



Exercise 3: Mammals or Non-mammals ¶

Cho dữ liệu Mammals trong tập tin mammal.xlsx.

Yêu cầu: Hãy đọc dữ liệu từ tập tin này, áp dụng Naive Bayes để thực hiện việc xác định có là mammals hay không dựa trên các thông tin như: 'Give birth', 'Can fly', 'Live in water', 'Have legs'

Yêu cầu:

1. Hãy chuẩn hóa dữ liệu cho phù hợp
2. Áp dụng Naive Bayes. Tìm kết quả.
3. Cho dữ liệu Test: $X_{\text{test}} = [["Yes", "No", "Yes", "No"], ["No", "No", "sometimes", "Yes"]] \Rightarrow y_{\text{test}}$
4. Kiểm tra độ chính xác

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

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

Out[2]:

	Give Birth	Can Fly	Live in Water	Have Legs	Class
Name					
human	1	0	0	1	1
python	0	0	0	0	0
salmon	0	0	1	0	0
whale	1	0	1	0	1
frog	0	0	2	1	0

```
In [3]: features = df[["Give Birth", "Can Fly", "Live in Water", "Have Legs"]]
target = df[["Class"]]
```

In [4]: `features.head()`

Out[4]:

	Give Birth	Can Fly	Live in Water	Have Legs
Name				
human	1	0	0	1
python	0	0	0	0
salmon	0	0	1	0
whale	1	0	1	0
frog	0	0	2	1

In [5]: `target.head()`

Out[5]:

	Class
Name	
human	1
python	0
salmon	0
whale	1
frog	0

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

#Create a Gaussian Classifier

`model = GaussianNB()`

Train the model using the training sets

`model.fit(features, target)`

c:\program files\python36\lib\site-packages\sklearn\utils\validation.py:578: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
 y = column_or_1d(y, warn=True)

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: 90.0 %

In [8]: `class_names = model.classes_`
`class_names`

Out[8]: `array([0, 1], dtype=int64)`

```
In [9]: # X_test = [["Yes", "No", "Yes", "No"], ["No", "No", "sometimes", "Yes"]]  
X_test = [[1, 0, 1, 0], [0, 0, 2, 1]]  
y_pred = model.predict(X_test)  
y_pred
```

```
Out[9]: array([1, 0], dtype=int64)
```

```
In [ ]:
```