AI VIET NAM – COURSE 2024

Probability Exercise

(Naive Bayes Classifier)

Ngày 20 tháng 7 năm 2024

Giả sử X có các đặc trưng thuộc tính độc lập với nhau $x_1, x_2, ..., x_n$, để phân loại X vào lớp một trong các lớp $C = c_1, c_2, ..., c_m$, dựa vào công thức Bayes ta có:

$$P(c|X) = \frac{P(X|c).P(c)}{P(X)}$$

Dựa vào ước lượng tối đa xác suất hậu nghiệm (MAP - Maximum A Posterior) ta được:

$$P(c|X) \propto P(X|c).P(c)$$

$$P(c|X) \propto P(x_1|c).P(x_2|c)...P(x_n|c).P(c)$$

1. BINARY CLASSIFICATION - PLAY TENNIS

Cho tập dữ liệu huấn luyện mô hình phân loại nhị phân Naive Bayes gồm các thuộc tính "Outlook", "Temperature", "Humidity", "Wind":

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Overcast	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes

Bảng 1: Play Tennis - Tập dữ liệu huấn luyện

Cho sự kiện thử nghiệm:
X = (Outlook=Sunny, Temperature=Cool, Humidity=High, Wind=Strong)

Câu hỏi 1: Xác suất xảy ra sự kiện "Play Tennis"="Yes" và sự kiện "Play Tennis"="No" lần lượt là:

a) P("Play Tennis" = "Yes") = 6/10, P("Play Tennis" = "No") = 4/10

b) P("Play Tennis" = "Yes") = 4/10, P("Play Tennis" = "No") = 6/10

c) P("Play Tennis" = "Yes") = 6/10, P("Play Tennis" = "No") = 6/10

d) P("Play Tennis" = "Yes") = 4/10, P("Play Tennis" = "No") = 4/10

Câu hỏi 2: Xác suất xảy ra sự kiện "Play Tennis"="Yes" khi sự kiện X xảy ra là:

a) P("Play Tennis" = "Yes"|X) $\propto$ 0.0014

b) P("Play Tennis" = "Yes"|X) $\propto$ 0.0028

c) P("Play Tennis" = "Yes"|X) $\propto$ 0.0188

d) P("Play Tennis" = "Yes"|X) $\propto$ 0.0098

Câu hỏi 3: Xác suất xảy ra sự kiện "Play Tennis"="No" khi sự kiện X xảy ra là:

a) P("Play Tennis" = "No" | X) $\propto$ 0.0014

b) P("Play Tennis" = "No" | X) $\propto$ 0.0028

c) P("Play Tennis" = "No" | X) $\propto$ 0.0188

d) P("Play Tennis" = "No" | X) $\propto$ 0.0098

Câu hỏi 4: Khi xảy ra sự kiện X, nhãn của "Play Tennis" sẽ là:

a) "Play Tennis" = "Yes"

b) "Play Tennis" = "No"

2. MULTI-LABEL CLASSIFICATION - TRAFFIC DATA Cho tập dữ liệu huấn luyện mô hình phân loại Naive Bayes gồm các thuộc tính "Day", "Season", "Fog", "Rain".

Day	Season	Fog	Rain	Class
Weekday	Spring	None	None	On Time
Weekday	Winter	None	Slight	On Time
Weekday	Winter	None	None	On Time
Holiday	Winter	High	Slight	Late
Saturday	Summer	Normal	None	On Time
Weekday	Autumn	Normal	None	Very Late
Holiday	Summer	High	Slight	On Time
Sunday	Summer	Normal	None	On Time
Weekday	Winter	High	Heavy	Very Late
Weekday	Summer	None	Slight	On Time
Saturday	Spring	High	Heavy	Cancelled
Weekday	Summer	High	Slight	On Time
Weekday	Winter	Normal	None	Late
Weekday	Summer	High	None	On Time
Weekday	Winter	Normal	Heavy	Vary Late
Saturday	Autumn	High	Slight	On Time
Weekday	Autumn	None	Heavy	On Time
Holiday	Spring	Normal	Slight	On Time
Weekday	Spring	Normal	None	On Time
Weekday	Spring	Normal	Heavy	On Time

Bảng 2: Traffic Data - Tập dữ liệu huấn luyện

Cho sự kiện thử nghiệm:
X = (Day=Weekday, Season=Winter, Fog=High, Rain=Heavy)

Câu hỏi 5: Xác suất xảy ra sự kiện "Class"="On Time", sự kiện "Class"="Late", sự kiện "Class"=" Very Late" và sự kiện "Class"="Cancelled" lần lượt là:

(A) P("Class" = "On Time") = 14/20, P("Class" = "Late") = 2/20,
P("Class" = "Very Late") = 3/20, P("Class" = "Cancelled") = 1/20

(B) P("Class" = "On Time") = 2/20, P("Class" = "Late") = 3/20,
P("Class" = "Very Late") = 1/20, P("Class" = "Cancelled") = 14/20

(C) P("Class" = "On Time") = 3/20, P("Class" = "Late") = 1/20,
P("Class" = "Very Late") = 2/20, P("Class" = "Cancelled") = 14/20

(D) P("Class" = "On Time") = 1/20, P("Class" = "Late") = 1/20,
P("Class" = "Very Late") = 14/20, P("Class" = "Cancelled") = 3/20

Câu hỏi 6: Xác suất xảy ra sự kiện "Class"="On Time" khi sự kiện X xảy ra là:

(A) P("Class" = "On Time" | X) $\propto$ 0.0222

(B) P("Class" = "On Time" | X) $\propto$ 0.0013

(C) P("Class" = "On Time" | X) $\propto$ 0.0026

(D) P("Class" = "On Time" | X) $\propto$ 0.0000

Câu hỏi 7: Xác suất xảy ra sự kiện "Class"="Late" khi sự kiện X xảy ra là:

(A) P("Class" = "Late" | X) $\propto$ 0.0222

(B) P("Class" = "Late" | X) $\propto$ 0.0013

(C) P("Class" = "Late" | X) $\propto$ 0.0026

(D) P("Class" = "Late" | X) $\propto$ 0.0000

Câu hỏi 8: Xác suất xảy ra sự kiện "Class"= "Very Late" khi sự kiện X xảy ra là:

(A) P("Class" = "Very Late" | X) $\propto$ 0.0222

(B) P("Class" = "Very Late" | X) $\propto$ 0.0013

(C) P("Class" = "Very Late" | X) $\propto$ 0.0026

(D) P("Class" = "Very Late" | X) $\propto$ 0.0000

Câu hỏi 9: Xác suất xảy ra sự kiện "Class"= Cancelled" khi sự kiện X xảy ra là:

(A) P("Class" = "Cancelled" | X) $\propto$ 0.0222

(B) P("Class" = "Cancelled" | X) $\propto$ 0.0013

(C) P("Class" = "Cancelled" | X) $\propto$ 0.0026

(D) P("Class" = "Cancelled" | X) $\propto$ 0.0000

Câu hỏi 10: Dự đoán "Class" của sự kiện X là:

(A) "On Time"

(B) "Late"

(C) "Very Late"

(D) "Cancelled"

3. IRIS CLASSIFICATION

Cho một tập dữ liệu huấn luyện phân loại hoa Iris dựa vào chiều dài cánh hoa như bảng dữ liệu bên dưới. Các bạn hãy trả lời các câu hỏi sau khi dùng Gaussian Naive Bayes cho data Iris này.

Length	1.4	1.0	1.3	1.9	2.0	1.8	3.0	3.8	4.1	3.9	4.2	3.4
Class	0	0	0	0	0	0	1	1	1	1	1	1

Bảng 3: Phân loại cánh hoa Iris dựa vào chiều dài cánh hoa - Tập dữ liệu huấn luyện

Câu hỏi 11: Giá trị mean và variance của biến đầu vào (Length) cho "Class"="0" lần lượt là:

a) mean = 1.566 và variance = 0.128

b) mean = 3.733 và variance = 0.172

c) mean = 1.566 và variance = 0.172

Câu hỏi 12: Giá trị mean và variance của biến đầu vào (Length) cho "Class"="1" lần lượt là:

a) mean = 1.566 và variance = 0.128

b) mean = 3.733 và variance = 0.172

c) mean = 1.566 và variance = 0.172

Câu hỏi 13: Cho dữ liệu kiểm thử X = (Length=3.4). Xác suất dữ liệu kiểm thử X thuộc vào "Class"="0" và "Class"="1" lần lượt là:

a) P("Class" = "0" | X) = $1.09 * 10^{-6}$ và P("Class" = "1" | X) = 0.3486

b) P("Class" = "0" | X) = $1.09 * 10^{-4}$ và P("Class" = "1" | X) = 0.3486

c) P("Class" = "0" | X) = $1.09 * 10^{-2}$ và P("Class" = "1" | X) = 0.3486

4. PLAY TENNIS CLASSIFIER IMPLEMENTATION

Cho trước dữ liệu thời tiết của 10 ngày (D1-D10, như bảng 1). Hãy phát triển chương trình sử dụng mô hình phân loại Naive Bayes để dự đoán xem ngày thứ 11 (D11), AD có thể chơi tennis hay không?

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D11	Sunny	Cool	High	Strong	???

Bảng 4: Play Tennis - Dữ liệu testing

(a) "Play Tennis" = "Yes"

(b) "Play Tennis" = "No"

Để hoàn thành bài tập này bạn cần hoàn thành các function sau đây bằng cách sử dụng thư viên numpy:

4.1 Hoàn thiện function **create_train_dataset()** để tổ chức dữ liệu bảng 1 vào array 2 chiều như bên dưới.

```
#########################
# Create data
#########################
import numpy as np

def create_train_data():

   #your code here ***********************

  return np.array(data)

train_data = create_train_data()
print(train_data)

*****************Sample Result when we print out train_data *****************
[['Sunny' 'Hot' 'High' 'Weak' 'no']
 ['Sunny' 'Hot' 'High' 'Strong' 'no']
 ['Overcast' 'Hot' 'High' 'Weak' 'yes']
 ['Rain' 'Mild' 'High' 'Weak' 'yes']
 ['Rain' 'Cool' 'Normal' 'Weak' 'yes']
 ['Rain' 'Cool' 'Normal' 'Strong' 'no']
 ['Overcast' 'Cool' 'Normal' 'Strong' 'yes']
 ['Overcast' 'Mild' 'High' 'Weak' 'no']
 ['Sunny' 'Cool' 'Normal' 'Weak' 'yes']
 ['Rain' 'Mild' 'Normal' 'Weak' 'yes']]
```

4.2 Hoàn thiện function **compute_prior_probability** tính P("Play Tennis" = "Yes") and tính P("Play Tennis" = "No") như bên dưới:

```
def compute_prior_probablity(train_data):
    y_unique = ['no', 'yes']
    prior_probability = np.zeros(len(y_unique))
    # your code here *****************
    return prior_probability

prior_probablity = compute_prior_probablity(train_data)
print("P(play tennis = No"), prior_probablity[0])
print("P(play tennis = Yes"), prior_probablity[1])
```

Câu hỏi 14: Kết quả nào sau đây là output từ chương trình trên:

a) P("Play Tennis" = "Yes") = 0.6, P("Play Tennis" = "No") = 0.4

b) P("Play Tennis" = "Yes") = 0.3, P("Play Tennis" = "No") = 0.7

c) P("Play Tennis" = "Yes") = 0.4, P("Play Tennis" = "No") = 0.8

d) P("Play Tennis" = "Yes") = 0.4, P("Play Tennis" = "No") = 0.3

4.3 Hoàn thiện function **compute_conditional_probability** để tính likelihood (The probability of "A" being True. Given "B" True, P(A|B)) như bên dưới:

```
def compute_conditional_probability(train_data):
    y_unique = ['no', 'yes']
    conditional_probability = []
    list_x_name = []
    for i in range(0,train_data.shape[1]-1):
        x_unique = np.unique(data[:,i])
        list_x_name.append(x_unique)

        # your code here ********************

conditional_probability.append(x_conditional_probability)
return conditional_probability, list_x_name

```

Câu hỏi 15: Hãy cho biết kết quả của đoạn chương trình sau đây:

```
train_data = create_train_data()
_, list_x_name  = compute_conditional_probability(train_data)
print("x1 = ",list_x_name[0])
print("x2 = ",list_x_name[1])
print("x3 = ",list_x_name[2])
print("x4 = ",list_x_name[3])
```

a) x1 = ['Cool' 'Hot' 'Mild']
x2 = ['Overcast' 'Rain' 'Sunny']
x3 = ['High' 'Normal']
x4 = ['Strong' 'Weak']

b) x1 = ['Overcast' 'Rain' 'Sunny']
x2 = ['Cool' 'Hot' 'Mild']
x3 = ['High' 'Normal']
x4 = ['Strong' 'Weak']

c) x1 = ['Strong' 'Weak']
x2 = ['Cool' 'Hot' 'Mild']
x3 = ['High' 'Normal']
x4 = ['Overcast' 'Rain' 'Sunny']

d) x1 = ['Overcast' 'Rain' 'Sunny']
x2 = ['Cool' 'Hot' 'Mild']
x3 = ['Strong' 'Weak']
x4 = ['High' 'Normal']

4.4 Hoàn thiện function **get_index_from_value** để tính trả về index tương ứng với feature name:

```
#This function is used to return the index of the feature name
def get_index_from_value(feature_name, list_features):
    return np.where(list_eatures == feature_ame)[0][0]
```

Câu hỏi 16: Hãy cho biết kết quả của đoạn chương trình sau đây:

```
train_data = create_train_data()
_, list_x_name  = compute_conditional_probability(train_data)
outlook = list_x_name[0]

i1 = get_index_from_value("Overcast", outlook)
i2 = get_index_from_value("Rain", outlook)
i3 = get_index_from_value("Sunny", outlook)

print(i1, i2, i3)
```

a) 1 2 0

b) 0 1 1

c) 0 1 2

d) 0 2 3

Câu hỏi 17: Hãy cho biết kết quả của đoạn chương trình sau đây:

```
train_data = create_train_data()
conditional_probability, list_x_name  = compute_conditional_probability(train_data)
# Compute P("Outlook"="Sunny"|Play Tennis"="Yes")
x1=get_index_from_value("Sunny",list_x_name[0])
print("P('Outlook'='Sunny'|Play Tennis'='Yes') = ", np.round(conditional_probability
    [0][1, x1],2))
```

a) P('Outlook'='Sunny'|Play Tennis'='Yes') = 0.27

b) P('Outlook'='Sunny'|Play Tennis'='Yes') = 0.47

c) P('Outlook'='Sunny'|Play Tennis'='Yes') = 0.37

d) P('Outlook'='Sunny'|Play Tennis'='Yes') = 0.17

Câu hỏi 18: Hãy cho biết kết quả của đoạn chương trình sau đây:

```
train_data = create_train_data()
conditional_probability, list_x_name  = compute_conditional_probability(train_data)
# Compute P("Outlook"="Sunny"|Play Tennis"="No")
x1=get_index_from_value("Sunny",list_x_name[0])
print("P('Outlook'='Sunny'|Play Tennis'='No') = ", np.round(conditional_probability
    [0][1, x1],2))
```

a) P('Outlook'='Sunny'|Play Tennis'='Yes') = 0.5

b) P('Outlook'='Sunny'|Play Tennis'='Yes') = 0.4

c) P('Outlook'='Sunny'|Play Tennis'='Yes') = 0.3

d) P('Outlook'='Sunny'|Play Tennis'='Yes') = 0.2

4.5 Hoàn thiện function **train_naive_bayes** như bên dưới:

```
############################
# Train Naive Bayes Model
############################
def train_naive_bayes(train_data):
    # Step 1: Calculate Prior Probability
    y_unique = ['no', 'yes']
    prior_probability = compute_prior_probablity(train_data)

    # Step 2: Calculate Conditional Probability
    conditional_probability, list_x_name  = compute_conditional_probability(train_data
    )

    return prior_probability,conditional_probability, list_x_name
```

4.6 Hoàn thiện function **prediction_play_tennis** để hỗ trợ AD có nên đi chơi tennis vào ngày D11 không:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D11	Sunny	Cool	High	Strong	???

Bảng 5: Play Tennis - Dữ liệu testing

```
####################
# Prediction
####################
def prediction_play_tennis(X, list_x_name, prior_probability, conditional_probability)
    :

    x1=get_index_from_value(X[0],list_x_name[0])
    x2=get_index_from_value(X[1],list_x_name[1])
    x3=get_index_from_value(X[2],list_x_name[2])
    x4=get_index_from_value(X[3],list_x_name[3])

    p0 = 0
    p1 = 0

    # your code here **********************

    if p0>p1:
        y_pred=0
    else:
        y_pred=1

    return y_pred
```

Câu hỏi 19: Hãy cho biết kết quả của đoạn chương trình sau đây:

```
X = ['Sunny','Cool', 'High', 'Strong']
data = create_train_data()
prior_probability,conditional_probability, list_x_name = train_naive_bayes(data)
pred =  prediction_play_tennis(X, list_x_name, prior_probability,
    conditional_probability)

if(pred):
  print("Ad should go!")
else:
  print("Ad should not go!")
```

a) Ad should not go!

b) Ad should go!

5. (OPTIONAL) IRIS CLASSIFIER IMPLEMENTATION

Cho trước dữ liệu chứa thông tin về hoa Iris gồm có sepal length, sepal width và petal length, và Species (bảng 6). Hãy phát triển chương trình sử dụng mô hình phân loại Gausian Naive Bayes để dự đoán chủng loại của hoa Iris. Dữ liệu hoa iris được lưu trữ trong file iris_data.txt có thể được tải về tại đây.

No.	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	6.4	3.1	5.5	1.8	Iris-virginica
4	6.0	3.0	4.8	1.8	Iris-virginica
5	6.0	2.2	4.0	1.0	Iris-versicolora
...	...	...	...	..	...

Bảng 6: Iris flower - Tập dữ liệu huấn luyện

Dựa vào hướng dẫn dưới đây để thực thi mã nguồn cho bài toán phân loại.

```
# Example 1
# X =[sepal length, sepal width, petal length, petal width]
X = [6.3 , 3.3, 6.0,  2.5]
train_data = create_train_data_iris()
y_unique = np.unique(train_data[:,4])
prior_probability, conditional_probability = train_gaussian_naive_bayes(train_data)
pred = y_unique[prediction_iris(X, prior_probability, conditional_probability)]
assert pred == "Iris-virginica"

#Example 2 ########################
# X =[sepal length, sepal width, petal length, petal width]
X = [5.0,2.0,3.5,1.0]
train_data = create_train_data_iris()
y_unique = np.unique(train_data[:,4])
prior_probability, conditional_probability = train_gaussian_naive_bayes(train_data)
pred = y_unique[prediction_iris(X, prior_probability, conditional_probability)]
assert pred == "Iris-versicolor"

#Example 3 ########################
X = [4.9,3.1,1.5,0.1]
# X =[sepal length, sepal width, petal length, petal width]
train_data = create_train_data_iris()
y_unique = np.unique(train_data[:,4])
prior_probability, conditional_probability = train_gaussian_naive_bayes(train_data)
pred = y_unique[prediction_iris(X, prior_probability, conditional_probability)]
assert pred == "Iris-setosa"
```