



Chapter 6 - exercise 2: Mushroom

Cho dữ liệu mushroom trong tập tin mushrooms.csv chứa thông tin của các mẫu nấm, nấm ăn được và không ăn được. Yêu cầu: Áp dụng thuật toán decision tree để cho biết nấm ăn được hay nấm độc dựa trên các thông tin được cung cấp.

- Dữ liệu có thể tham khảo và download tại: <https://www.kaggle.com/jnduli/decision-tree-classifier-for-mushroom-dataset/data> (<https://www.kaggle.com/jnduli/decision-tree-classifier-for-mushroom-dataset/data>)

Data Infomation

Bộ dữ liệu chứa 23 thuộc tính. Thuộc tính "class" là class attribute: Attribute Information: (classes: edible=e, poisonous=p)

- cap-shape: bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s
- cap-surface: fibrous=f,grooves=g,scaly=y,smooth=s
- cap-color: brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y
- bruises: bruises=t,no=f
- odor: almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s
- gill-attachment: attached=a,descending=d,free=f,notched=n
- gill-spacing: close=c,crowded=w,distant=d
- gill-size: broad=b,narrow=n
- gill-color: black=k,brown=n,buff=b,chocolate=h,gray=g, green=r,orange=o,pink=p,purple=u,red=e,white=w,yellow=y
- stalk-shape: enlarging=e,tapering=t
- stalk-root: bulbous=b,club=c,cup=u,equal=e,rhizomorphs=z,rooted=r,missing=?
- stalk-surface-above-ring: fibrous=f,scaly=y,silky=k,smooth=s
- stalk-surface-below-ring: fibrous=f,scaly=y,silky=k,smooth=s
- stalk-color-above-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y
- stalk-color-below-ring: brown=n,buff=b,cinnamon=c,gray=g,orange=o,pink=p,red=e,white=w,yellow=y
- veil-type: partial=p,universal=u
- veil-color: brown=n,orange=o,white=w,yellow=y
- ring-number: none=n,one=o,two=t
- ring-type: cobwebby=c,evanescent=e,flaring=f,large=l,none=n,pendant=p,sheathing=s,zone=z
- spore-print-color: black=k,brown=n,buff=b,chocolate=h,green=r,orange=o,purple=u,white=w,yellow=y
- population: abundant=a,clustered=c,numerous=n,scattered=s,several=v,solitary=y
- habitat: grasses=g,leaves=l,meadows=m,paths=p,urban=u,waste=w,woods=d

Yêu cầu:

- Đọc dữ liệu, tìm hiểu sơ bộ về dữ liệu. Chuẩn hóa dữ liệu nếu cần
- Tạo X_train, X_test, y_train, y_test từ dữ liệu chuẩn hóa với tỷ lệ dữ liệu test là 0.3



- Áp dụng Decision Tree, Tìm kết quả
- Kiểm tra độ chính xác
- Trực quan hóa Decision Tree
- Đánh giá mô hình.
- Ghi mô hình nếu mô hình phù hợp

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

```
In [2]: dataset = pd.read_csv('mushrooms.csv', sep=",")
print(dataset.shape)
#dataset.info()

(8124, 23)
```

```
In [3]: dataset.head()
```

Out[3]:

	class	cap- shape	cap- surface	cap- color	bruises	odor	gill- attachment	gill- spacing	gill- size	gill- color	...	stalk- surface- below- ring	st: co abo r
0	p	x	s	n	t	p	f	c	n	k	...	s	
1	e	x	s	y	t	a	f	c	b	k	...	s	
2	e	b	s	w	t	l	f	c	b	n	...	s	
3	p	x	y	w	t	p	f	c	n	n	...	s	
4	e	x	s	g	f	n	f	w	b	k	...	s	

5 rows × 23 columns



```
In [4]: # Vì các biến phân loại không tồn tại mối quan hệ thứ tự => cần chuẩn hóa bằng one
```

```
In [5]: y = dataset['class']
x = dataset.drop(['class'], axis=1)
x = pd.get_dummies(x)
```

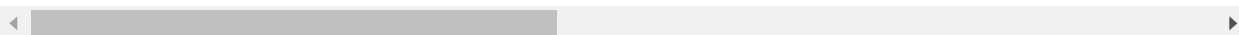


In [6]: `x.head()`

Out[6]:

	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	surfa
0	0	0	0	0	0	1	0	0	1	
1	0	0	0	0	0	1	0	0	1	
2	1	0	0	0	0	0	0	0	1	
3	0	0	0	0	0	1	0	0	0	
4	0	0	0	0	0	1	0	0	1	

5 rows × 117 columns



In [7]: `y.head()`

Out[7]:

```
0    p
1    e
2    e
3    p
4    e
Name: class, dtype: object
```

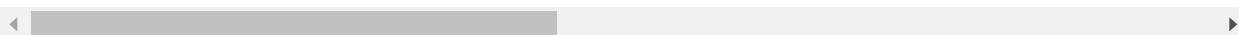
In [8]: `# trong trường hợp có quá nhiều cột dữ liệu có thể dùng dummy encoder để tạo các c`
`features= pd.get_dummies(x, drop_first=True)`
`target = y`

In [9]: `features.head()`

Out[9]:

	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	surfa
0	0	0	0	0	0	1	0	0	1	
1	0	0	0	0	0	1	0	0	1	
2	1	0	0	0	0	0	0	0	1	
3	0	0	0	0	0	1	0	0	0	
4	0	0	0	0	0	1	0	0	1	

5 rows × 117 columns



In [10]: `# Đếm theo Loại`
`occ = target.value_counts()`
`occ`

Out[10]: `e 4208`
`p 3916`
Name: class, dtype: int64



```
In [11]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2)
```

```
In [12]: from sklearn.tree import DecisionTreeClassifier
from sklearn.utils.validation import column_or_1d
```

```
In [13]: tree_n = DecisionTreeClassifier()
tree_n.fit(X_train, y_train)
```

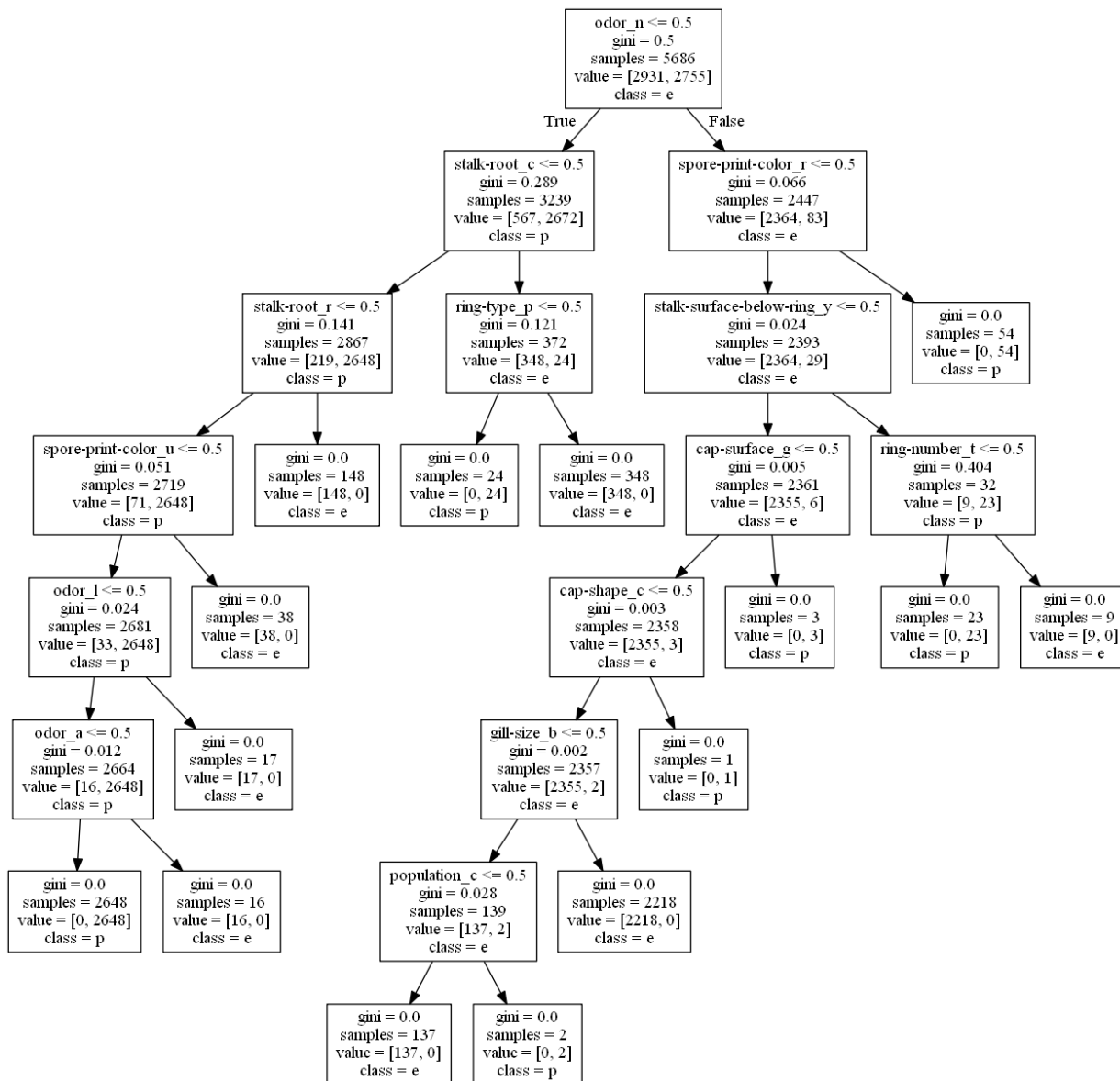
```
Out[13]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
max_features=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort=False, random_state=None,
splitter='best')
```

```
In [14]: from IPython.display import Image
from sklearn import tree
import pydotplus
```



```
In [15]: dot_data = tree.export_graphviz(tree_n, out_file=None,
                                         feature_names=features.columns,
                                         class_names=['e', 'p'])
graph = pydotplus.graph_from_dot_data(dot_data)
Image(graph.create_png())
```

Out[15]:



```
In [16]: # Kiểm tra độ chính xác
print("The Training prediction accuracy is:",
      tree_n.score(X_train,y_train)*100,"%")
print("The Testing prediction accuracy is:",
      tree_n.score(X_test,y_test)*100,"%")
```

The Training prediction accuracy is: 100.0 %
The Testing prediction accuracy is: 100.0 %

```
In [17]: # Đánh giá model
```

```
In [18]: y_pred = tree_n.predict(X_test)
```



In [19]: `yTrain_pred = tree_n.predict(X_train)`

In [20]: `# Xem kết quả thống kê`
`from sklearn.metrics import classification_report, confusion_matrix`
`print(confusion_matrix(y_test, y_pred))`
`print(classification_report(y_test, y_pred))`

```
[[1277    0]
 [    0 1161]]
```

	precision	recall	f1-score	support
e	1.00	1.00	1.00	1277
p	1.00	1.00	1.00	1161
micro avg	1.00	1.00	1.00	2438
macro avg	1.00	1.00	1.00	2438
weighted avg	1.00	1.00	1.00	2438

Nhận xét:

- Cả train và test đều có Score cao, không bị overfitting/underfitting
- => Mdoel phù hợp