisionTreeClassifier(criterion="entropy", random_state=0)
clf.fit(X_train, y_train)
```

✓ 0.1s Python

```
Nguyễn Thị Tường Vi - 6351071077
```

DecisionTreeClassifier

```
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

from sklearn import metrics

print('Nguyễn Thị Tường Vi - 6351071077')

tree_pred = clf.predict(X_test)

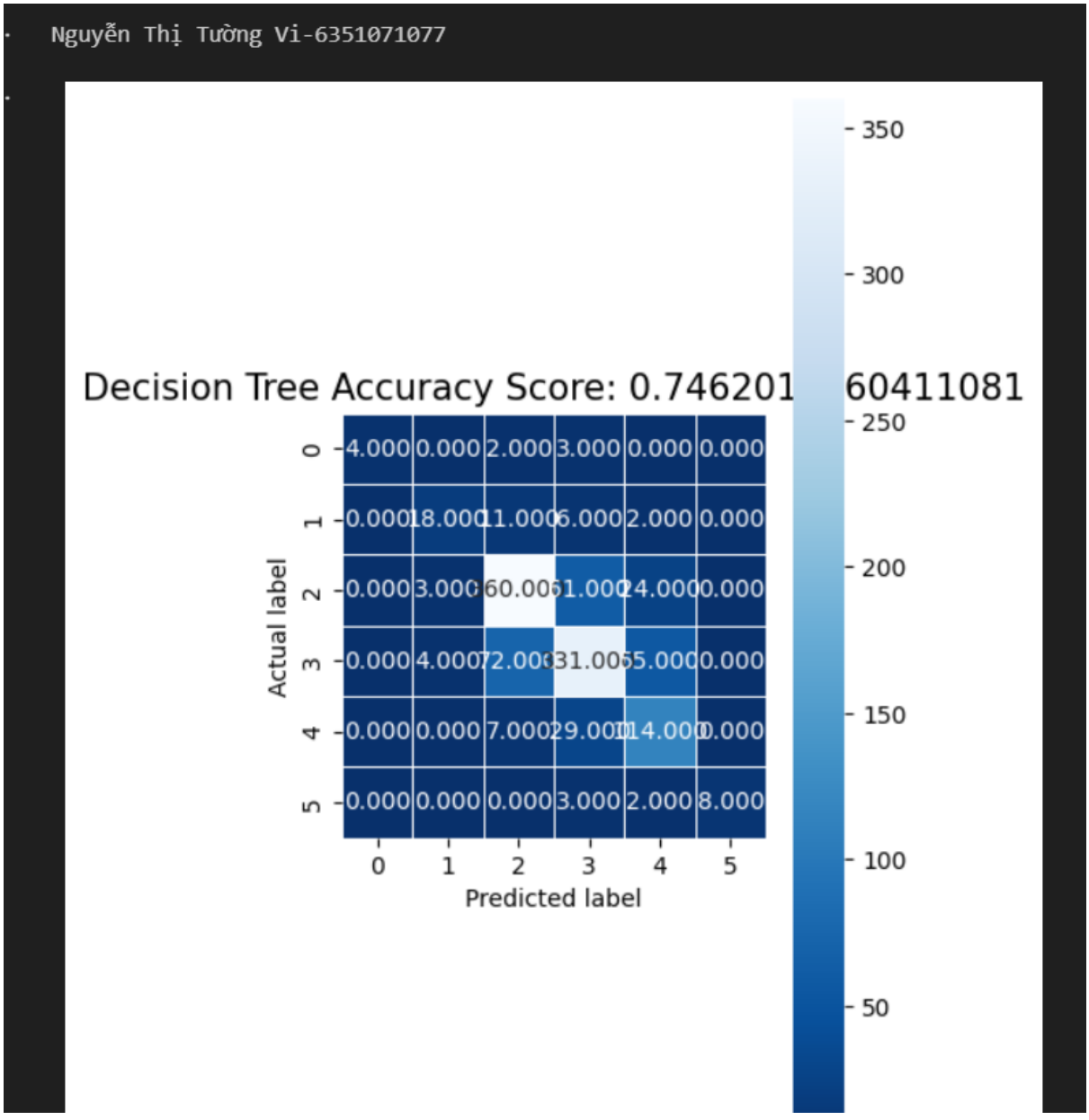
tree_score = metrics.accuracy_score(y_test, tree_pred)

print("Accuracy:", tree_score)

print("Report:", metrics.classification_report(y_test, tree_pred))

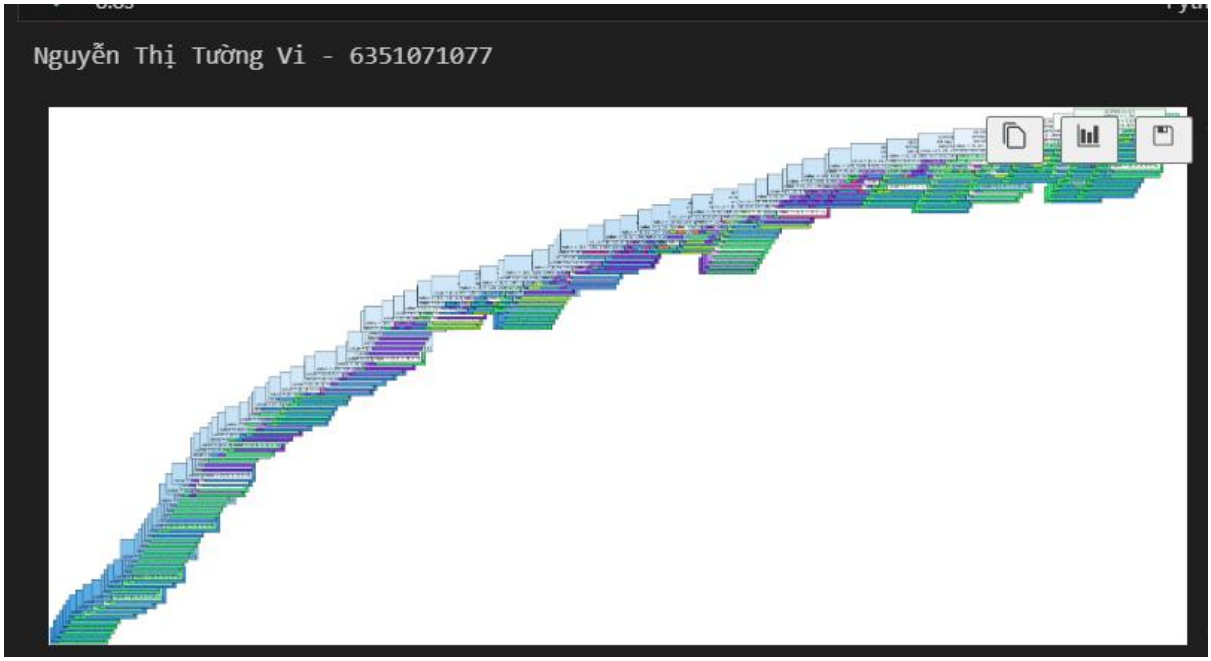
```
Nguyễn Thị Tường Vi - 6351071077
Accuracy: 0.7462019660411081
Report:               precision    recall  f1-score   support

           3       1.00      0.44      0.62         9
           4       0.72      0.49      0.58        37
           5       0.80      0.80      0.80       448
           6       0.76      0.72      0.74       462
           7       0.58      0.76      0.66       150
           8       1.00      0.62      0.76        13

    accuracy                           0.75      1119
   macro avg       0.81      0.64      0.69      1119
weighted avg       0.76      0.75      0.75      1119
```

```
print('Nguyễn Thị Tường Vi-6351071077')
plt.figure(figsize = (4,8))
sns.heatmap(tree_cm, annot = True, fmt=".3f", linewidth =.5, square = True, cmap='Blues_r');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
title = 'Decision Tree Accuracy Score: {0}'.format(tree_score)
plt.title(title, size = 15);
```



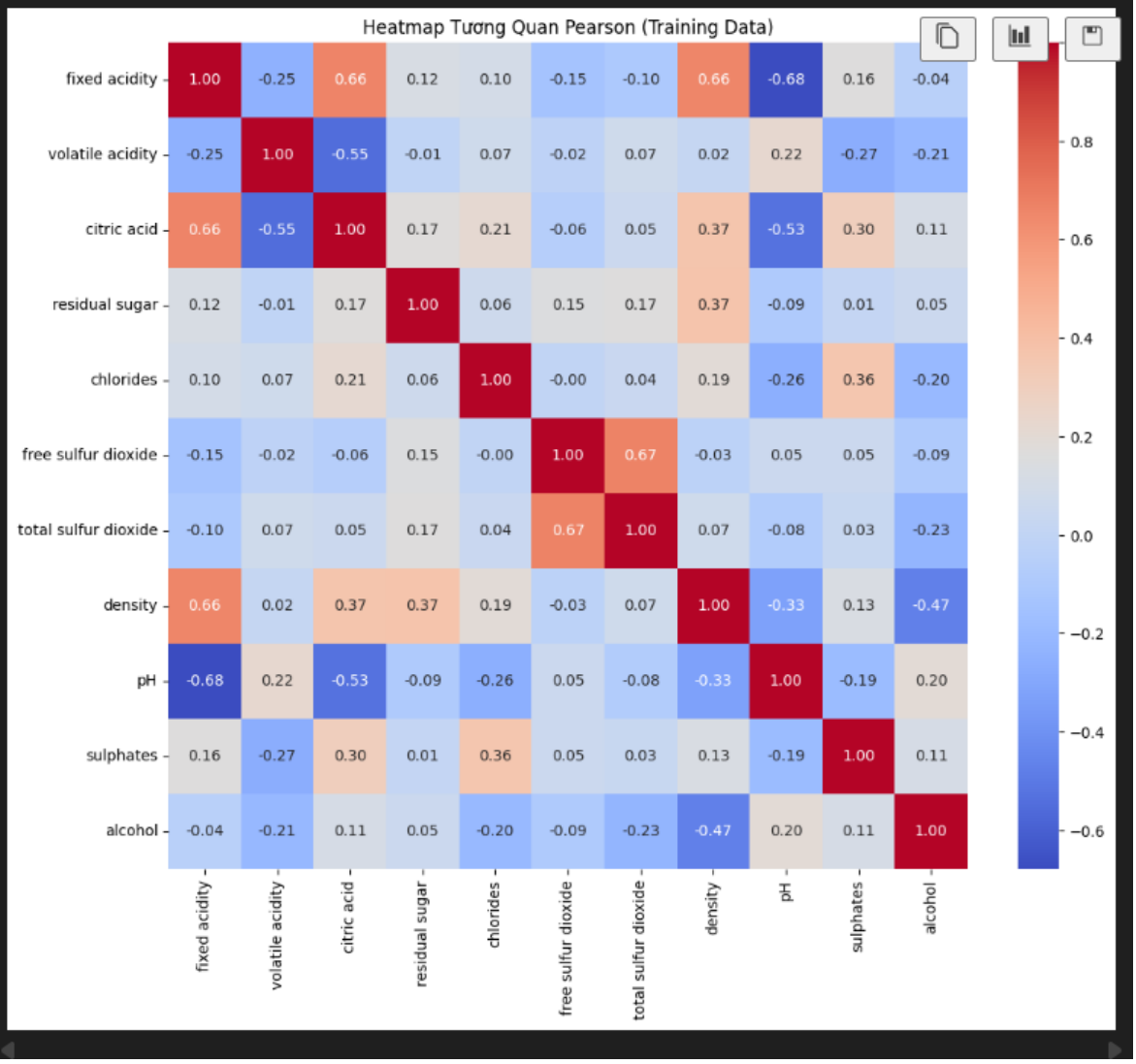

```
print('Nguyễn Thị Tường Vi - 6351071077')
fig, ax = plt.subplots(figsize=(50,24))
```

```
tree.plot_tree(clf, filled=True, fontsize = 10)
plt.savefig('decision_tree', dpi=100)
plt.show()
```

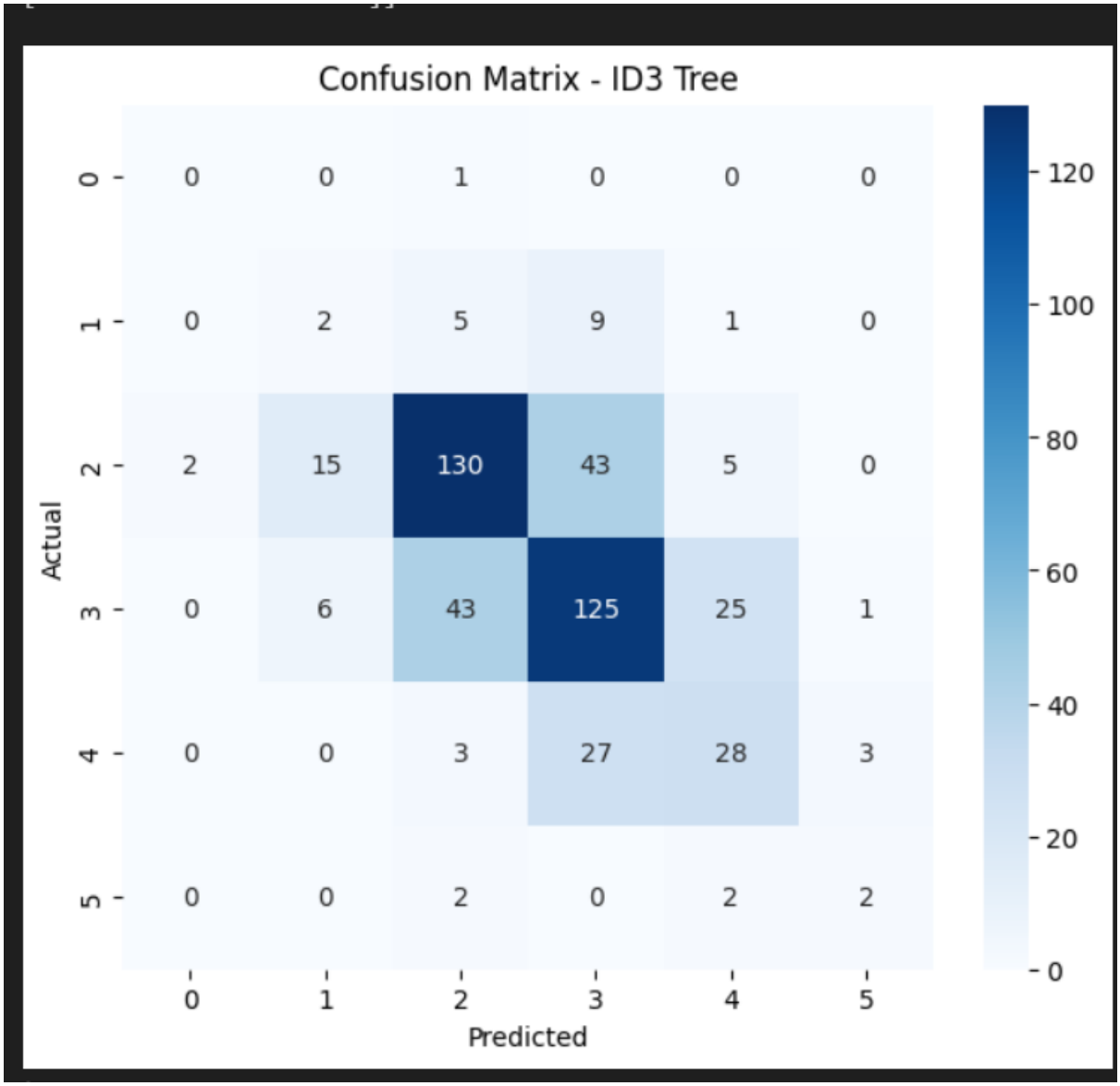


Câu 8:

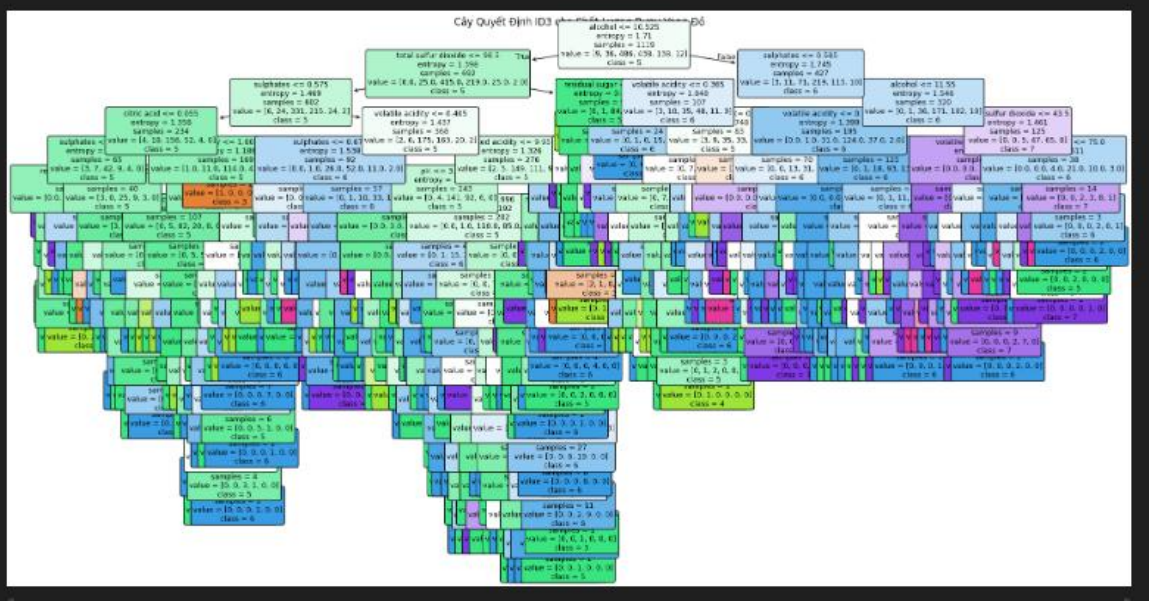
```
Số mẫu train: 1119
Số mẫu test : 480
```




```
Confusion Matrix:
 [[  0   0   1   0   0   0]
 [  0   2   5   9   1   0]
 [  2  15 130  43   5   0]
 [  0   6  43 125  25   1]
 [  0   0   3  27  28   3]
 [  0   0   2   0   2   2]]
```


Confusion Matrix - ID3 Tree
Actual
Predicted
0 0 0 1 0 0 0
1 0 2 5 9 1 0
2 2 15 130 43 5 0
3 0 6 43 125 25 1
4 0 0 3 27 28 3
5 0 0 2 0 2 2
0 1 2 3 4 5
120
100
80
60
40
20
0

```
Classification Report:
              precision    recall  f1-score   support

           3       0.00      0.00      0.00         1
           4       0.09      0.12      0.10        17
           5       0.71      0.67      0.69       195
           6       0.61      0.62      0.62       200
           7       0.46      0.46      0.46        61
           8       0.33      0.33      0.33         6

    accuracy                           0.60       480
   macro avg       0.37      0.37      0.37       480
weighted avg       0.61      0.60      0.60       480
```



Câu 9:

gnb = GaussianNB()

bayes_pred = gnb.fit(X_train, y_train).predict(X_test)

print('Nguyễn Thị Tường Vi-6351071077')

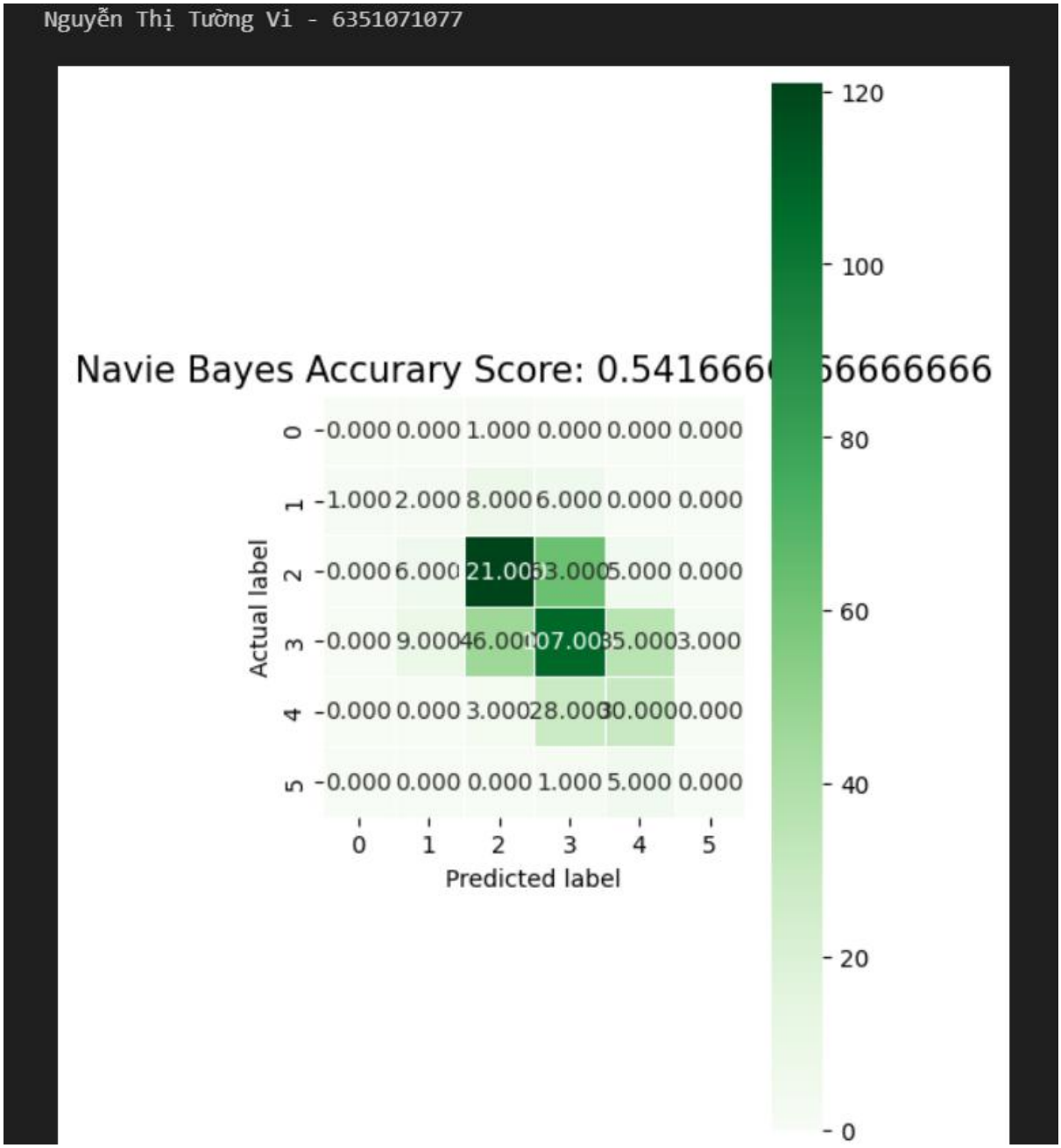
bayes_score = metrics.accuracy_score(y_test, bayes_pred)

print("Accuracy:", bayes_score)

print("Report:", metrics.classification_report(y_test, bayes_pred))

```
Nguyễn Thị Tường Vi-6351071077
Accuracy: 0.5416666666666666
Report:               precision    recall  f1-score   support

           3       0.00      0.00      0.00         1
           4       0.12      0.12      0.12        17
           5       0.68      0.62      0.65       195
           6       0.52      0.54      0.53       200
           7       0.40      0.49      0.44        61
           8       0.00      0.00      0.00         6

    accuracy                           0.54       480
   macro avg       0.29      0.29      0.29       480
weighted avg       0.55      0.54      0.54       480
```

```
bayes_cm = metrics.confusion_matrix(y_test, bayes_pred)

print('Nguyễn Thị Tường Vi - 6351071077')

plt.figure(figsize=(4,8))

sns.heatmap(bayes_cm, annot=True, fmt=".3f", linewidths = .5, square = True, cmap =
'Greens');

plt.ylabel('Actual label');

plt.xlabel('Predicted label');

title = 'Navie Bayes Accurary Score: {0}'.format(bayes_score)

plt.title(title, size = 15);
```



Câu 10:

(tương tự)….

Bài 5 (Trang 26, 27)

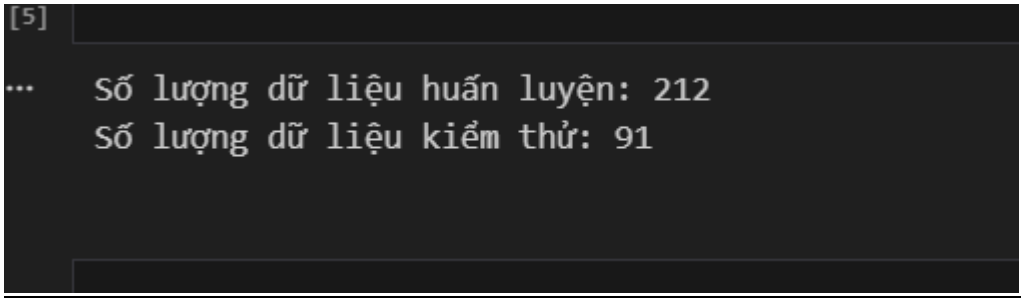
Câu a)

import pandas as pd

import numpy as np

import seaborn as sns

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt
print('Nguyễn Thị Tường Vi - 6351071077')
data = pd.read_csv(r'c:\Users\PC\Downloads\heart.csv')
X = data.drop('target', axis=1)
y = data['target']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)
print("Số lượng dữ liệu huấn luyện:", X_train.shape[0])
print("Số lượng dữ liệu kiểm thử:", X_test.shape[0])
```



Câu 3)

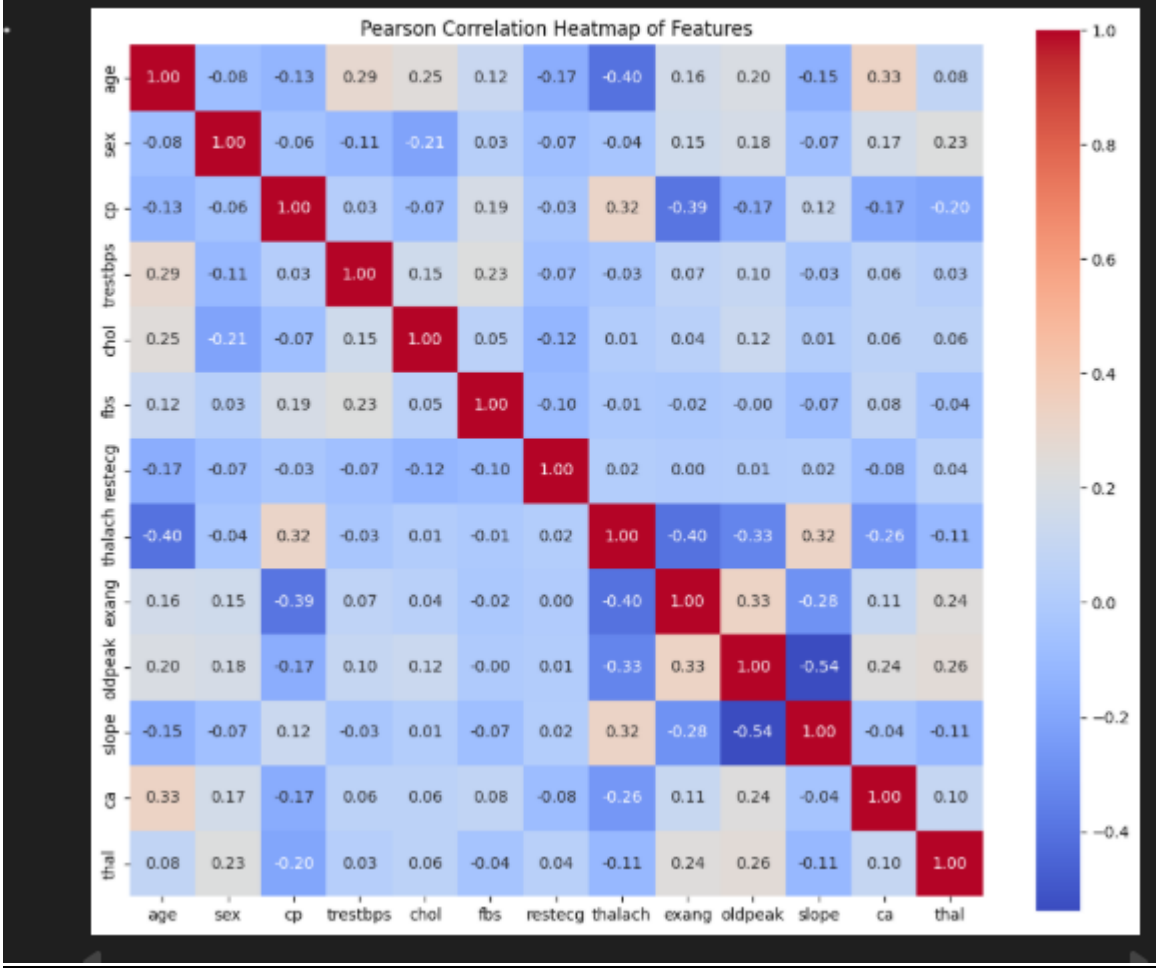
```
print('Nguyễn Thị Tường Vi - 6351071077')
correlation_matrix = X_train.corr(method='pearson')
print(correlation_matrix)
```

```
Nguyễn Thị Tường Vi - 6351071077
                age       sex        cp  trestbps      chol       fbs  \
age        1.000000 -0.076955 -0.127758  0.288078  0.247557  0.121209
sex       -0.076955  1.000000 -0.057577 -0.110884 -0.212008  0.034951
cp        -0.127758 -0.057577  1.000000  0.025663 -0.072715  0.186225
trestbps   0.288078 -0.110884  0.025663  1.000000  0.146431  0.226035
chol       0.247557 -0.212008 -0.072715  0.146431  1.000000  0.051839
fbs        0.121209  0.034951  0.186225  0.226035  0.051839  1.000000
restecg   -0.165419 -0.074695 -0.031449 -0.065109 -0.118689 -0.098113
thalach   -0.404615 -0.038822  0.317051 -0.032537  0.009340 -0.013058
exang      0.161146  0.151385 -0.390310  0.071371  0.039480 -0.020018
oldpeak    0.204483  0.176680 -0.165942  0.103757  0.122876 -0.002365
slope     -0.151790 -0.066913  0.124755 -0.033806  0.014369 -0.074125
ca         0.330450  0.169985 -0.167850  0.058233  0.057201  0.084714
thal       0.081448  0.225652 -0.199712  0.028320  0.060382 -0.042658

            restecg   thalach     exang   oldpeak     slope        ca      thal
age       -0.165419 -0.404615  0.161146  0.204483 -0.151790  0.330450  0.081448
sex       -0.074695 -0.038822  0.151385  0.176680 -0.066913  0.169985  0.225652
cp        -0.031449  0.317051 -0.390310 -0.165942  0.124755 -0.167850 -0.199712
trestbps  -0.065109 -0.032537  0.071371  0.103757 -0.033806  0.058233  0.028320
chol      -0.118689  0.009340  0.039480  0.122876  0.014369  0.057201  0.060382
fbs       -0.098113 -0.013058 -0.020018 -0.002365 -0.074125  0.084714 -0.042658
restecg    1.000000  0.019833  0.002176  0.012953  0.018522 -0.082971  0.038891
thalach    0.019833  1.000000 -0.403342 -0.332174  0.323170 -0.260242 -0.110076
...
oldpeak    0.012953 -0.332174  0.326970  1.000000 -0.539355  0.235034  0.255085
slope      0.018522  0.323170 -0.284329 -0.539355  1.000000 -0.044287 -0.112275
ca        -0.082971 -0.260242  0.110556  0.235034 -0.044287  1.000000  0.098418
thal       0.038891 -0.110076  0.236343  0.255085 -0.112275  0.098418  1.000000
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Câu 4)

```
print('Nguyễn Thị Tường Vi - 6351071077')
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm', square=True)
plt.title("Pearson Correlation Heatmap of Features")
plt.show()
```

```
Nguyễn Thị Tường Vi - 6351071077
```



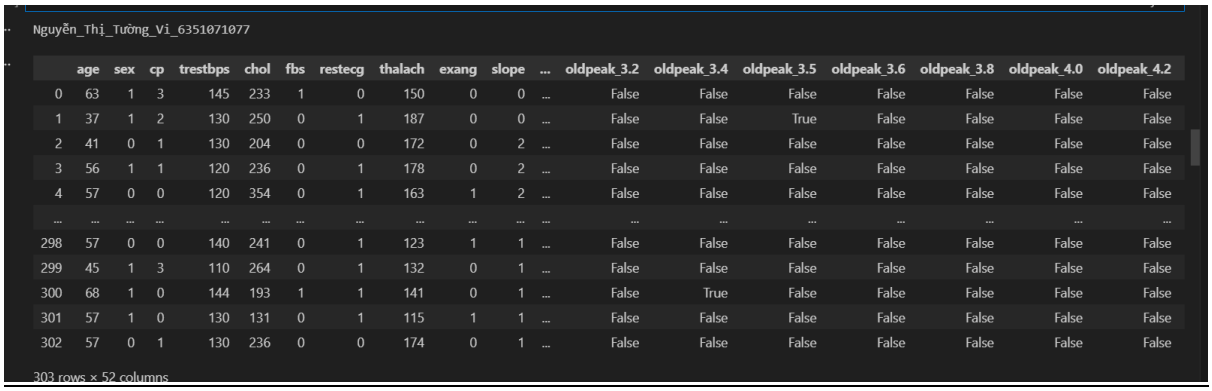

```
print('Nguyễn Thi Tường Vi')

features = data.drop('target', axis=1)

labels = data['target']

print('Nguyễn Thi Tường Vi - 6351071077')

features.select_dtypes(exclude=['int64']).columns
```

```
Nguyễn Thị Tường Vi - 6351071077

Index(['oldpeak'], dtype='object')
```

```
print('Nguyễn_Thi_Tường_Vi_6351071077')
```

features_onehot = pd.get_dummies(features,
columns=features.select_dtypes(exclude=['int64']).columns)

features_onehot



Nguyễn_Thị_Tường_Vi_6351071077

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	slope	...	oldpeak_3.2	oldpeak_3.4	oldpeak_3.5	oldpeak_3.6	oldpeak_3.8	oldpeak_4.0	oldpeak_4.2
0	63	1	3	145	233	1	0	150	0	0	...	False	False	False	False	False	False	False
1	37	1	2	130	250	0	1	187	0	0	...	False	False	True	False	False	False	False
2	41	0	1	130	204	0	0	172	0	2	...	False	False	False	False	False	False	False
3	56	1	1	120	236	0	1	178	0	2	...	False	False	False	False	False	False	False
4	57	0	0	120	354	0	1	163	1	2	...	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	1	...	False	False	False	False	False	False	False
299	45	1	3	110	264	0	1	132	0	1	...	False	False	False	False	False	False	False
300	68	1	0	144	193	1	1	141	0	1	...	False	True	False	False	False	False	False
301	57	1	0	130	131	0	1	115	1	1	...	False	False	False	False	False	False	False
302	57	0	1	130	236	0	0	174	0	1	...	False	False	False	False	False	False	False

303 rows × 52 columns

Câu 6:

X_train = features_onehot[: 212]

X_test = features_onehot[91:]

y_train = labels [:212]

y_test = labels [91:]

Câu 7:

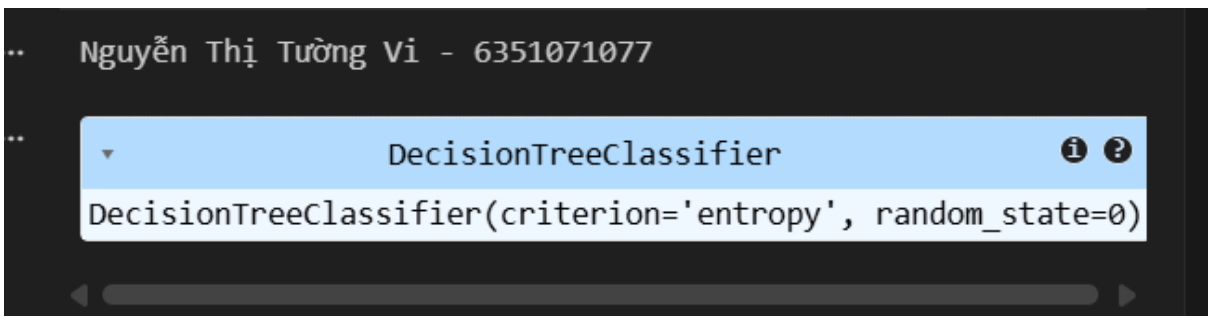
(tương tự)…. from sklearn import tree

from sklearn.tree import DecisionTreeClassifier

print('Nguyễn Thị Tường Vi - 6351071077')

clf = tree.DecisionTreeClassifier(criterion="entropy", random_state=0)

clf.fit(X_train, y_train)



Nguyễn Thị Tường Vi - 6351071077

DecisionTreeClassifier

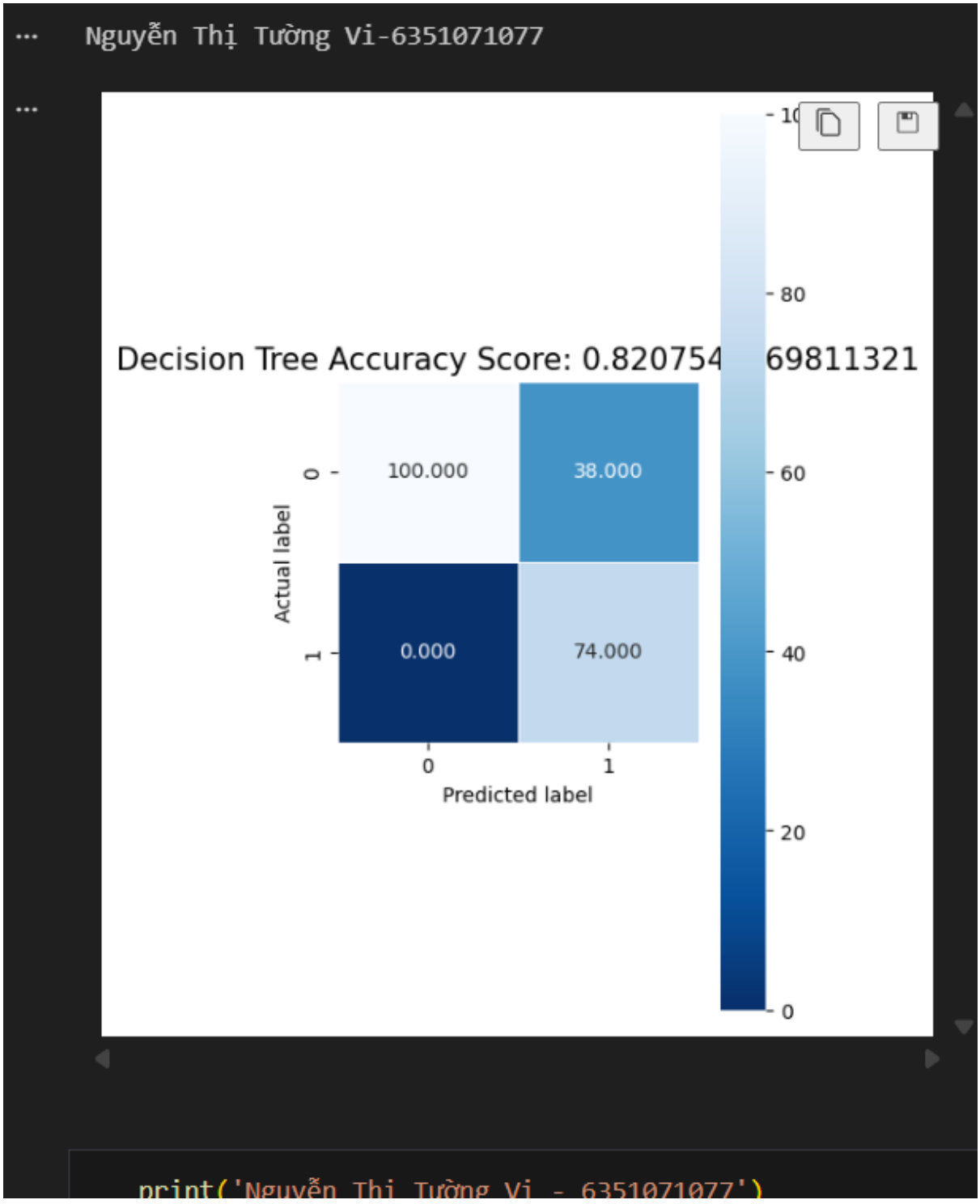
DecisionTreeClassifier(criterion='entropy', random_state=0)

from sklearn import metrics

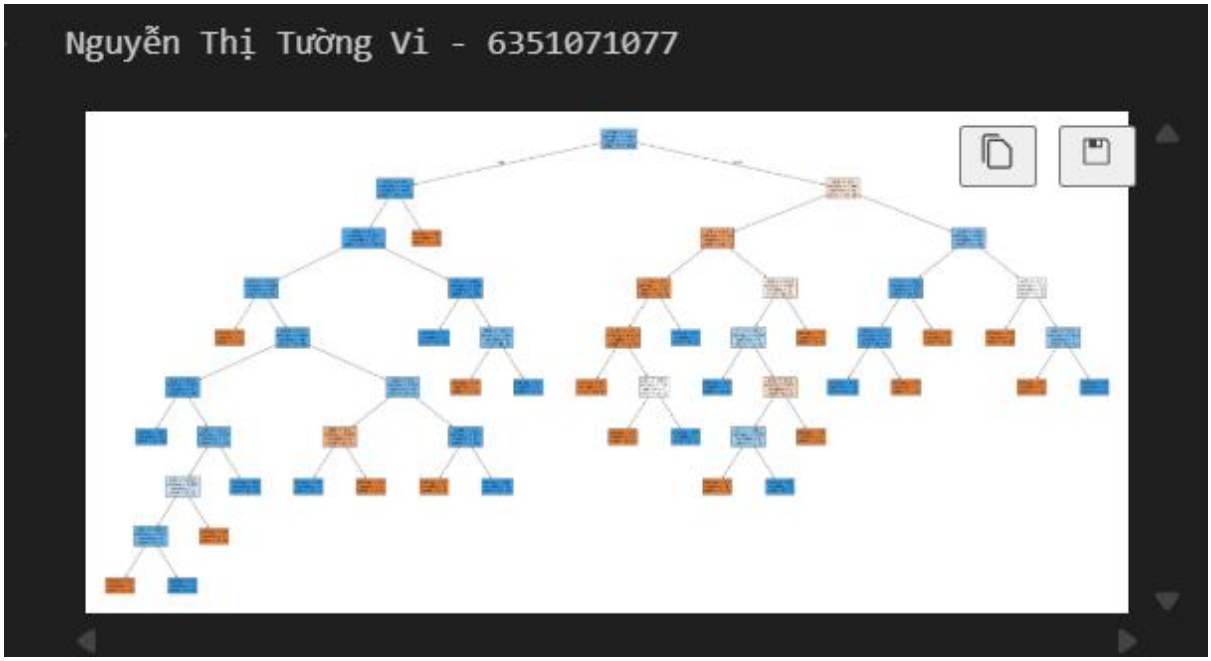
```
print('Nguyễn Thị Tường Vi - 6351071077')
tree_pred = clf.predict(X_test)

tree_score = metrics.accuracy_score(y_test, tree_pred)
print("Accuracy:", tree_score)
print("Report:", metrics.classification_report(y_test, tree_pred))
```

```
Nguyễn Thị Tường Vi - 6351071077
Accuracy: 0.8207547169811321
Report:               precision    recall  f1-score   suppor

           0       1.00      0.72      0.84       138
           1       0.66      1.00      0.80        74

    accuracy                           0.82       212
   macro avg       0.83      0.86      0.82       212
weighted avg       0.88      0.82      0.82       212
```

```
tree_cm = metrics.confusion_matrix(y_test, tree_pred)

print('Nguyễn Thị Tường Vi-6351071077')
plt.figure(figsize = (4,8))
sns.heatmap(tree_cm, annot = True, fmt=".3f", linewidth =.5, square = True, cmap='Blues_r');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
title = 'Decision Tree Accuracy Score: {0}'.format(tree_score)
plt.title(title, size = 15);
```




```
print('Nguyễn Thị Tường Vi - 6351071077')
fig, ax = plt.subplots(figsize=(50,24))
tree.plot_tree(clf, filled=True, fontsize = 10)
plt.savefig('decision_tree', dpi=100)
plt.show()
```



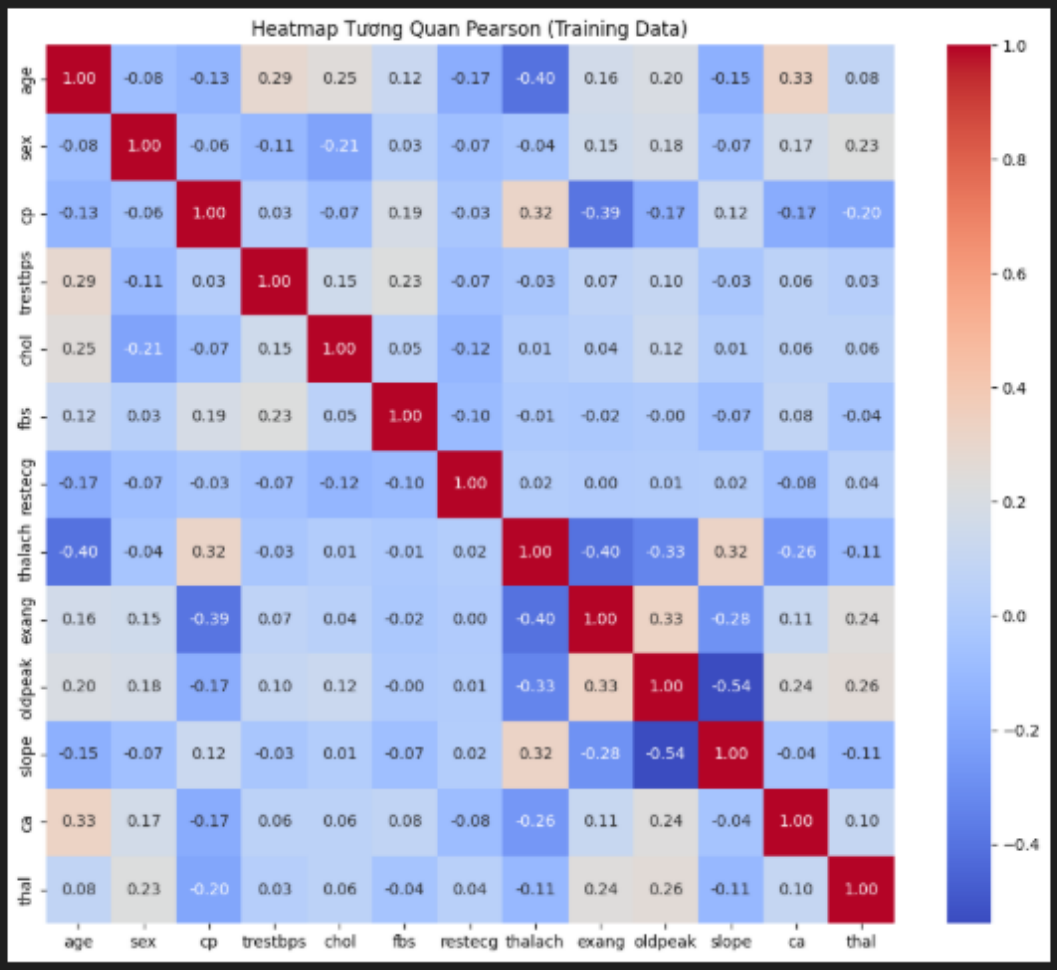
Câu 8:

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=42
)

print("Số mẫu train:", X_train.shape[0])
print("Số mẫu test :", X_test.shape[0])

correlation_matrix = X_train.corr(method="pearson")

plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Heatmap Tương Quan Pearson (Training Data)")
plt.show()

id3_tree = DecisionTreeClassifier(
    criterion="entropy",
```

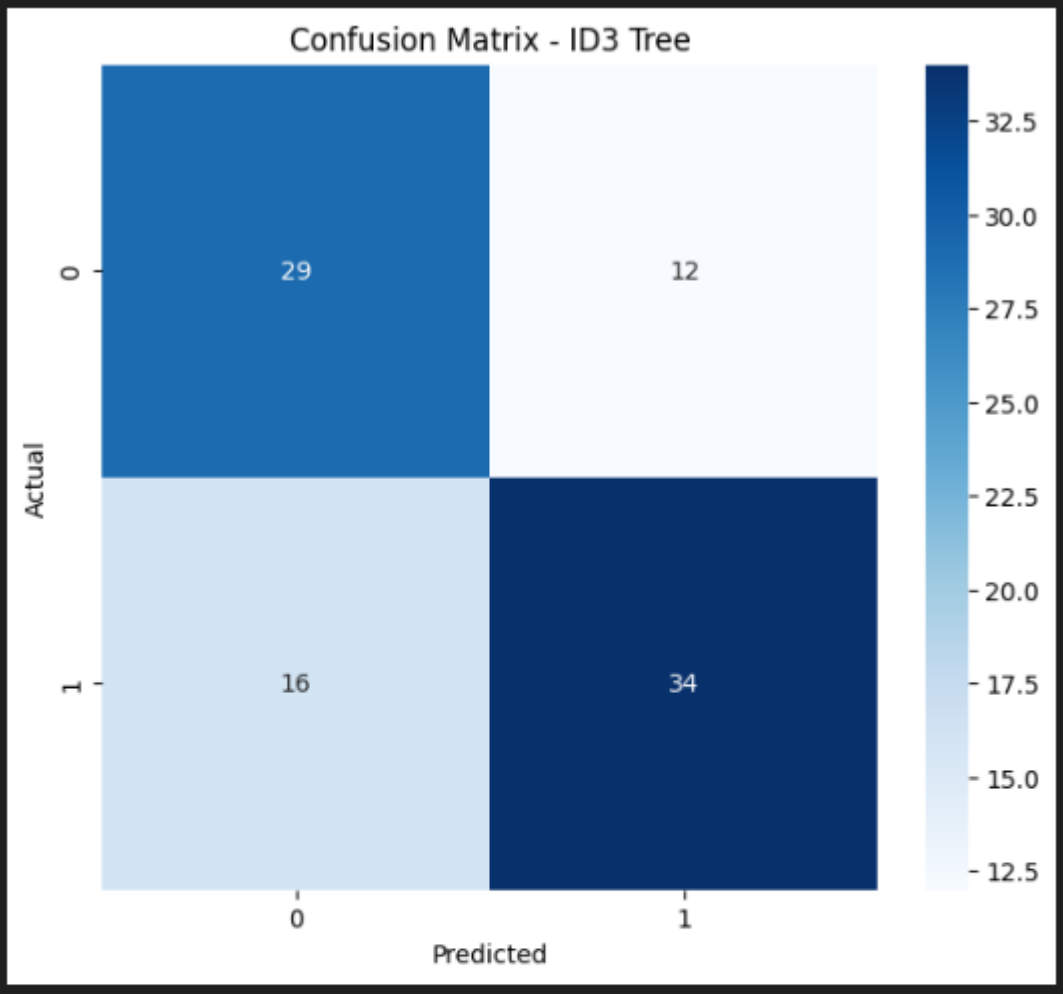
```
    random_state=42
)
id3_tree.fit(X_train, y_train)

y_pred = id3_tree.predict(X_test)

cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", cm)

plt.figure(figsize=(7, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix - ID3 Tree")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

plt.figure(figsize=(22, 12))
plot_tree(
    id3_tree,
    feature_names=X.columns,
    class_names=[str(c) for c in sorted(y.unique())],
    filled=True,
    rounded=True,
    fontsize=9
)
plt.title("Cây Quyết Định ID3 cho Chất Lượng Rượu Vang Đỏ")
```

plt.show()

```
Số mẫu train: 212
Số mẫu test : 91
```

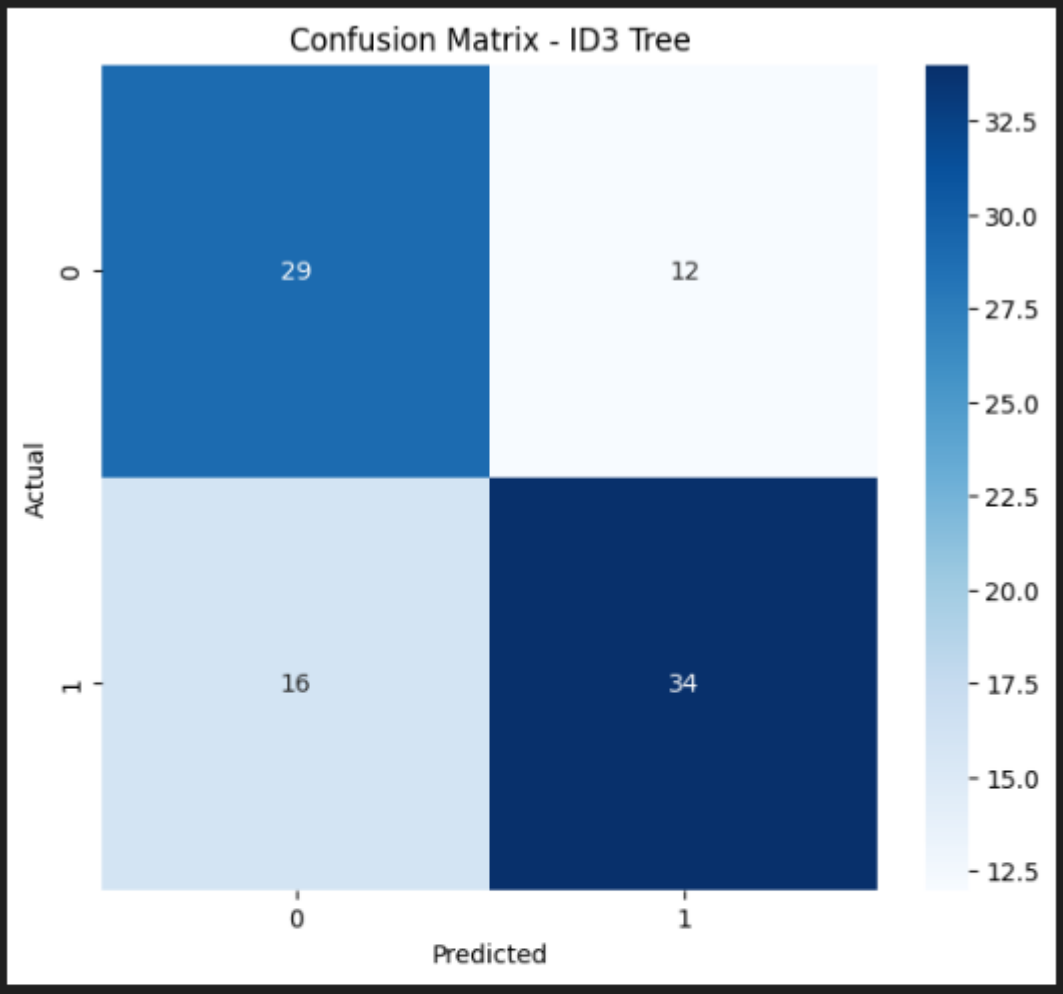



```
Confusion Matrix:
 [[29 12]
 [16 34]]
```



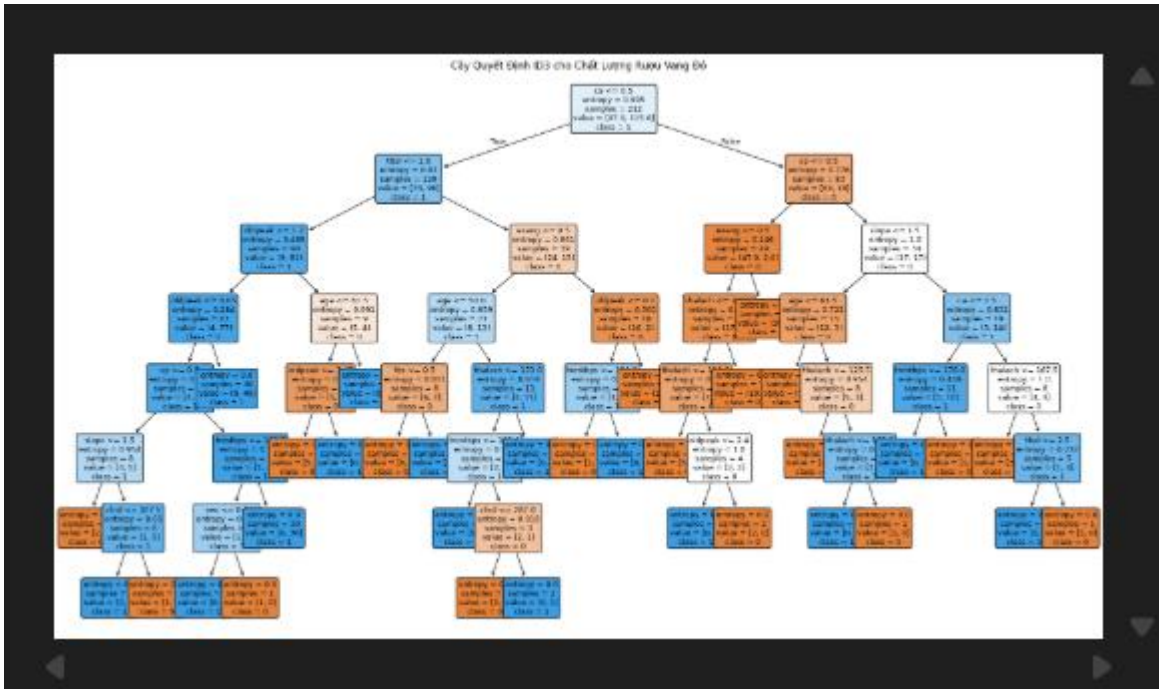
Classification Report:

	precision	recall	f1-score	support
0	0.64	0.71	0.67	41
1	0.74	0.68	0.71	50
accuracy			0.69	91
macro avg	0.69	0.69	0.69	91
weighted avg	0.70	0.69	0.69	91



Classification Report:

	precision	recall	f1-score	support
0	0.64	0.71	0.67	41
1	0.74	0.68	0.71	50
accuracy			0.69	91
macro avg	0.69	0.69	0.69	91
weighted avg	0.70	0.69	0.69	91



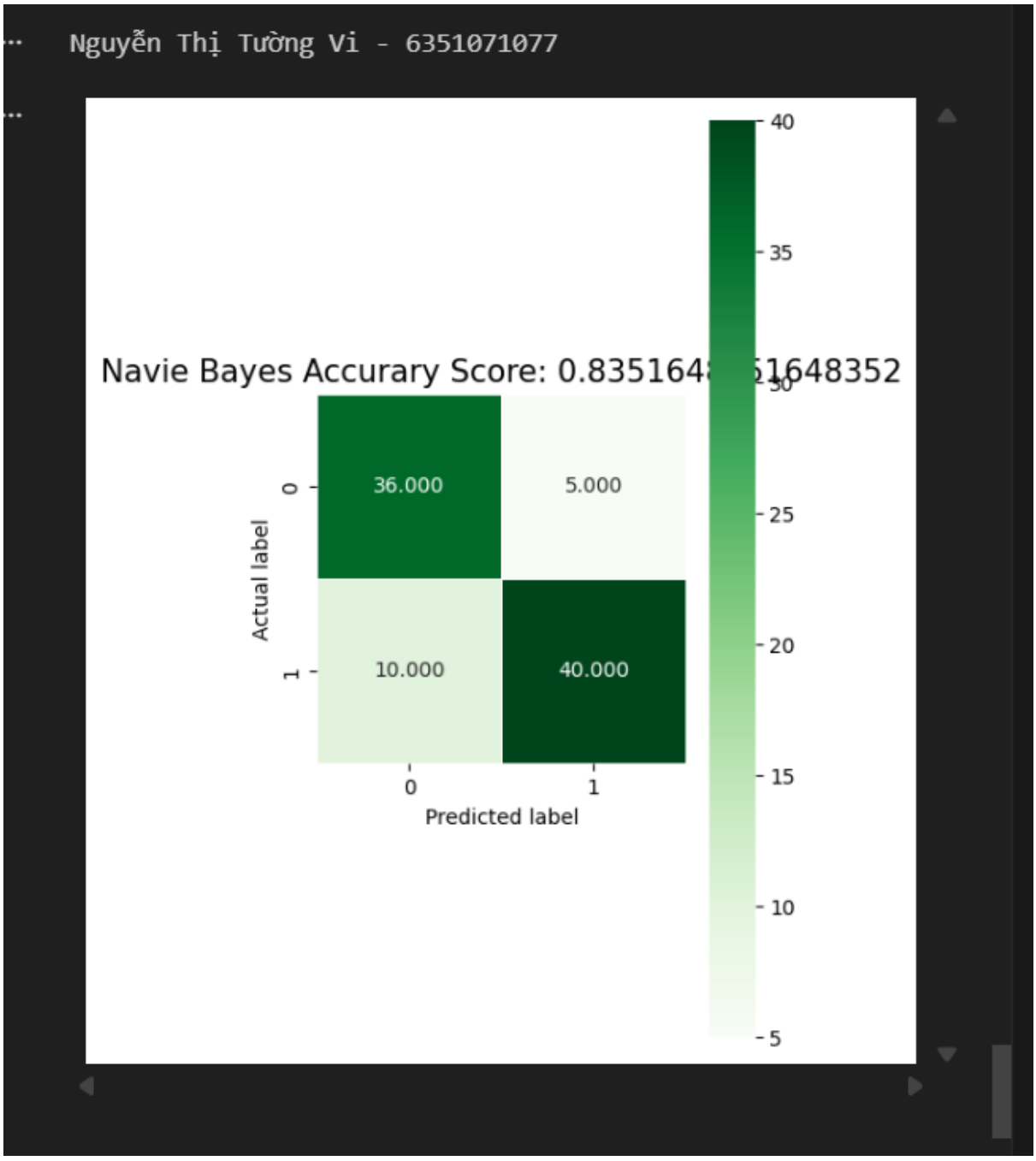
Câu 9:

```
gnb = GaussianNB()
bayes_pred = gnb.fit(X_train, y_train).predict(X_test)
print('Nguyễn Thị Tường Vi-6351071077')
bayes_score = metrics.accuracy_score(y_test, bayes_pred)
print("Accuracy:", bayes_score)
print("Report:", metrics.classification_report(y_test, bayes_pred))
```

```
Nguyễn Thị Tường Vi-6351071077
Accuracy: 0.8351648351648352
Report:               precision    recall  f1-score   suppor

           0       0.78      0.88      0.83        41
           1       0.89      0.80      0.84        50

    accuracy                           0.84        91
   macro avg       0.84      0.84      0.83        91
weighted avg       0.84      0.84      0.84        91
```

```
bayes_cm = metrics.confusion_matrix(y_test, bayes_pred)
print('Nguyễn Thị Tường Vi - 6351071077')
```

```
plt.figure(figsize=(4,8))
sns.heatmap(bayes_cm, annot=True, fmt=".3f", linewidths = .5, square = True, cmap =
'Greens');
plt.ylabel('Actual label');
plt.xlabel('Predicted label');
title = 'Navie Bayes Accurary Score: {0}'.format(bayes_score)
plt.title(title, size = 15);
```



Câu 10:

Câu 10

CART

Tiêu chí	Giá trị
Accuracy Score	0.8026 (80.26%)
True Negative (TN) - Lớp 0 (Đúng 0)	25
True Positive (TP) - Lớp 1 (Đúng 1)	36
False Negative (FN) - Lớp 1 bị nhầm thành 0	5
False Positive (FP) - Lớp 0 bị nhầm thành 1	10

Nhận xét:

* Mô hình CART đạt độ chính xác tốt (80.26%).

* Mô hình có xu hướng dự đoán đúng Lớp 1 (36 TP) tốt hơn so với số lỗi dự đoán Lớp 1 thành Lớp 0 (5 FN).

* Lỗi chính là việc dự đoán sai Lớp 0 thành Lớp 1 (10 FP).

ID3

Tiêu chí	Giá trị
Accuracy Score	0.8026 (80.26%)
True Negative (TN) - Lớp 0 (Đúng 0)	24
True Positive (TP) - Lớp 1 (Đúng 1)	37
False Negative (FN) - Lớp 1 bị nhầm thành 0	4
False Positive (FP) - Lớp 0 bị nhầm thành 1	11

Nhận xét:

* Mô hình ID3 đạt độ chính xác tương đương với CART (80.26%).

* So với CART, ID3 mắc ít lỗi False Negative hơn (4 so với 5), tức là nó ít bỏ sót trường hợp Lớp 1 hơn.

* Tuy nhiên, ID3 lại mắc nhiều lỗi False Positive hơn (11 so với 10), tức là nó dự đoán sai Lớp 0 thành Lớp 1 nhiều hơn.

* Về cơ bản, hiệu suất của CART và ID3 là gần như nhau trên tập dữ liệu này.

****Navie Bayes****

Tiêu chí	Giá trị
Accuracy Score	0.6315 (63.15%)
True Negative (TN) - Lớp 0 (Đúng 0)	24
True Positive (TP) - Lớp 1 (Đúng 1)	24
False Negative (FN) - Lớp 1 bị nhầm thành 0	17
False Positive (FP) - Lớp 0 bị nhầm thành 1	11

****Nhận xét:****

* Mô hình Naive Bayes đạt độ chính xác thấp nhất (63.15%) so với hai mô hình Decision Tree.

* Mô hình này có số lần dự đoán đúng Lớp 0 và Lớp 1 bằng nhau (24 TN và 24 TP), cho thấy sự cân bằng trong dự đoán đúng, nhưng tổng số lỗi lại cao.

* Số lượng False Negative (17 FN) rất cao, cho thấy mô hình này bỏ sót rất nhiều trường hợp thực tế thuộc Lớp 1 (dự đoán sai thành Lớp 0). Đây là lỗi lớn nhất của mô hình này.

Bài 6 (Trang 27)

Câu 4:

```
features = df_clean.drop('class', axis=1)

labels = df_clean['class']
```

Câu 5:

```
print('Nguyễn Thị Tường Vi - 6351071077')

features.select_dtypes(exclude=['int64']).columns
```

```
Nguyễn Thị Tường Vi - 6351071077

Index(['cap-shape', 'cap-surface', 'cap-color', 'bruises', 'odor',
       'gill-attachment', 'gill-spacing', 'gill-size', 'gill-color',
       'stalk-shape', 'stalk-root', 'stalk-surface-above-ring',
       'stalk-surface-below-ring', 'stalk-color-above-ring',
       'stalk-color-below-ring', 'veil-type', 'veil-color', 'ring-number',
       'ring-type', 'spore-print-color', 'population', 'habitat', 'dataset'],
      dtype='object')
```

print('Nguyễn Thị Tường Vi - 6351071077')

features_onebot = pd.get_dummies(features,
columns=features.select_dtypes(exclude=['int64']).columns)

features_onebot

Nguyễn Thị Tường Vi - 6351071077

	cap-shape_b	cap-shape_c	cap-shape_f	cap-shape_k	cap-shape_s	cap-shape_x	cap-surface_f	cap-surface_g	cap-surface_s	cap-surface_y	...	population_v	population_y	habitat_d	habitat_g	habitat_l	habitat_m	habitat_p	habit
0	False	False	False	False	False	True	False	False	True	False	...	False	False	False	False	False	False	False	
1	False	False	False	False	False	True	False	False	True	False	...	False	False	False	True	False	False	False	F
2	True	False	False	False	False	False	False	False	True	False	...	False	False	False	False	False	True	False	F
3	False	False	False	False	False	True	False	False	False	True	...	False	False	False	False	False	False	False	
4	False	False	False	False	False	True	False	False	True	False	...	False	False	False	True	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
8119	False	False	False	True	False	False	False	False	True	False	...	False	False	False	False	True	False	False	F
8120	False	False	False	False	False	True	False	False	True	False	...	True	False	False	False	True	False	False	F
8121	False	False	True	False	False	False	False	False	True	False	...	False	False	False	False	True	False	False	F
8122	False	False	False	True	False	False	False	False	False	True	...	True	False	False	False	True	False	False	F
8123	False	False	False	False	False	True	False	False	True	False	...	False	False	False	False	True	False	False	F

8124 rows × 118 columns

Câu 6:

print('Nguyễn Thị Tường Vi - 6351071077')

X_train, X_test, y_train, y_test = train_test_split(

features_onebot, labels, test_size=0.30, random_state=0, stratify=labels)

print("Train:", len(X_train))

print("Test :", len(X_test))

```
Nguyễn Thị Tường Vi - 6351071077
Train: 5686
Test : 2438
```

Câu 7:

print('Nguyễn Thị Tường Vi - 6351071077')

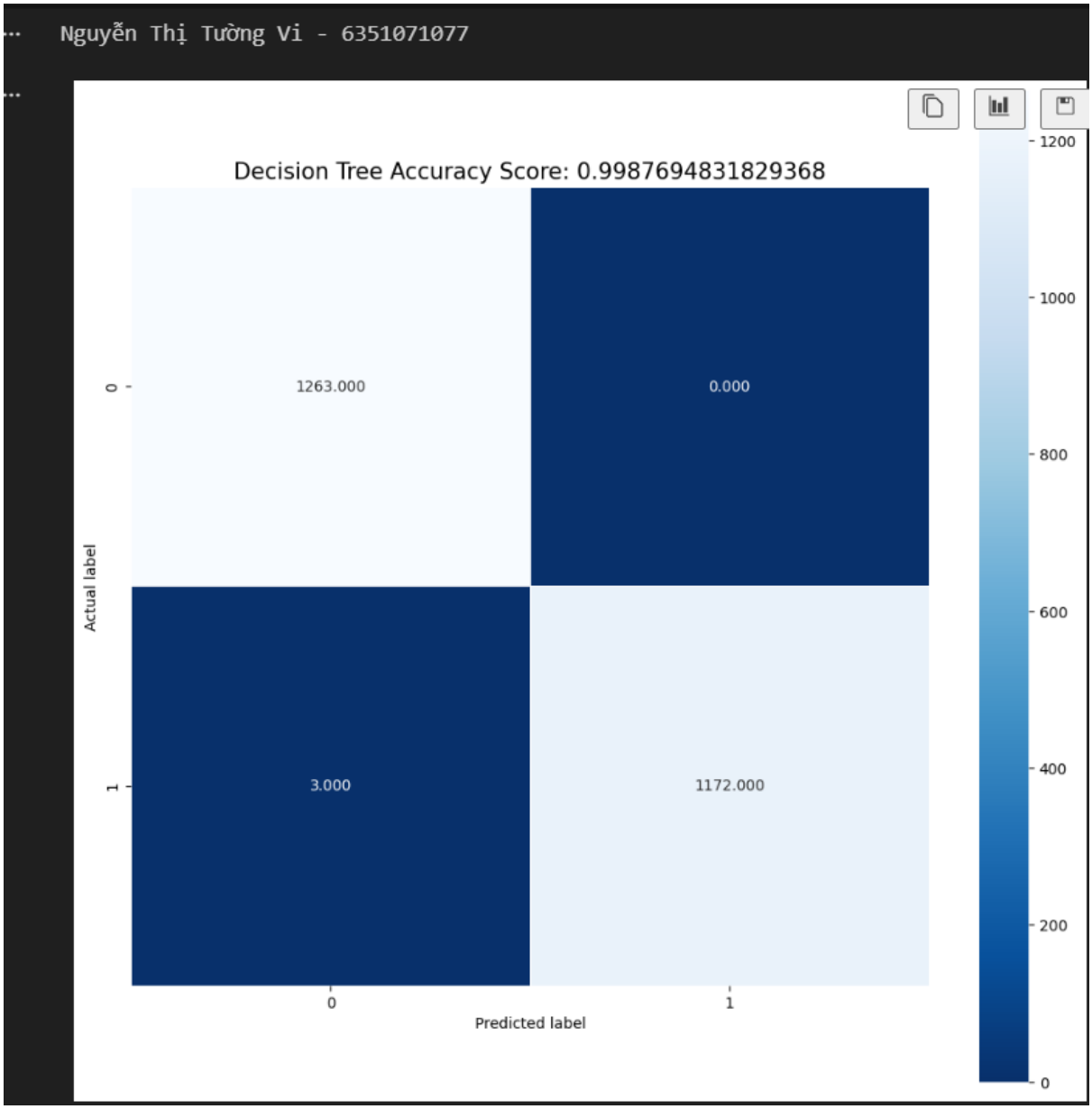
```
clf = tree.DecisionTreeClassifier(criterion="entropy", random_state=0)
clf.fit(X_train, y_train)
```

```
Nguyễn Thị Tường Vi - 6351071077

DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)
```

```
print('Nguyễn Thị Tường Vi - 6351071077')
tree_pred = clf.predict(X_test)
tree_score = metrics.accuracy_score(y_test, tree_pred)
print("Accuracy:", tree_score)
print("Report:", metrics.classification_report(y_test, tree_pred))
```

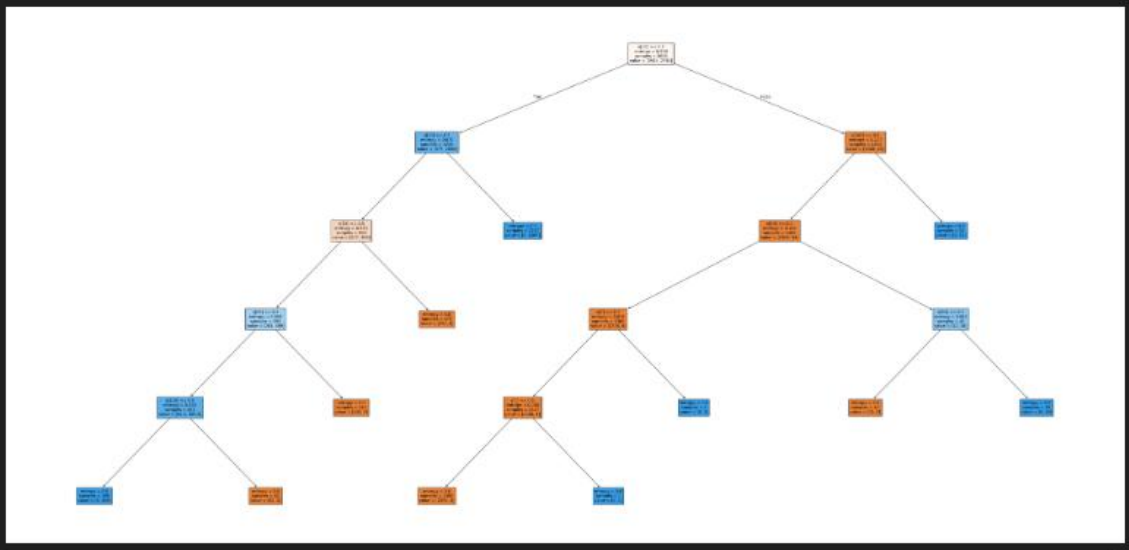
```
Nguyễn Thị Tường Vi - 6351071077
Accuracy: 0.9987694831829368
Report:               precision    recall  f1-score   support

           e       1.00      1.00      1.00      1263
           p       1.00      1.00      1.00      1175

    accuracy                           1.00      2438
   macro avg       1.00      1.00      1.00      2438
weighted avg       1.00      1.00      1.00      2438
```

```
print('Nguyễn Thị Tường Vi - 6351071077')
plt.figure(figsize=(12,12))
sns.heatmap(tree_cm, annot=True, fmt=".3f", linewidths=.5, square=True, cmap='Blues_r')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
plt.title(f'Decision Tree Accuracy Score: {tree_score}', size=15)
plt.show()
```




```
print('Nguyễn Thị Tường Vi - 6351071077')
fig, ax = plt.subplots(figsize=(50, 24))
tree.plot_tree(clf, filled=True, fontsize=10)
plt.savefig('decision_tree', dpi=100)
plt.show()
```

```
Nguyễn Thị Tường Vi - 6351071077
```



Câu 8:

print('Nguyễn Thị Tường Vi - 6351071077')

model_cart = tree.DecisionTreeClassifier(criterion='gini', max_depth=5, random_state=42)

model_cart.fit(X_train, y_train)

y_pred_cart = model_cart.predict(X_test)

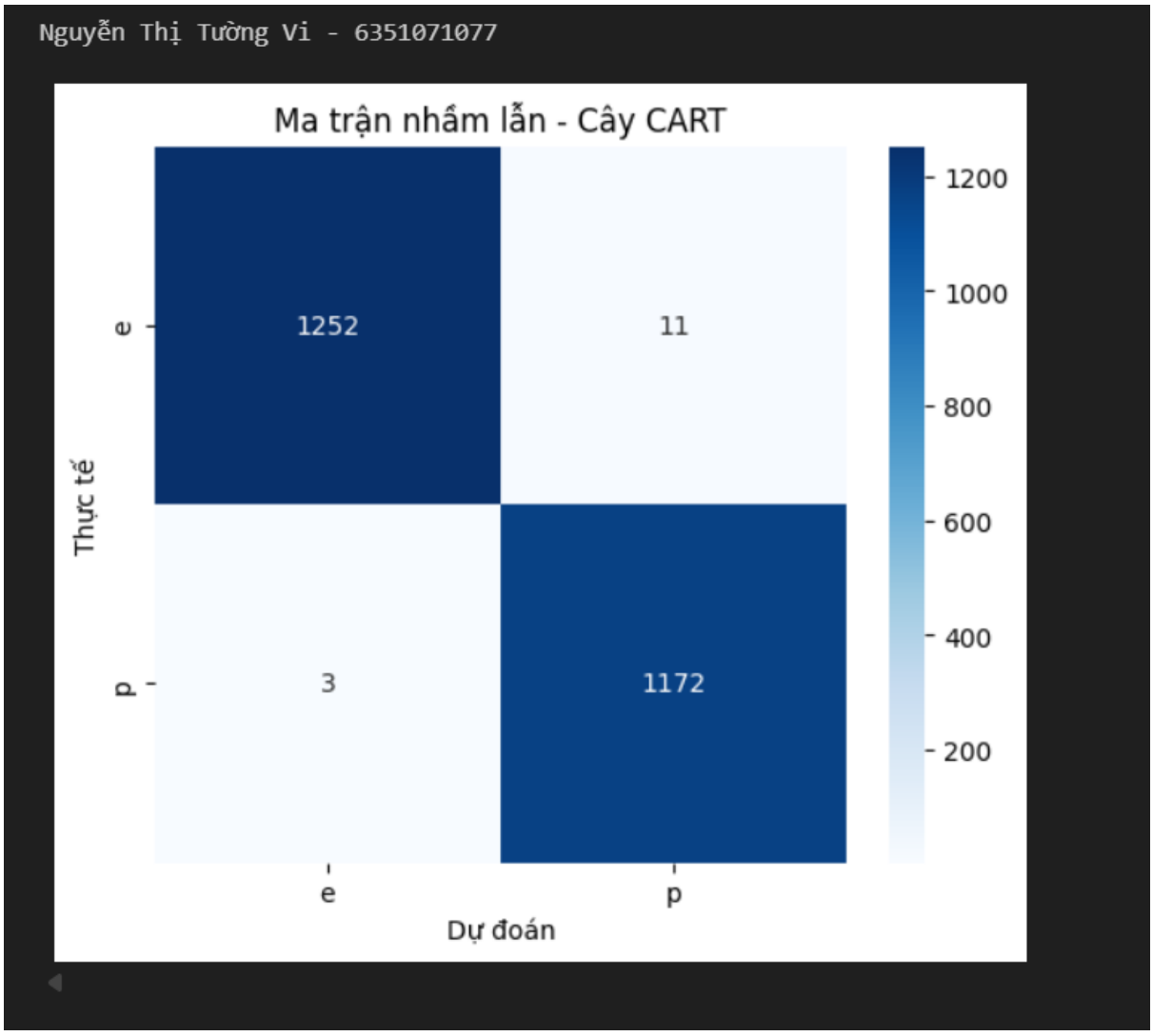
print("--- Kết quả Mô hình Cây CART ---")

print(metrics.classification_report(y_test, y_pred_cart))

```
Nguyễn Thị Tường Vi - 6351071077
--- Kết quả Mô hình Cây CART ---
              precision    recall  f1-score   support

           e       1.00      0.99      0.99      1263
           p       0.99      1.00      0.99      1175

    accuracy                           0.99      2438
   macro avg       0.99      0.99      0.99      2438
weighted avg       0.99      0.99      0.99      2438
```

print('Nguyễn Thị Tường Vi - 6351071077')

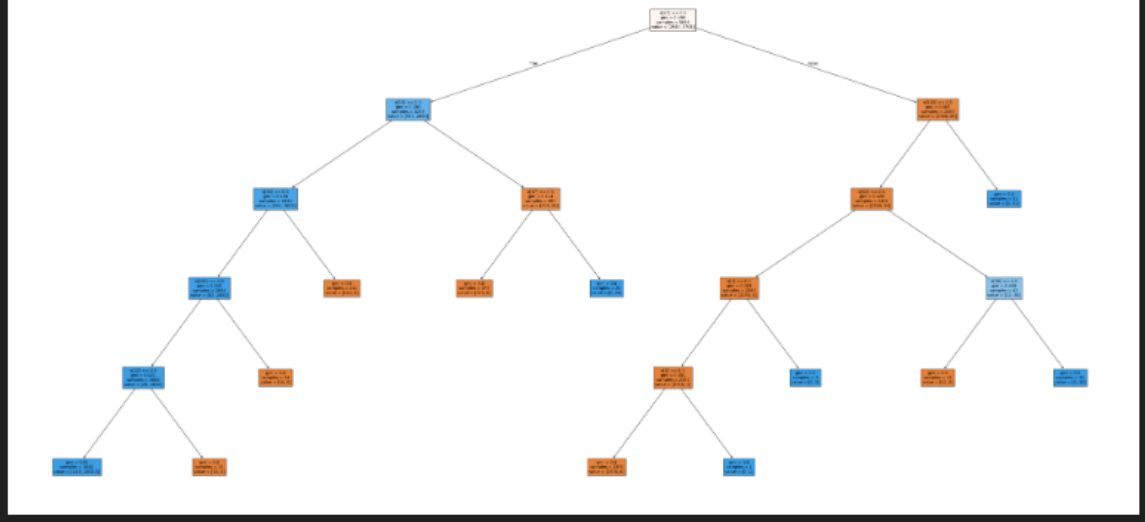
```
cm_cart = metrics.confusion_matrix(y_test, y_pred_cart)
plt.figure(figsize=(6, 5))
sns.heatmap(cm_cart, annot=True, fmt='d', cmap='Blues',
            xticklabels=model_cart.classes_, yticklabels=model_cart.classes_)
plt.title('Ma trận nhầm lẫn - Cây CART')
plt.xlabel('Dự đoán')
plt.ylabel('Thực tế')
plt.show()
```




```
print('Nguyễn Thị Tường Vi - 6351071077')
fig, ax = plt.subplots(figsize=(50, 24))
tree.plot_tree(model_cart, filled=True, fontsize=10)
plt.savefig('decision_tree_cart', dpi=100)
```

plt.show()

```
Nguyễn Thị Tường Vi - 6351071077
```



Câu 9:

gnb = GaussianNB()

bayes_pred = gnb.fit(X_train, y_train).predict(X_test)

print('Nguyễn Thị Tường Vi - 6351071077')

bayes_score = metrics.accuracy_score(y_test, bayes_pred)

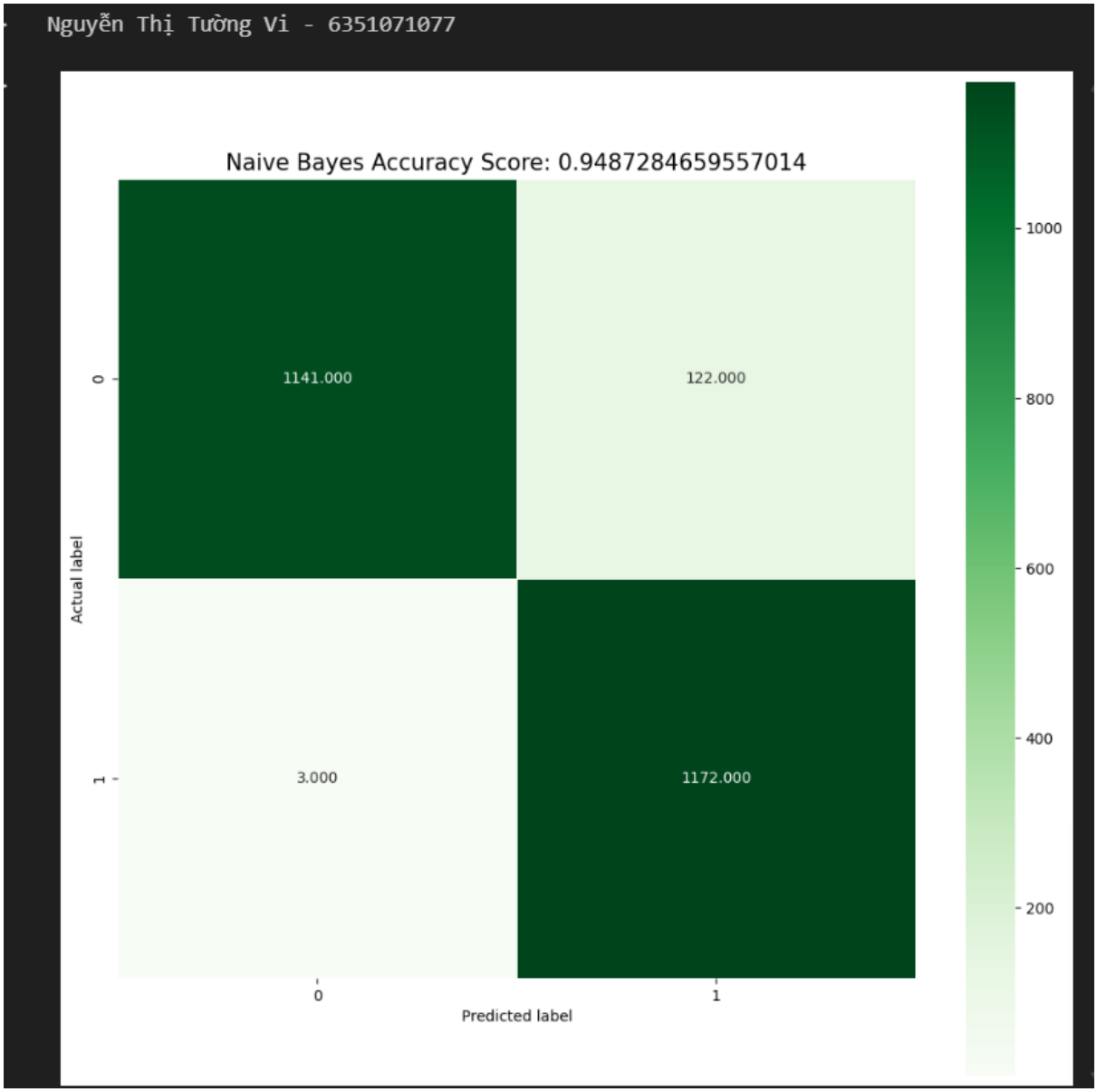
print("Accuracy:", bayes_score)

print("Report:", metrics.classification_report(y_test, bayes_pred))

```
Nguyễn Thị Tường Vi - 6351071077
Accuracy: 0.9487284659557014
Report:               precision    recall  f1-score   support

           e       1.00      0.90      0.95      1263
           p       0.91      1.00      0.95      1175

    accuracy                           0.95      2438
   macro avg       0.95      0.95      0.95      2438
weighted avg       0.95      0.95      0.95      2438
```

print('Nguyễn Thị Tường Vi - 6351071077')

```
bayes_cm = metrics.confusion_matrix(y_test, bayes_pred)
plt.figure(figsize=(12,12))
sns.heatmap(bayes_cm, annot=True, fmt=".3f", linewidths=.5, square=True, cmap='Greens')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
plt.title(f'Naive Bayes Accuracy Score: {bayes_score}', size=15)
plt.show()
```



Câu 10:

****Câu 10****

CART

Tiêu chí	Giá trị
Accuracy Score	0.8026 (80.26%)
True Negative (TN) - Lớp 0 (Đúng 0)	25
True Positive (TP) - Lớp 1 (Đúng 1)	36
False Negative (FN) - Lớp 1 bị nhầm thành 0	5
False Positive (FP) - Lớp 0 bị nhầm thành 1	10

Nhận xét:

* Mô hình CART đạt độ chính xác tốt (80.26%).

* Mô hình có xu hướng dự đoán đúng Lớp 1 (36 TP) tốt hơn so với số lỗi dự đoán Lớp 1 thành Lớp 0 (5 FN).

* Lỗi chính là việc dự đoán sai Lớp 0 thành Lớp 1 (10 FP).

ID3

Tiêu chí	Giá trị
Accuracy Score	0.8026 (80.26%)
True Negative (TN) - Lớp 0 (Đúng 0)	24
True Positive (TP) - Lớp 1 (Đúng 1)	37
False Negative (FN) - Lớp 1 bị nhầm thành 0	4
False Positive (FP) - Lớp 0 bị nhầm thành 1	11

Nhận xét:

* Mô hình ID3 đạt độ chính xác tương đương với CART (80.26%).

* So với CART, ID3 mắc ít lỗi False Negative hơn (4 so với 5), tức là nó ít bỏ sót trường hợp Lớp 1 hơn.

* Tuy nhiên, ID3 lại mắc nhiều lỗi False Positive hơn (11 so với 10), tức là nó dự đoán sai Lớp 0 thành Lớp 1 nhiều hơn.

* Về cơ bản, hiệu suất của CART và ID3 là gần như nhau trên tập dữ liệu này.

****Navie Bayes****

Tiêu chí	Giá trị
Accuracy Score	0.6315 (63.15%)
True Negative (TN) - Lớp 0 (Đúng 0)	24
True Positive (TP) - Lớp 1 (Đúng 1)	24
False Negative (FN) - Lớp 1 bị nhầm thành 0	17
False Positive (FP) - Lớp 0 bị nhầm thành 1	11

****Nhận xét:****

* Mô hình Naive Bayes đạt độ chính xác thấp nhất (63.15%) so với hai mô hình Decision Tree.

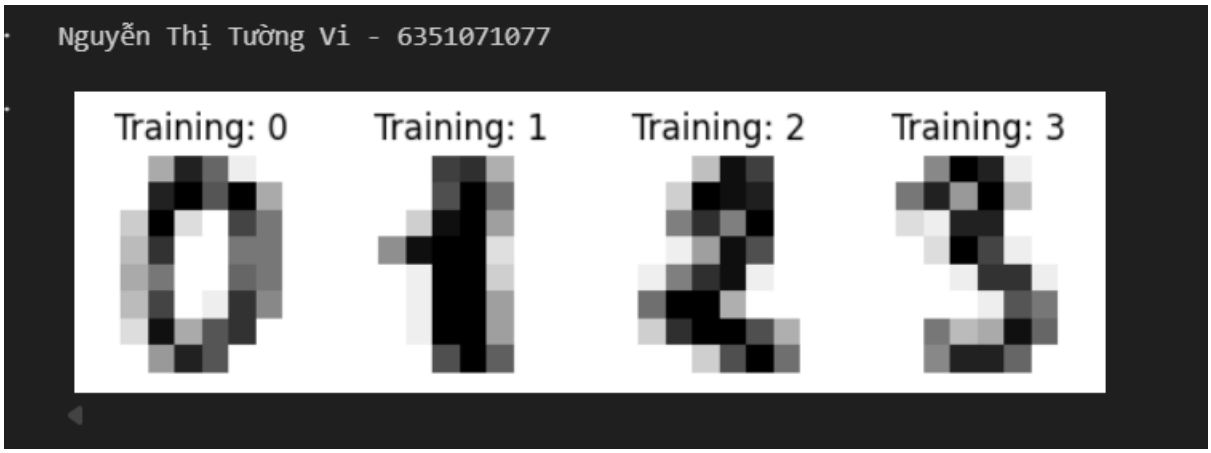
* Mô hình này có số lần dự đoán đúng Lớp 0 và Lớp 1 bằng nhau (24 TN và 24 TP), cho thấy sự cân bằng trong dự đoán đúng, nhưng tổng số lỗi lại cao.

* Số lượng False Negative (17 FN) rất cao, cho thấy mô hình này bỏ sót rất nhiều trường hợp thực tế thuộc Lớp 1 (dự đoán sai thành Lớp 0). Đây là lỗi lớn nhất của mô hình này.

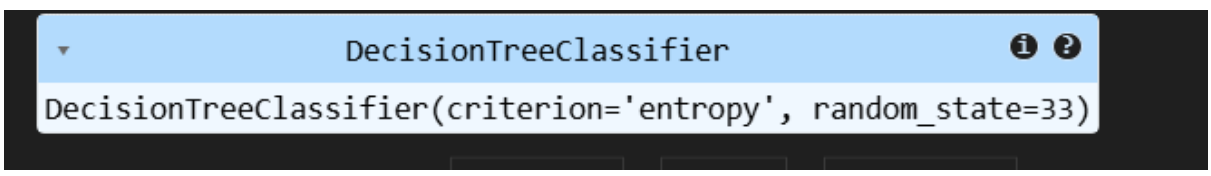
Bài 7 (Trang 27 - 31)

```
%matplotlib inline

import matplotlib

import matplotlib.pyplot as plt

from sklearn import datasets, tree, metrics

from sklearn.model_selection  import train_test_split


digits = datasets.load_digits()
```

```
print('Nguyễn Thị Tường Vi - 6351071077')
_, axes = plt.subplots(1, 4)
images_and_labels = list(zip(digits.images, digits.target))
for ax, (image, label) in zip(axes, images_and_labels[:4]):
    ax.set_axis_off()
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    ax.set_title('Training: %i' % label)
plt.show()
```

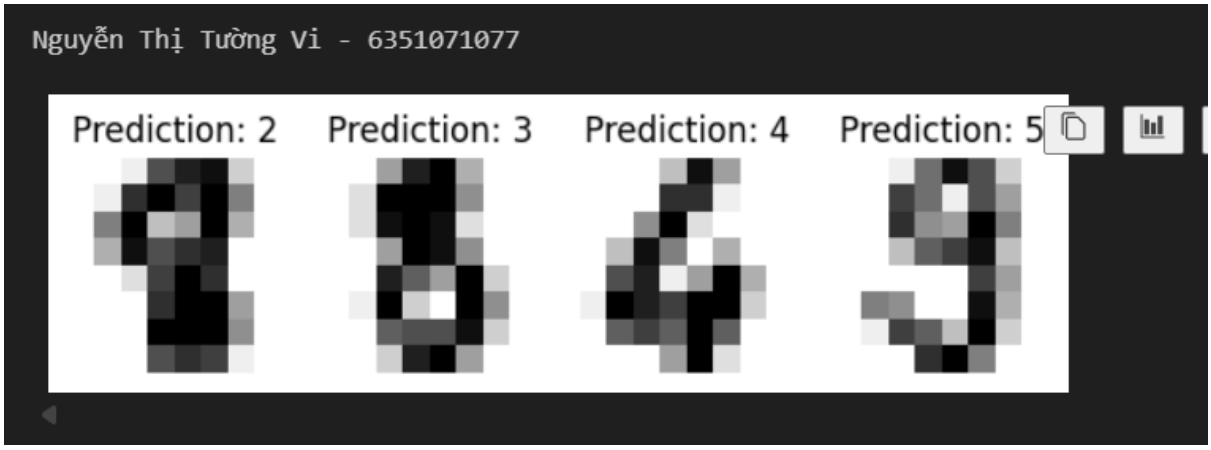



```
n_samples = len(digits.images)
data = digits.images.reshape((n_samples, -1))

classifier = tree.DecisionTreeClassifier(criterion ="entropy" , random_state=33 )

X_train , X_test , y_train , y_test = train_test_split(
    data , digits.target , test_size = 0.2 , shuffle = False
)
classifier.fit(X_train , y_train)
```



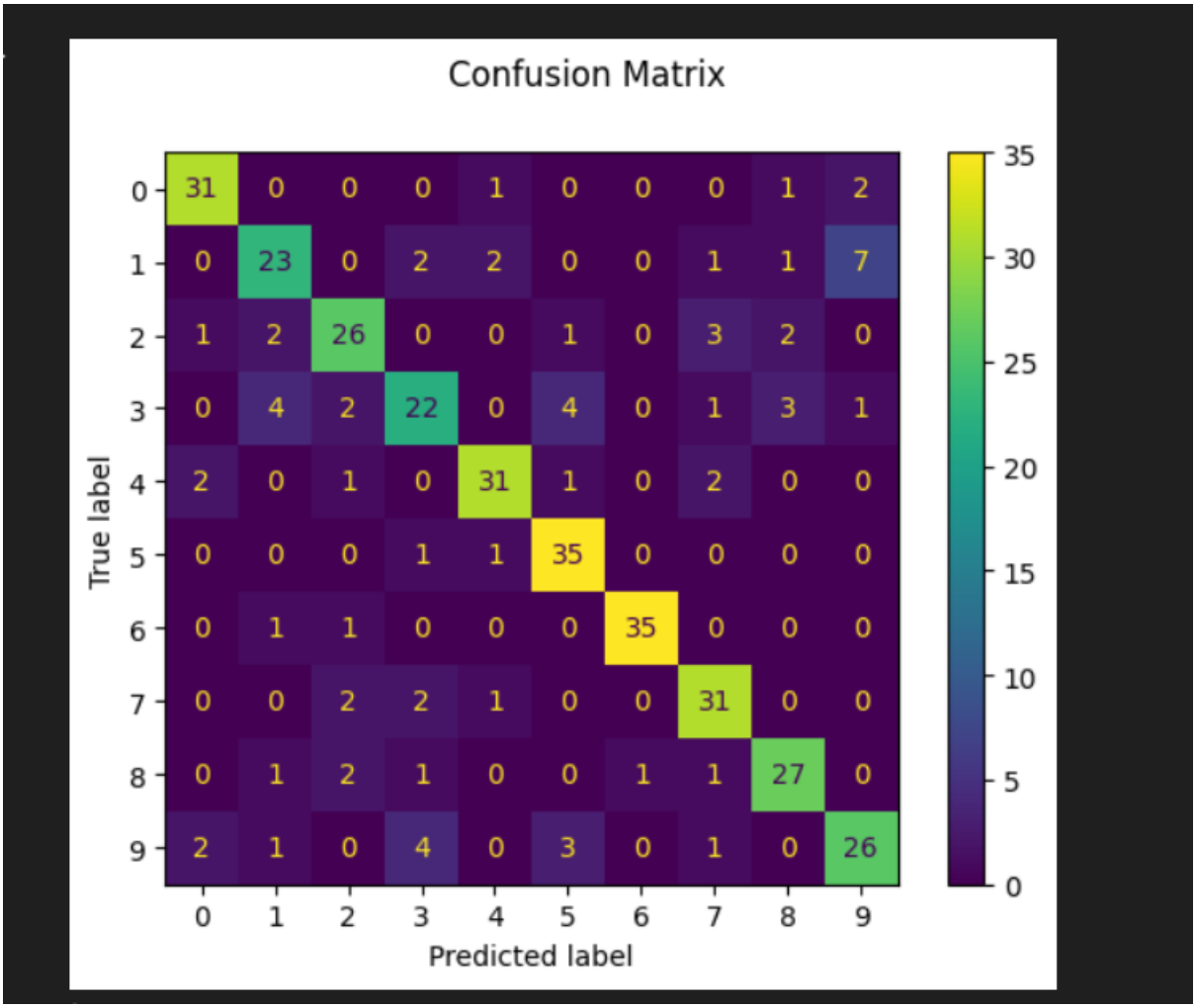

```
predicted = classifier.predict(X_test)
```

```
print('Nguyễn Thị Tường Vi - 6351071077')
_, axes = plt.subplots(1, 4)
images_and_predictions = list(zip(digits.images[n_samples // 2:], predicted))
for ax, (image, prediction) in zip(axes, images_and_predictions[:4]):
    ax.set_axis_off()
    ax.imshow(image, cmap=plt.cm.gray_r, interpolation='nearest')
    ax.set_title('Prediction: %i' % prediction)
plt.show()
```



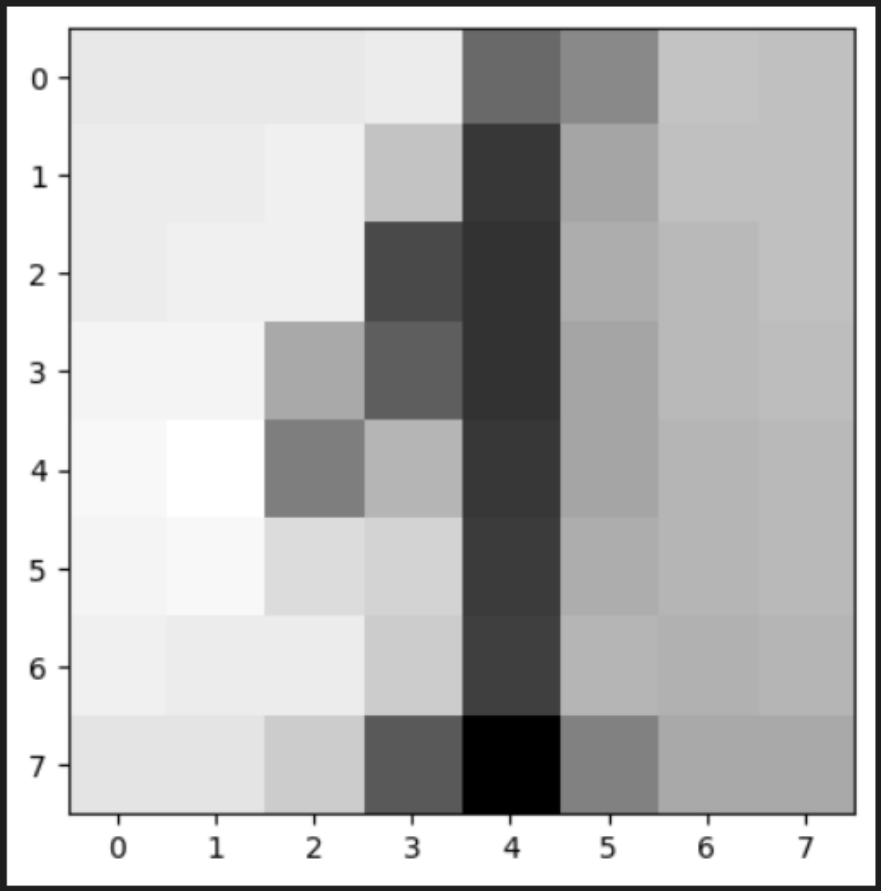

```
print('Nguyễn Thị Tường Vi - 6351071077')
print("Classification report for classifier %s:\n%s\n"
      % (classifier, metrics.classification_report(y_test, predicted)))

disp = metrics.ConfusionMatrixDisplay.from_estimator(classifier, X_test, y_test)
disp.figure_.suptitle("Confusion Matrix")
print('Nguyễn Thị Tường Vi - 6351071077')
print("Confusion matrix:\n%s" % disp.confusion_matrix)
```

```
Nguyễn Thị Tường Vi - 6351071077
Classification report for classifier DecisionTreeClassifier(criterion='entropy', random_state=33):
              precision    recall  f1-score   support

           0       0.86      0.89      0.87        35
           1       0.72      0.64      0.68        36
           2       0.76      0.74      0.75        35
           3       0.69      0.59      0.64        37
           4       0.86      0.84      0.85        37
           5       0.80      0.95      0.86        37
           6       0.97      0.95      0.96        37
           7       0.78      0.86      0.82        36
           8       0.79      0.82      0.81        33
           9       0.72      0.70      0.71        37

    accuracy                           0.80       360
   macro avg       0.80      0.80      0.79       360
weighted avg       0.80      0.80      0.79       360


Nguyễn Thị Tường Vi - 6351071077
Confusion matrix:
[[31  0  0  0  1  0  0  0  1  2]
 [ 0 23  0  2  2  0  0  1  1  7]
 [ 1  2 26  0  0  1  0  3  2  0]
...
 [ 0  1  1  0  0  0 35  0  0  0]
 [ 0  0  2  2  1  0  0 31  0  0]
 [ 0  1  2  1  0  0  1  1 27  0]
 [ 2  1  0  4  0  3  0  1  0 26]]
```




```
from PIL import Image , ImageOps

import numpy as np


print('Nguyễn Thị Tường Vi - 6351071077')

img = Image.open(r'c:\Users\PC\Downloads\test1_image_Bai7
(1).jpg').convert("L").resize((8,8))

img = ImageOps.invert(img)

im2arr = np.array(img)

plt.imshow(im2arr, cmap=plt.cm.gray_r, interpolation='nearest')
```

```
Nguyễn Thị Tường Vi - 6351071077

<matplotlib.image.AxesImage at 0x1c0662abed0>
```



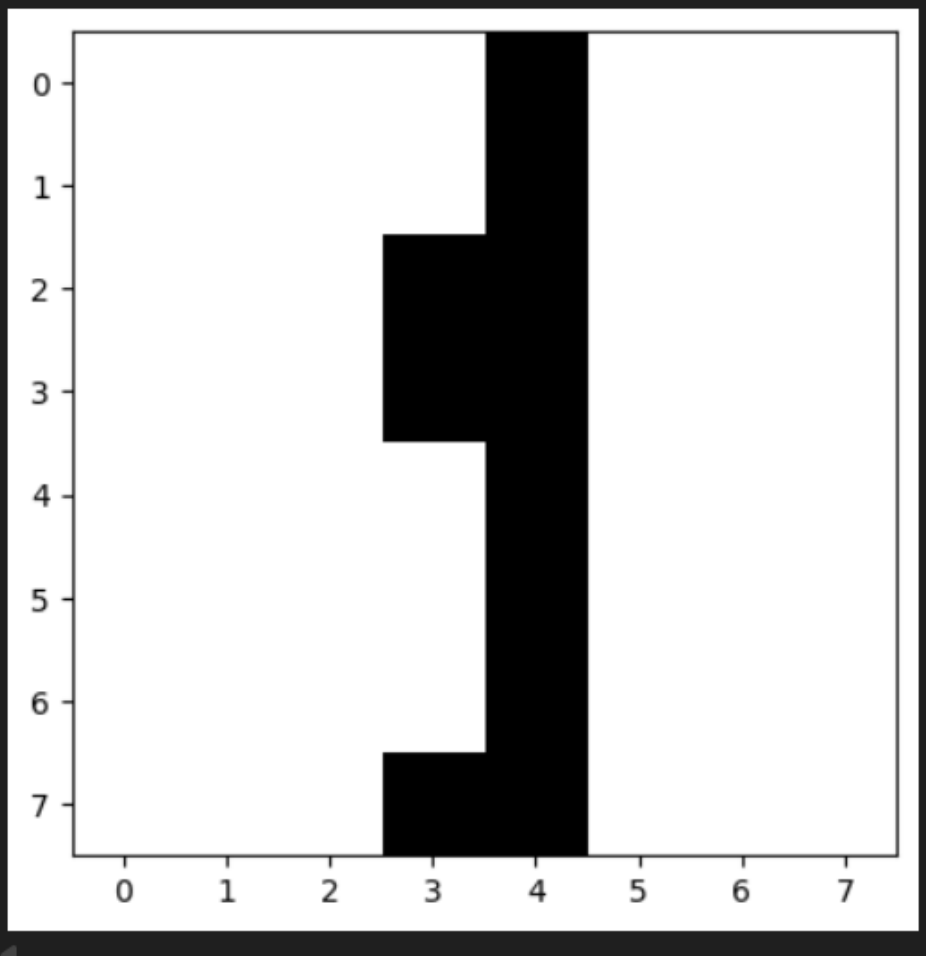

```
print('Nguyễn Thị Tường Vi - 6351071077')

img1d = im2arr.reshape([1,64])

img1d[img1d > 109] = 155

img1d[img1d < 110] = 0

img1d
```

```
Nguyễn Thị Tường Vi - 6351071077

array([[  0,   0,   0,   0, 155,   0,   0,   0,   0,   0,   0,   0, 155,
          0,   0,   0,   0,   0,   0, 155, 155,   0,   0,   0,   0,   0,
          0, 155, 155,   0,   0,   0,   0,   0,   0,   0, 155,   0,   0,
          0,   0,   0,   0,   0, 155,   0,   0,   0,   0,   0,   0,   0,
        155,   0,   0,   0,   0,   0,   0, 155, 155,   0,   0,   0]],
      dtype=uint8)
```

print('Nguyễn Thị Tường Vi - 6351071077')

plt.imshow(im2arr , cmap=plt.cm.gray_r, interpolation='nearest')

```
Nguyễn Thị Tường Vi - 6351071077

<matplotlib.image.AxesImage at 0x1c066347d90>
```



print('Nguyễn Thị Tường Vi - 6351071077')

y_pred = classifier.predict(img1d)

print(y_pred)

```
Nguyễn Thị Tường Vi - 6351071077
[1]
```

Bài 2 (Trang 32)

```
import pandas as pd

# Precision và Recall của từng phương pháp
methods = ["ID3", "C4.5", "KNN", "SVM"]
precisions = [0.82, 0.86, 0.78, 0.90]
recalls =    [0.80, 0.83, 0.74, 0.88]

# Tính F-measure cho mỗi phương pháp
f_measures = []
for p, r in zip(precisions, recalls):
    f = 2 * (p * r) / (p + r)
    f_measures.append(f)

# Đưa vào DataFrame để so sánh
df = pd.DataFrame({
    "Phương pháp": methods,
    "Precision": precisions,
    "Recall": recalls,
    "F-measure": f_measures
})
```

print(df)

```
  Phương pháp  Precision  Recall  F-measure
0         ID3       0.82    0.80   0.809877
1        C4.5       0.86    0.83   0.844734
2         KNN       0.78    0.74   0.759474
3         SVM       0.90    0.88   0.889888
```

--- HẾT---