



## Exercise 4: Mammal or Reptile

### Cho dữ liệu như sau:

```
data = pd.DataFrame({"toothed":
["True","True","True","False","True","True","True","True","True","False"], "hair":
["True","True","False","True","True","True","False","False","True","False"], "breathes":
["True","True","True","True","True","True","False","True","True","True"], "legs":
["True","True","False","True","True","True","False","False","True","True"], "species":
["Mammal","Mammal","Reptile","Mammal","Mammal","Mammal","Reptile","Reptile","Mammal","Reptile"]
columns=["toothed","hair","breathes","legs","species"])
```

### Cho dữ liệu test như sau:

```
X_test = [ ["True", "True", "True", "True"], ["True", "False", "True", "False"],
["False","False","False","False"]]
```

### Yêu cầu 1: Từ data, hãy thực hiện

1. Chuẩn hóa dữ liệu
2. Áp dụng naive bayes. Tìm kết quả Y1 (Y Test)
3. Kiểm tra độ chính xác

### Yêu cầu 2: Từ data, hãy thực hiện

1. Chuẩn hóa dữ liệu
2. Tạo X\_train, X\_test, y\_train, y\_test từ dữ liệu chuẩn hóa với tỷ lệ dữ liệu test là 0.2
3. Áp dụng Naive Bayes, Tìm kết quả
4. Kiểm tra độ chính xác

```
In [1]: # Mammal or Reptile (Động vật có vú hay bò sát)
import pandas as pd
# Cho dữ liệu training như sau:
data = pd.DataFrame({"toothed":["True","True","True","False","True","True","True",
                                "hair":["True","True","False","True","True","True","False","F
                                "breathes":["True","True","True","True","True","True","False"
                                "legs":["True","True","False","True","True","True","False","F
                                "species":["Mammal","Mammal","Reptile","Mammal","Mammal","Mam
                                columns=["toothed","hair","breathes","legs","species"]})

# Cho dữ liệu test như sau:
X_test = [ ["True", "True", "True", "True"], ["True", "False", "True", "False"], ["F
```

```
In [2]: data
```

```
Out[2]:
```

	toothed	hair	breathes	legs	species
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
2	True	False	True	False	Reptile
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile
8	True	True	True	True	Mammal
9	False	False	True	True	Reptile

```
In [3]: features = data[["toothed","hair","breathes","legs"]]  
target = data[["species"]]
```

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

```
Out[4]:
```

	toothed	hair	breathes	legs
0	True	True	True	True
1	True	True	True	True
2	True	False	True	False
3	False	True	True	True
4	True	True	True	True
5	True	True	True	True
6	True	False	False	False
7	True	False	True	False
8	True	True	True	True
9	False	False	True	True

```
In [5]: features.dtypes
```

```
Out[5]: toothed    object  
hair          object  
breathes      object  
legs          object  
dtype: object
```

```
In [6]: features = features.replace("True",1.)  
features = features.replace("False",0.)
```

In [7]: features

Out[7]:

	toothed	hair	breathes	legs
0	1.0	1.0	1.0	1.0
1	1.0	1.0	1.0	1.0
2	1.0	0.0	1.0	0.0
3	0.0	1.0	1.0	1.0
4	1.0	1.0	1.0	1.0
5	1.0	1.0	1.0	1.0
6	1.0	0.0	0.0	0.0
7	1.0	0.0	1.0	0.0
8	1.0	1.0	1.0	1.0
9	0.0	0.0	1.0	1.0

In [8]: target

Out[8]:

	species
0	Mammal
1	Mammal
2	Reptile
3	Mammal
4	Mammal
5	Mammal
6	Reptile
7	Reptile
8	Mammal
9	Reptile

In [9]: target.dtypes

Out[9]: species     object  
dtype: object

```
In [10]: from sklearn.naive_bayes import BernoulliNB
clf = BernoulliNB()
clf.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[10]: BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
```

```
In [11]: prediction = clf.predict([[1., 0., 1., 0.]])
print(prediction)

['Reptile']
```

```
In [12]: X_test = pd.DataFrame(X_test)
X_test = X_test.replace("True", 1.0)
X_test = X_test.replace("False", 0.0)
prediction = clf.predict(X_test)
print(prediction)

['Mammal' 'Reptile' 'Reptile']
```

```
In [13]: # tham do ket qua tong quat
from sklearn.metrics import classification_report, confusion_matrix
y_pred = clf.predict(X_test)

y_test = pd.DataFrame(pd.Series(['Mammal', 'Reptile', 'Reptile']))

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[1 0]
 [0 2]]
```

	precision	recall	f1-score	support
Mammal	1.00	1.00	1.00	1
Reptile	1.00	1.00	1.00	2
avg / total	1.00	1.00	1.00	3

```
In [14]: # cách khác
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.5,
y_test
```

```
Out[14]:
```

	species
3	Mammal
7	Reptile

```
In [15]: classifier = BernoulliNB()
classifier.fit(X_train, y_train)
```

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[15]: BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
```

```
In [16]: y_pred = classifier.predict(X_test)
y_pred
```

```
Out[16]: array(['Mammal', 'Reptile'], dtype='<U7')
```

```
In [17]: from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
[[1 0]
 [0 1]]
```

	precision	recall	f1-score	support
Mammal	1.00	1.00	1.00	1
Reptile	1.00	1.00	1.00	1
avg / total	1.00	1.00	1.00	2

```
In [ ]:
```