Exercise 1: Evade

Cho dữ liệu evade trong tập tin evade.xlsx.

Yêu cầu: Hãy đọc dữ liệu từ tập tin này, áp dụng Naive Bayes để thực hiện việc xác định có evade hay không dựa trên các thông tin như: 'Refund', 'Marital Status', 'Taxable Income' ¶

Cho dữ liệu Test:

X_test = [["No", "Married", 120000],["Yes","Divorce",25000]]

Yêu cầu:

1. Hãy chuẩn hóa dữ liệu cho phù hợp
2. Áp dụng Naive Bayes. Tìm kết quả Y1 (Y test)
3. Kiểm tra độ chính xác

In [1]:
```
import pandas as pd
```

In [2]:
```
df = pd.read_excel('evade.xlsx', index_col = 0)
df
```

Out[2]:

	Refund	Marital_Status	Taxable_Income	Evade
Tid				
1	1	0	125000	0
2	0	1	100000	0
3	0	0	70000	0
4	1	1	120000	0
5	0	2	95000	1
6	0	1	60000	0
7	1	2	220000	0
8	0	0	85000	1
9	0	1	75000	0
10	0	0	90000	1

In [3]:
```
features = df[["Refund","Marital_Status","Taxable_Income"]]
target = df[["Evade"]]
```

In [4]:
```
features
```

Out[4]:

	Refund	Marital_Status	Taxable_Income
Tid			
1	1	0	125000
2	0	1	100000
3	0	0	70000
4	1	1	120000
5	0	2	95000
6	0	1	60000
7	1	2	220000
8	0	0	85000
9	0	1	75000
10	0	0	90000

In [5]:
```
target
```

Out[5]:

	Evade
Tid	
1	0
2	0
3	0
4	0
5	1
6	0
7	0
8	1
9	0
10	1

In [6]:
```
from sklearn.naive_bayes import GaussianNB
import numpy as np
from sklearn.utils.validation import column_or_1d

#Create a Gaussian Classifier
model = GaussianNB()

# Train the model using the training sets
model.fit(features, column_or_1d(target))
```

Out[6]:
```
GaussianNB(priors=None)
```

In [7]:

```
import numpy as np
# Kiểm tra độ chính xác
print("The prediction accuracy is: ", model.score(features,np.array(target))*100,"
```

```
The prediction accuracy is:  100.0 %
```

In [8]:

```
class_names = model.classes_
class_names
```

Out[8]:

```
array([0, 1], dtype=int64)
```

In [9]:

```
# X_test = [["No", "Married", 120000],["No","Single",90000]]
X_test = [[0, 1, 120000], [0, 0, 90000]]
y_pred = model.predict(X_test)
y_pred
```

Out[9]:

```
array([0, 1], dtype=int64)
```

In [10]:

```
import pickle
# Save to file in the current working directory
pkl_filename = "pickle_model.pkl"
with open(pkl_filename, 'wb') as file:
    pickle.dump(model, file)
```

In [11]:

```
with open(pkl_filename, 'rb') as file:
    pickle_model = pickle.load(file)
```

In [12]:

```
X_test = [[0, 0, 75000]]
y_pred = pickle_model.predict(X_test)
y_pred
```

Out[12]:

```
array([0], dtype=int64)
```

In [ ]: