MYONGJI UNIVERSITY

Artificial Intelligence

Lecture 8. Classification

Random Forest 실습

Spring 2022

Prof. Jonghoon Chun, Ph.D.

E-mail : jchun@mju.ac.kr
Lecture Note: http://lms.mju.ac.kr

학습 데이터와 테스트 데이터 나누기

- train_test_split(data, label, options)
  - 학습 데이터와 테스트 데이터를 자동으로 나누어주는 함수
  - data: 2차원(pandas의 dataframe 또는 numpy의 2차원 array)
  - label: 1차원(pandas의 Series 또는 numpy의 1차원 array)
  - Options
    - test_size: 테스트 데이터 비율, default는 0.25
    - random_state: seed for random number generator
    - Eg) test_size=0.33, random_state=42

In [ ]:
```
import pandas as pd
from sklearn.model_selection import train_test_split

# 학습 전용 데이터와 테스트 전용 데이터로 나누기
train_data, test_data, train_label, test_label = \
train_test_split(csv_data, csv_label)
```

Random Forests

```
from sklearn.ensemble import RandomForestClassifier
…
clf = RandomForestClassifier( )
clf.fit(data_train, label_train)
predict = clf.predict(data_test)
```

- RandomForestClassifier parameter
  - n_estimators: default 100, number of trees in the forest
  - criterion: default gini, 혹은 entropy로 설정할 수 있음
  - min_impurity_split: threshould 값 이하면 decision tree 생성을 stop
  - 기타 다양한 parameter들이 존재

독버섯 분류

- 데이터 다운로드

In [140]:
```
import urllib.request as req
local= "mushroom.csv"
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/mushroom/agaricus-lepiota.data"
req.urlretrieve(url, local)
print("ok")
```

```
ok
```

```
p,x,s,n,t,p,f,c,n,k,e,e,s,s,w,w,p,w,o,p,k,s,u
e,x,s,y,t,a,f,c,b,k,e,c,s,s,w,w,p,w,o,p,n,n,g
e,b,s,w,t,l,f,c,b,n,e,c,s,s,w,w,p,w,o,p,n,n,m
p,x,y,w,t,p,f,c,n,n,e,e,s,s,w,w,p,w,o,p,k,s,u
e,x,s,g,f,n,f,w,b,k,t,e,s,s,w,w,p,w,o,e,n,a,g
e,x,y,y,t,a,f,c,b,n,e,c,s,s,w,w,p,w,o,p,k,n,g
e,b,s,w,t,a,f,c,b,g,e,c,s,s,w,w,p,w,o,p,k,n,m
```

Column 1: p(독버섯), e(식용)
2: 버섯의 머리모양
4: 버섯의 머리색 …
등 총 22개의 feature로 구성

One-hot encoding

In [141]:
```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.model_selection import train_test_split

# 데이터 읽어 들이기
mr = pd.read_csv("mushroom.csv", header=None)

# label 분리
df = pd.DataFrame(mr.iloc[:, 0]) # 1 column만 선택하면 Series가 되므로 다시 dataframe으로 만듦

# 두번째 컬럼부터 마지막 컬럼까지 one-hot encoding하고 label에 붙임
df = df.join(pd.get_dummies(mr.iloc[:, 1:]))
print(df)
```

	0	1_b	1_c	1_f	1_k	1_s	1_x	2_f	2_g	2_s	...	21_s	21_v	21_y	\
0	p	0	0	0	0	0	1	0	0	1	...	1	0	0	
1	e	0	0	0	0	0	1	0	0	1	...	0	0	0	
2	e	1	0	0	0	0	0	0	0	1	...	0	0	0	
3	p	0	0	0	0	0	1	0	0	0	...	1	0	0	
4	e	0	0	0	0	0	1	0	0	1	...	0	0	0	
5	e	0	0	0	0	0	1	0	0	0		0	0	0	

학습 및 테스트

```
data = df.iloc[:, 1:]
label = df.loc[:, 0]

# 학습 전용 데이터와 테스트 전용 데이터로 나누기
data_train, data_test, label_train, label_test = train_test_split(data, label)

# 데이터 학습시키기
clf = RandomForestClassifier()
clf.fit(data_train, label_train)

# 데이터 예측하기
predict = clf.predict(data_test)

# 결과 테스트하기
result = pd.DataFrame({"label": label_test, "pre": predict})
print(result[0:10])

ac_score = metrics.accuracy_score(label_test, predict)
print("정답률 =", ac_score)
```

```
      label pre
1473      e   e
8103      e   e
1289      e   e
5275      p   p
5365      p   p
1320      e   e
5997      e   e
6451      p   p
7465      p   p
4504      p   p
정답률 = 1.0
```

MYONGJI UNIVERSITY

END