# AI를 이용한
# 식용 버섯 분류

2019108265 컴퓨터공학전공 양진석

# 목차

# 01 | 개요 및 필요성

# 개요 및 필요성


**매년 일어나는 독버섯 중독 사고**

**장마철 독버섯 급증**

**추석, 가을 산행길에 독버섯 노출**

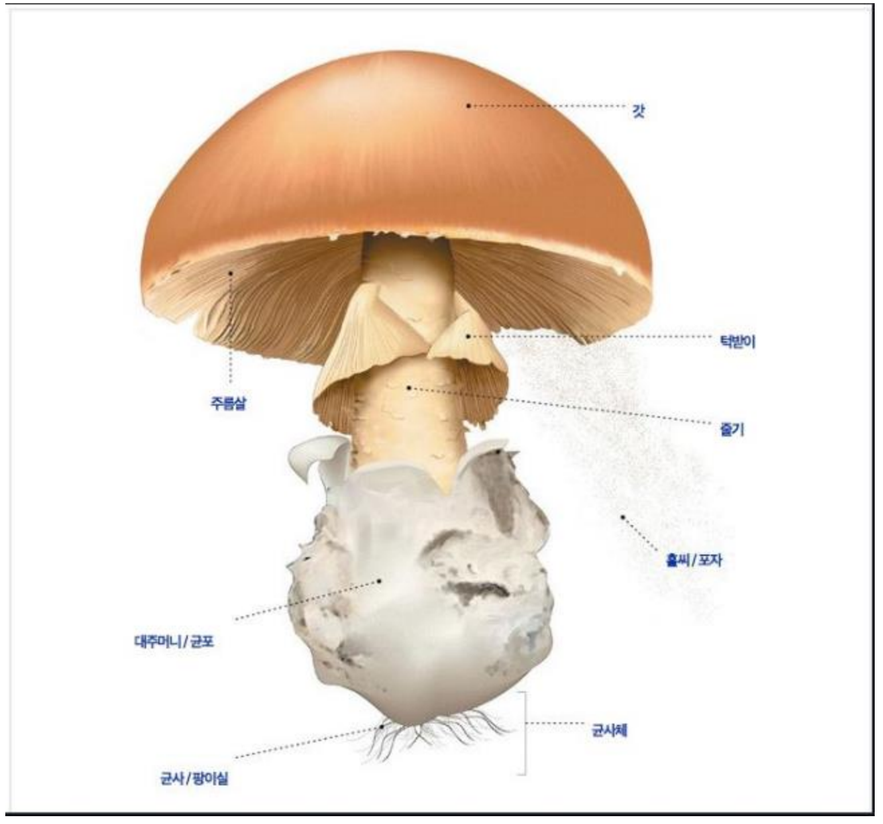
# 개요 및 필요성

## "AI로 독버섯 중독 막고 싶어요" 대구시교육청-MS, AI모델 해커톤 실시

10일 한국마이크로소프트에서 대구 지역 소프트웨어(SW)·인공지능(AI)융합 진로 탐색 학생 동아리 소속 중·고등학생 60여 명이 SW·AI융합 동아리 AI모델 해커톤에 참가했다. 대구시교육청 제공

# 02 | 데이터셋

# 데이터셋



# 데이터셋

| class | | cap-shape | | cap-surface | | cap-color | | bruises | |
|---|---|---|---|---|---|---|---|---|---|
| edible=e, poisonous=p | | bell=b,conical=c,convex=x,flat=f, knobbed=k,sunken=s | | fibrous=f,grooves=g,scaly=y,smooth=s | | brown=n,buff=b,cinnamon=c,gray=g,green=r,pink=p,purple=u,red=e,white=w,yellow=y | | bruises=t,no=f | |
| e | 52% | x | 45% | y | 40% | n | 28% | true | |
| p | 48% | f | 39% | s | 31% | g | 23% | 0 | 0% |
| | | Other (1316) | 16% | Other (2324) | 29% | Other (4000) | 49% | false | |
| | | | | | | | | 0 | 0% |

| odor | | gill-attachment | | gill-spacing | | gill-size | | gill-color | |
|---|---|---|---|---|---|---|---|---|---|
| almond=a,anise=l,creosote=c,fishy=y,foul=f,musty=m,none=n,pungent=p,spicy=s | | attached=a, descending=d, free=f, notched=n | | close=c,crowded=w,distant=d | | broad=b,narrow=n | | black=k,brown=n,buff=b, chocolate=h,gray=g, green=r,orange=o,pink=p, purple=u,red=e,white=w,yellow=y | |
| n | 43% | true | | c | 84% | b | 69% | b | 21% |
| f | 27% | 0 | 0% | w | 16% | n | 31% | p | 18% |
| Other (2436) | 30% | false | | | | | | Other (4904) | 60% |
| | | 0 | 0% | | | | | | |

# 데이터셋

class : 독성 유무

cap-shape : 머리 모양

cap-surface : 머리 표면

cap-color : 머리 색상

bruises : 버섯의 멍 유무

odor : 냄새

gill-attachment : 주름 부착 방식

gill-spacing : 주름 사이의 간격

gill-size : 주름 크기

gill-color : 주름 색상

stalk-shpae : 줄기 모양

stalk-root : 줄기의 뿌리 부분 모양

# 데이터셋

stalk-surface-above-ring : 턱받이 위의 줄기 표면

stalk-surface-below-ring : 턱받이 아래의 줄기 표면

stalk-color-above-ring : 턱받이 위의 줄기 색상

stalk-color-below-ring : 턱받이 아래의 줄기 색상

veil-type : 버섯의 부속물 종류

veil-color : 버섯 부속물의 색상

ring-number : 턱받이의 수

ring-type : 턱받이의 종류

spore-print-color : 포자 자국의 색상

population : 개체군의 분포 형태

habitat :  버섯이 자라는 환경

# 03 | 데이터분석

# 데이터분석

```
# 수학 및 과학 계산을 위한 라이브러리
import numpy as np
#데이터 조작 및 분석을 위한 라이브러리
import pandas as pd
#데이터 시각화 도구 중 하나로, 통계 그래픽 생성
import seaborn as sns
#시각화 도구 중 하나로, 그래프 및 차트를 생성
import matplotlib.pyplot as plt
```

주요 라이브러리 호출

# 데이터분석

```
df = pd.read_csv('../input/mushroom-classification/mushrooms.csv')
df.head()
```

| | class | cap-shape | cap-surface | cap-color | bruises | odor | gill-attachment | gill-spacing | gill-size | gill-color | ... | stalk-surface-below-ring | stalk-color-above-ring | stalk-color-below-ring | veil-type | veil-color | ring-number | ring-type | spore-print-color | population | habitat |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | p | x | s | n | t | p | f | c | n | k | ... | s | w | w | p | w | o | p | k | s | u |
| 1 | e | x | s | y | t | a | f | c | b | k | ... | s | w | w | p | w | o | p | n | n | g |
| 2 | e | b | s | w | t | l | f | c | b | n | ... | s | w | w | p | w | o | p | n | n | m |
| 3 | p | x | y | w | t | p | f | c | n | n | ... | s | w | w | p | w | o | p | k | s | u |
| 4 | e | x | s | g | f | n | f | w | b | k | ... | s | w | w | p | w | o | e | n | a | g |

5 rows × 23 columns

csv 파일을 읽어와 DataFrame으로 저장
head()함수를 통해 DataFrame의 처음 다섯 행을 표시

# 데이터분석

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8124 entries, 0 to 8123
Data columns (total 23 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   class                     8124 non-null   object
 1   cap-shape                 8124 non-null   object
 2   cap-surface               8124 non-null   object
 3   cap-color                 8124 non-null   object
 4   bruises                   8124 non-null   object
 5   odor                      8124 non-null   object
 6   gill-attachment           8124 non-null   object
 7   gill-spacing              8124 non-null   object
 8   gill-size                 8124 non-null   object
 9   gill-color                8124 non-null   object
 10  stalk-shape               8124 non-null   object
 11  stalk-root                8124 non-null   object
 12  stalk-surface-above-ring  8124 non-null   object
 13  stalk-surface-below-ring  8124 non-null   object
 14  stalk-color-above-ring    8124 non-null   object
 15  stalk-color-below-ring    8124 non-null   object
 16  veil-type                 8124 non-null   object
 17  veil-color                8124 non-null   object
 18  ring-number               8124 non-null   object
 19  ring-type                 8124 non-null   object
...
 21  population                8124 non-null   object
 22  habitat                   8124 non-null   object
dtypes: object(23)
memory usage: 1.4+ MB
```

데이터프레임은 8123개의 행과 6개의 열로 이루어짐

데이터타입을 보여줌과
동시에 결측지 유무를 보여줌

전체 데이터프레임이 약 1.4MB의 메모리는 차지

# 데이터분석

```
df['class'].replace(to_replace=['e','p'], value=['edible','poisonous'],inplace=True)
df['cap-shape'].replace(to_replace=['b', 'c','f','x','k','s'], value=['bell','conical','convex','flat','knobbed','sunken'],inplace=True)
df['cap-surface'].replace(to_replace=['f','g','y','s'], value=['fibrous','grooves','scaly','smooth'],inplace=True)
df['cap-color'].replace(to_replace=['n','b','c','g','r','p','u','e','w','y'], value=['brown','buff','cinnamon','gray','green','pink','purple','red','white','yellow'],inplace=True)
df['bruises'].replace(to_replace=['t', 'f'], value=['bruises','no'],inplace=True)
df['odor'].replace(to_replace=['a','l','c','y','f','m','n','p','s'], value=['almond','anise','creosote','fishy','foul','musty','none','pungent','spicy'],inplace=True)
df['gill-attachment'].replace(to_replace=['a','d','f','n'], value=['attached','descending','free','notched'],inplace=True)
df['gill-spacing'].replace(to_replace=['c','w','d'], value=['close','crowded','distant'],inplace=True)
df['gill-size'].replace(to_replace=['b', 'n'], value=['broad','narrow'],inplace=True)
df['gill-color'].replace(to_replace=['k','n','b','h','g','r','o','p','u','e','w','y'], value=['black','brown','buff','chocolate','gray','green','orange','pink','purple','red','white','yellow'],inplace=True)
df['stalk-shape'].replace(to_replace=['e', 't'], value=['enlarging','tapering'],inplace=True)
df['stalk-root'].replace(to_replace=['b','c','u','e','z','r','?'], value=['bulbous','club','cup','equal','rhizomorphs','rooted','missing'],inplace=True)
df['stalk-surface-above-ring'].replace(to_replace=['f','y','k','s'], value=['fibrous','scaly','silky','smooth'],inplace=True)
df['stalk-surface-below-ring'].replace(to_replace=['f','y','k','s'], value=['fibrous','scaly','silky','smooth'],inplace=True)
df['stalk-color-above-ring'].replace(to_replace=['n','b','c','g','o','p','e','w','y'], value=['brown','buff','cinnamon','gray','orange','pink','red','white','yellow'],inplace=True)
df['stalk-color-below-ring'].replace(to_replace=['n','b','c','g','o','p','e','w','y'], value=['brown','buff','cinnamon','gray','orange','pink','red','white','yellow'],inplace=True)
df['veil-type'].replace(to_replace=['p', 'u'], value=['partial','universal'],inplace=True)
df['veil-color'].replace(to_replace=['n','o','w','y'], value=['brown','orange','white','yellow'],inplace=True)
df['ring-number'].replace(to_replace=['n','o','t'], value=['none','one','two'],inplace=True)
df['ring-type'].replace(to_replace=['c','e','f','l','n','p','s','z'], value=['cobwebby','evanescent','flaring','large','none','pendant','sheathing','zone'],inplace=True)
df['spore-print-color'].replace(to_replace=['k','n','b','h','r','o','u','w','y'], value=['black','brown','buff','chocolate','green','orange','purple','white','yellow'],inplace=True)
df['population'].replace(to_replace=['a','c','n','s','v','y'], value=['abundant','clustered','numerous','scattered','several','solitary'],inplace=True)
df['habitat'].replace(to_replace=['g','l','m','p','u','w','d'], value=['grasses','leaves','meadows','paths','urban','waste','woods'],inplace=True)
```

## 데이터 값을 명확하게 표시

# 데이터분석

```
df.head()
```

| | class | cap-shape | cap-surface | cap-color | bruises | odor | gill-attachment | gill-spacing | gill-size | gill-color | ... | stalk-surface-below-ring | stalk-color-above-ring | stalk-color-below-ring | veil-type | veil-color | ring-number | ring-type | spore-print-color | population | habitat |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | poisonous | flat | smooth | brown | bruises | pungent | free | close | narrow | black | ... | smooth | white | white | partial | white | one | pendant | black | scattered | urban |
| 1 | edible | flat | smooth | yellow | bruises | almond | free | close | broad | black | ... | smooth | white | white | partial | white | one | pendant | brown | numerous | grasses |
| 2 | edible | bell | smooth | white | bruises | anise | free | close | broad | brown | ... | smooth | white | white | partial | white | one | pendant | brown | numerous | meadows |
| 3 | poisonous | flat | scaly | white | bruises | pungent | free | close | narrow | brown | ... | smooth | white | white | partial | white | one | pendant | black | scattered | urban |
| 4 | edible | flat | smooth | gray | no | none | free | crowded | broad | black | ... | smooth | white | white | partial | white | one | evanescent | brown | abundant | grasses |

5 rows × 23 columns

# 데이터분석

```
df.describe()
```

| | class | cap-shape | cap-surface | cap-color | bruises | odor | gill-attachment | gill-spacing | gill-size | gill-color | ... | stalk-surface-below-ring | stalk-color-above-ring | stalk-color-below-ring | veil-type | veil-color | ring-number | ring-type | spore-print-color | population | habitat |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | ... | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 | 8124 |
| unique | 2 | 6 | 4 | 10 | 2 | 9 | 2 | 2 | 2 | 12 | ... | 4 | 9 | 9 | 1 | 4 | 3 | 5 | 9 | 6 | 7 |
| top | edible | flat | scaly | brown | no | none | free | close | broad | buff | ... | smooth | white | white | partial | white | one | pendant | white | several | woods |
| freq | 4208 | 3656 | 3244 | 2284 | 4748 | 3528 | 7914 | 6812 | 5612 | 1728 | ... | 4936 | 4464 | 4384 | 8124 | 7924 | 7488 | 3968 | 2388 | 4040 | 3148 |

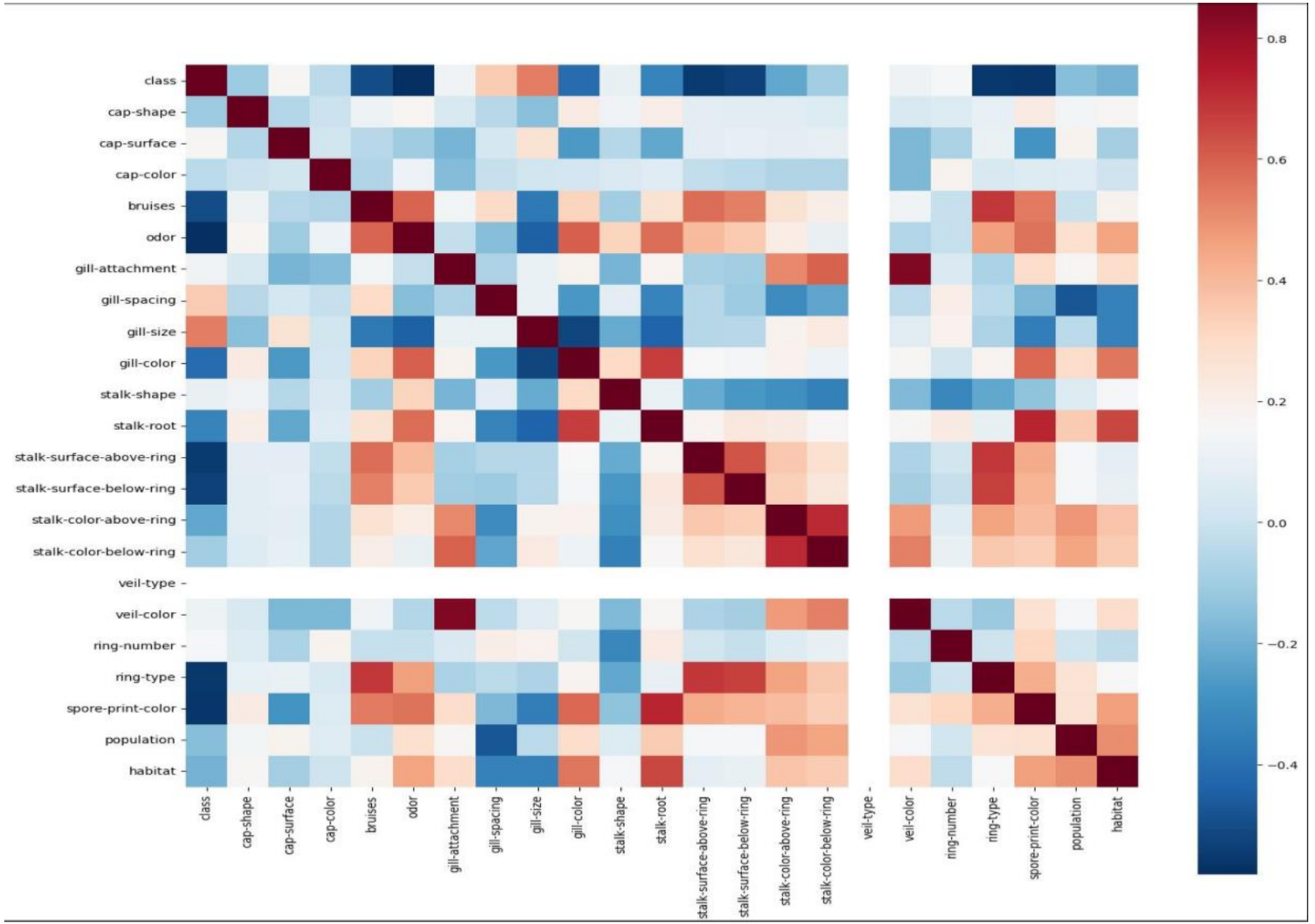
4 rows × 23 columns

count - 해당 열의 총 행 수
unique - 고유값의 수 (중복 X)
top - 최빈값
freq - 최빈값의 빈도

# 데이터분석

```
corr = df.apply(lambda x : pd.factorize(x)[0]).corr(method='pearson', min_periods=1)
# 프로팅할 히트맵의 크기
plt.figure(figsize=(16,16))
#히트맵으로 시각화
sns.heatmap(corr, cmap = "RdBu_r", vmax=0.9, square=True) # 값이 0.9 초과시 색상이 진해짐, 모양은 정사각형
```

# 데이터분석


class
cap-shape
cap-surface
cap-color
bruises
odor
gill-attachment
gill-spacing
gill-size
gill-color
stalk-shape
stalk-root
stalk-surface-above-ring
stalk-surface-below-ring
stalk-color-above-ring
stalk-color-below-ring
veil-type
veil-color
ring-number
ring-type
spore-print-color
population
habitat
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

# 데이터분석

버섯 부속물의 종류는 모두 하얀색??

모든 값이 partial(불완전)로 동일

# 데이터분석

gill-size - 주름 크기
gill-spacing - 주름 사이의 간격
cap-surface - 머리 표면
ring-number - 턱받이의 수
gill-attachment - 주름 부착 방식
veil-color - 버섯 부속물의 색상
stalk-shape - 줄기 모양

# 데이터분석

```
IF = corr['class'].sort_values(ascending=False).head(10).to_frame()
IF.head(8)
```

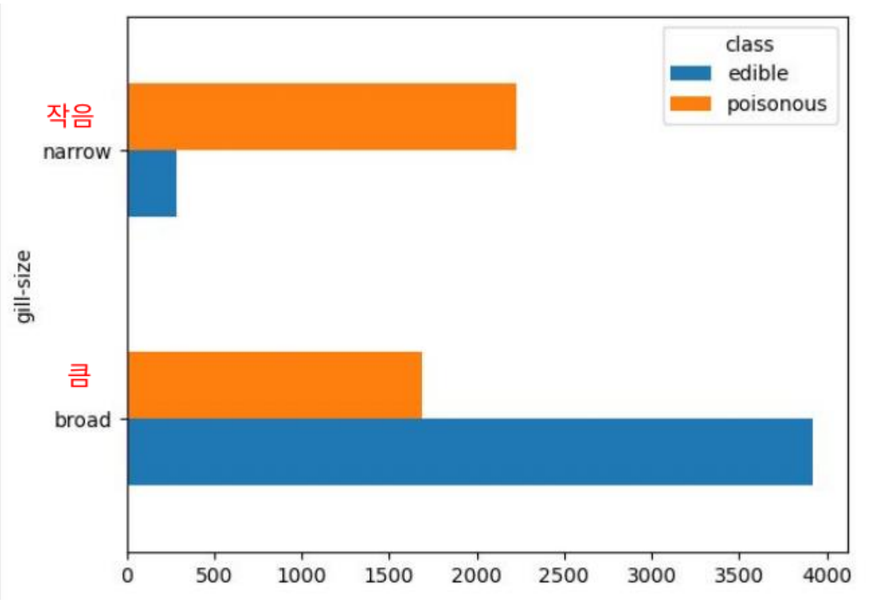
| | class |
|---|---|
| class | 1.000000 |
| gill-size | 0.540024 |
| gill-spacing | 0.348387 |
| cap-surface | 0.169663 |
| ring-number | 0.152261 |
| gill-attachment | 0.129200 |
| veil-color | 0.120766 |
| stalk-shape | 0.102019 |

class와 상관관계가 큰 순서대로
상위 8개의 데이터프레임을 출력

# 데이터분석(주름 크기)

```
print(df.groupby('gill-size')['class'].value_counts())
df.groupby('gill-size')['class'].value_counts().unstack().plot.barh()
```

```
gill-size  class
broad      edible       3920
           poisonous    1692
narrow     poisonous    2224
           edible        288
Name: count, dtype: int64
```

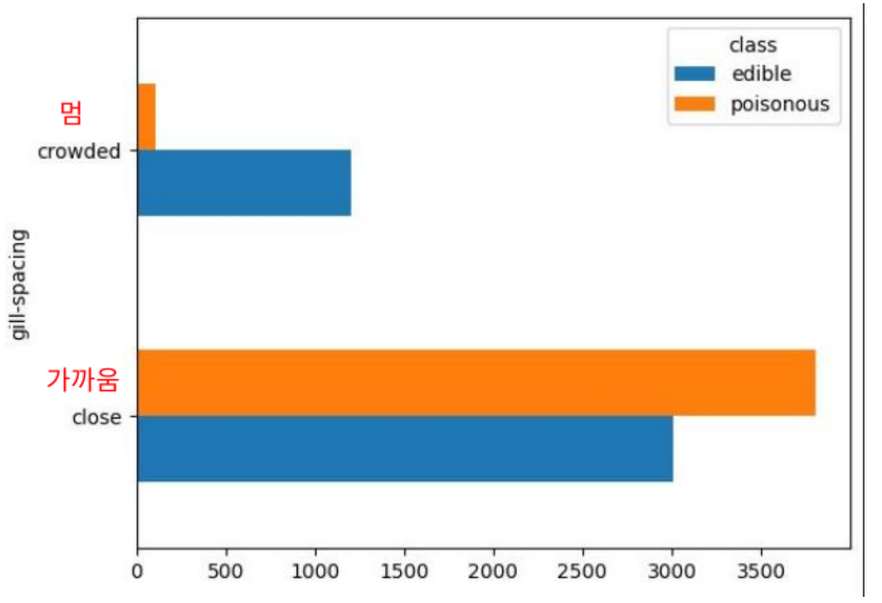



주름 크기가 작을수록 독성
크기가 클수록 식용이 더 많음

# 데이터분석(주름 사이 간격)

```
print(df.groupby('gill-spacing')['class'].value_counts())
df.groupby('gill-spacing')['class'].value_counts().unstack().plot.barh()
```

```
gill-spacing  class
close         poisonous    3804
              edible       3008
crowded       edible       1200
              poisonous     112
Name: count, dtype: int64

<Axes: ylabel='gill-spacing'>
```

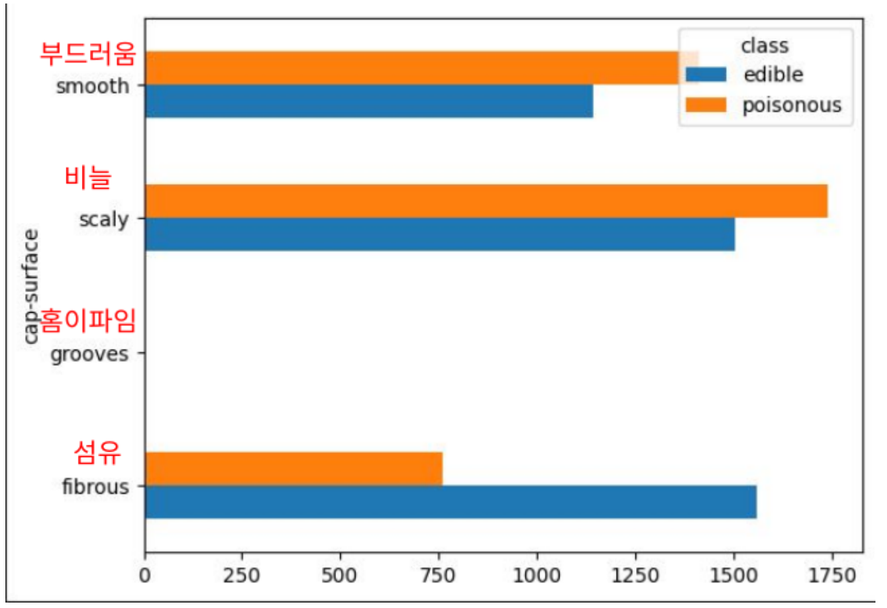



주름 사이 간격이 멀수록 식용
가까울수록 독성이 더 많음

# 데이터분석(머리 표면)

```
print(df.groupby('cap-surface')['class'].value_counts())
df.groupby('cap-surface')['class'].value_counts().unstack().plot.barh()
```

```
cap-surface  class
fibrous      edible       1560
             poisonous     760
grooves      poisonous       4
scaly        poisonous    1740
             edible       1504
smooth       poisonous    1412
             edible       1144
Name: count, dtype: int64
```

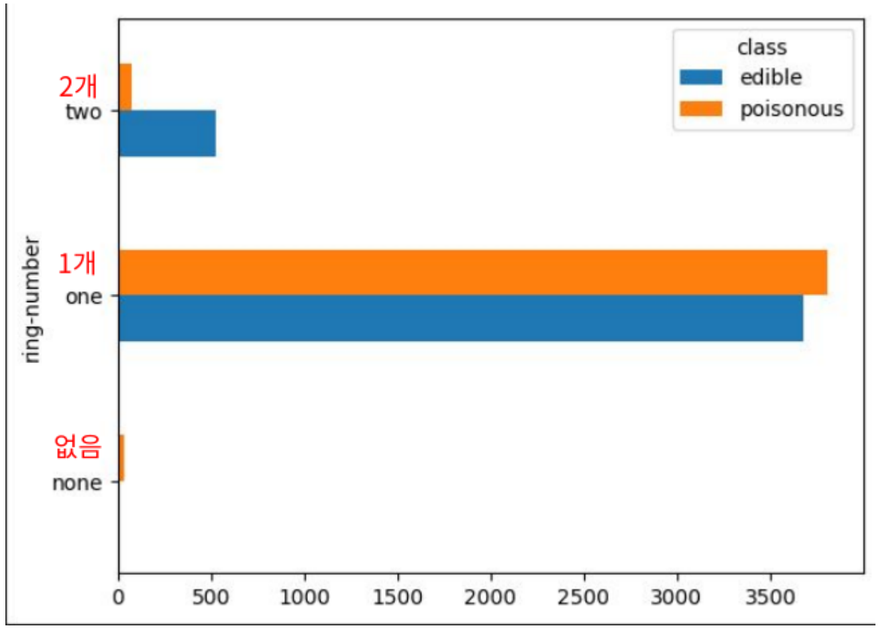



**표면이 부드럽거네 비늘이 있을수록 독성이 많음**

# 데이터분석(턱받이 수)

```
print(df.groupby('ring-number')['class'].value_counts())
df.groupby('ring-number')['class'].value_counts().unstack().plot.barh()
```

```
ring-number  class
none         poisonous      36
one          poisonous    3808
             edible       3680
two          edible        528
             poisonous      72
Name: count, dtype: int64
```

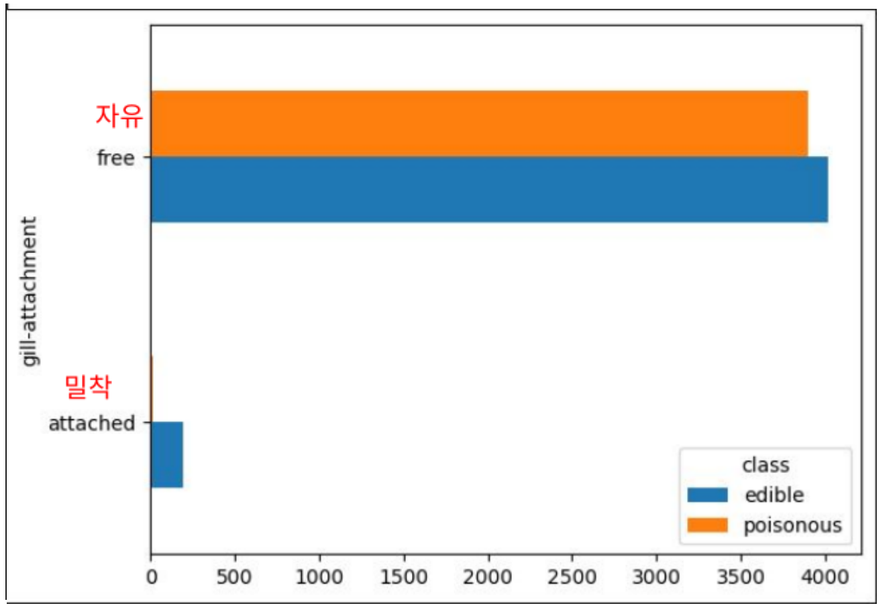



턱받이 수가 1개이면 독성과 식용 버섯의 수가 균일하지만, 턱받이 수가 2개일땐 식용 버섯의 수가 많음

# 데이터분석(주름 부착 방식)

```
print(df.groupby('gill-attachment')['class'].value_counts())
df.groupby('gill-attachment')['class'].value_counts().unstack().plot.barh()
```

```
gill-attachment  class
attached         edible        192
                 poisonous      18
free             edible        4016
                 poisonous     3898
Name: count, dtype: int64
```

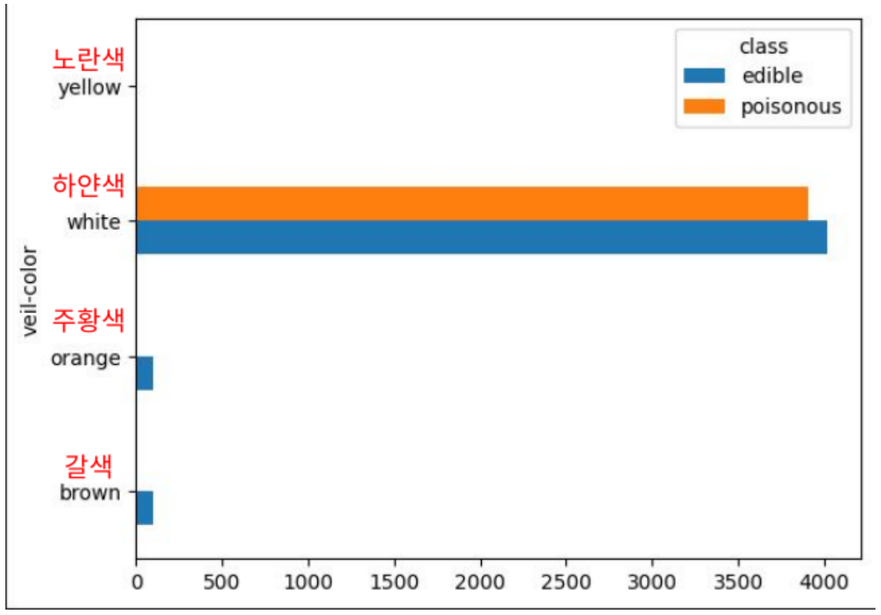



주름 부착 방식이 자유로울 경우 독성과 식용이 균일한 반면, 밀착돼 있을 경우 식용이 더 많음

# 데이터분석(부속물의 색상)

```
print(df.groupby('veil-color')['class'].value_counts())
df.groupby('veil-color')['class'].value_counts().unstack().plot.barh()
```

```
veil-color  class
brown       edible          96
orange      edible          96
white       edible        4016
            poisonous     3908
yellow      poisonous        8
Name: count, dtype: int64
```

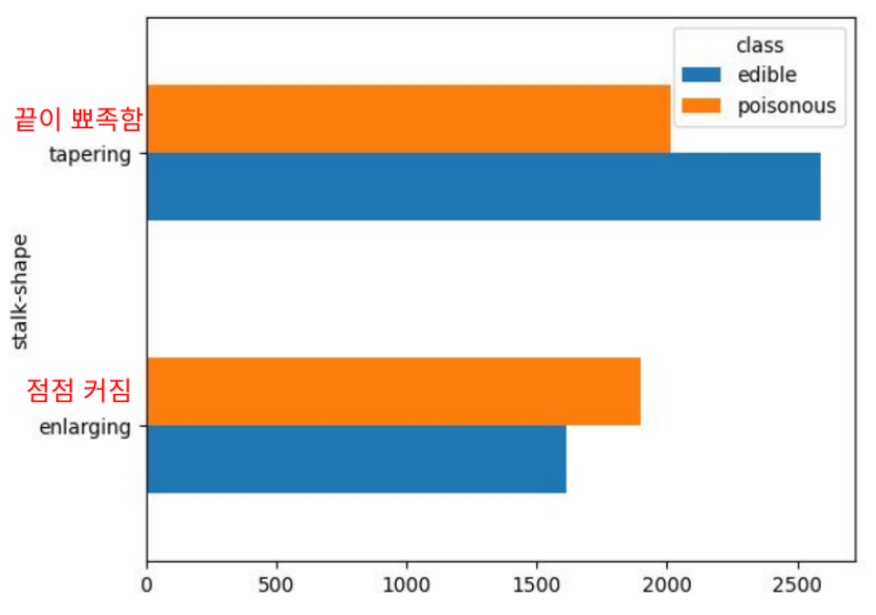



**전체적으로 독성과**
**식용의 편차가 균일하다**

# 데이터분석(줄기 모양)

```
print(df.groupby('stalk-shape')['class'].value_counts())
df.groupby('stalk-shape')['class'].value_counts().unstack().plot.barh()
```

```
stalk-shape  class
enlarging    poisonous    1900
             edible       1616
tapering     edible       2592
             poisonous    2016
Name: count, dtype: int64
```




줄기의 끝이 뾰족할 경우 식용이 많으며
줄기의 끝이 점점 커질 경우 독성이 많다

# 04 | 모델 학습

# 모델 학습

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

scikit-learn 라이브러리에서 제공되는 기능 사용

train_test_split : 데이터셋을 훈련세트와 테스트 세트로 무작위 분할, 모델의 일반화 성능을 평가

accuracy_score : 모델의 예측과 실제 타겟 값 간의 정확도를 계산

# 모델 학습

```
## Feature engineering

df['class'].replace(to_replace=['edible','poisonous'], value=['0','1'],inplace=True)
df.head()
```

| | class | cap-shape | cap-surface | cap-color | bruises | odor | gill-attachment | gill-spacing | gill-size | gill-color | ... | stalk-surface-below-ring | stalk-color-above-ring | stalk-color-below-ring | veil-type | veil-color | ring-number | ring-type | spore-print-color | population | habitat |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | flat | smooth | brown | bruises | pungent | free | close | narrow | black | ... | smooth | white | white | partial | white | one | pendant | black | scattered | urban |
| 1 | 0 | flat | smooth | yellow | bruises | almond | free | close | broad | black | ... | smooth | white | white | partial | white | one | pendant | brown | numerous | grasses |
| 2 | 0 | bell | smooth | white | bruises | anise | free | close | broad | brown | ... | smooth | white | white | partial | white | one | pendant | brown | numerous | meadows |
| 3 | 1 | flat | scaly | white | bruises | pungent | free | close | narrow | brown | ... | smooth | white | white | partial | white | one | pendant | black | scattered | urban |
| 4 | 0 | flat | smooth | gray | no | none | free | crowded | broad | black | ... | smooth | white | white | partial | white | one | evanescent | brown | abundant | grasses |

5 rows × 23 columns

데이터프레임의 'class' 열에서 'edible'을 '0'으로, 'poisonous'를 '1'로 대체

모델 훈련 시 더 적합한 형태로 종속 변수를 사용할 수 있음

# 모델 학습

```
# Split the data

X = df.drop('class', axis=1)
y = df['class']
```

데이터프레임 변수 할당

'X' : 독립 변수 데이터프레임, 'class'열을 제외한 모든 열의 데이터 포함
'Y' : 종속 변수 시리즈, 'class'열의 데이터 포함

모델을 훈련할 때 독립 변수와 종속 변수를 각각 사용할 수 있게 됨

# 모델 학습

```
# handling categorical features

X = pd.get_dummies(X)
X.head()
```

| | cap-shape_bell | cap-shape_conical | cap-shape_convex | cap-shape_flat | cap-shape_knobbed | cap-shape_sunken | cap-surface_fibrous | cap-surface_grooves | cap-surface_scaly | cap-surface_smooth | ... | population_scattered | population_several | population_solitary | habitat_grasses | habitat_leaves |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | False | False | False | True | False | False | False | False | False | True | ... | True | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | True | ... | False | False | False | True | False |
| 2 | True | False | False | False | False | False | False | False | False | True | ... | False | False | False | False | False |
| 3 | False | False | False | True | False | False | False | False | True | False | ... | True | False | False | False | False |
| 4 | False | False | False | True | False | False | False | False | False | True | ... | False | False | False | True | False |

5 rows × 117 columns

데이터프레임 X의 특성을 원-핫 인코딩하여 더미 변수로 변환하는 작업

원-핫 인코딩 : 표현하고 싶은 단어의 인덱스에 1의 값을 부여하고 다른 인덱스에는 0을 부여하는 표현방식
더미 변수 : 어떤 속성이 존재할 경우 값을 1, 존재하지 않을 경우 값을 0으로 코딩한 변수

각 범주형 변수에 대한 더미 변수가 추가된 새로운 데이터프레임 생성
머신러닝 모델에서 범주형 변수를 처리하는데 유용

# 모델 학습

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=44)
```

데이터셋인 x와 y를 훈련 세트와 테스트 세트로 분할

# 05 | 모델 분석 및 비교

# 모델 분석 및 비교

## RandomForest

```
from sklearn.ensemble import RandomForestClassifier

# Fitting Random Forest Classification
classifier = RandomForestClassifier(n_estimators = 200)
classifier.fit(X_train, y_train)

# predict
RF_pred = classifier.predict(X_test)
accuracy = accuracy_score(y_test, RF_pred)
accuracy
```

```
1.0
```

트리의 개수가 200개인 랜덤 포레스트 모델 생성

훈련 데이터를 사용하여 모델 학습

테스트 데이터에 대한 예측 수행

실제 타겟값과 모델의 예측값은 100퍼센트로 정확함

# 모델 분석 및 비교

## SGDClassifier

```
from sklearn.linear_model import SGDClassifier

# fit to SGDClassifier
sgd= SGDClassifier()
sgd.fit(X_train, y_train)

# predict
SGD_pred = sgd.predict(X_test)
acc = accuracy_score(y_test, SGD_pred)
print(acc)
```

```
0.999507631708518
```

확률적 경사 하강법을 사용하는 선형 분류기 생성

훈련 데이터를 사용하여 모델 학습

테스트 데이터에 대한 예측 수행

실제 타겟값과 모델의 예측값은 약 99퍼센트로 정확함

# 모델 분석 및 비교

## LogisticRegression

```
from sklearn.linear_model import LogisticRegression

#fit to LogReg
lr = LogisticRegression()
lr.fit(X_train, y_train)

# Predict
LR_pred = lr.predict(X_test)
acc = accuracy_score(y_test, LR_pred)
print(acc)
```

```
1.0
```

로지스틱 회귀 모델 생성

훈련 데이터를 사용하여 모델 학습

테스트 데이터에 대한 예측 수행

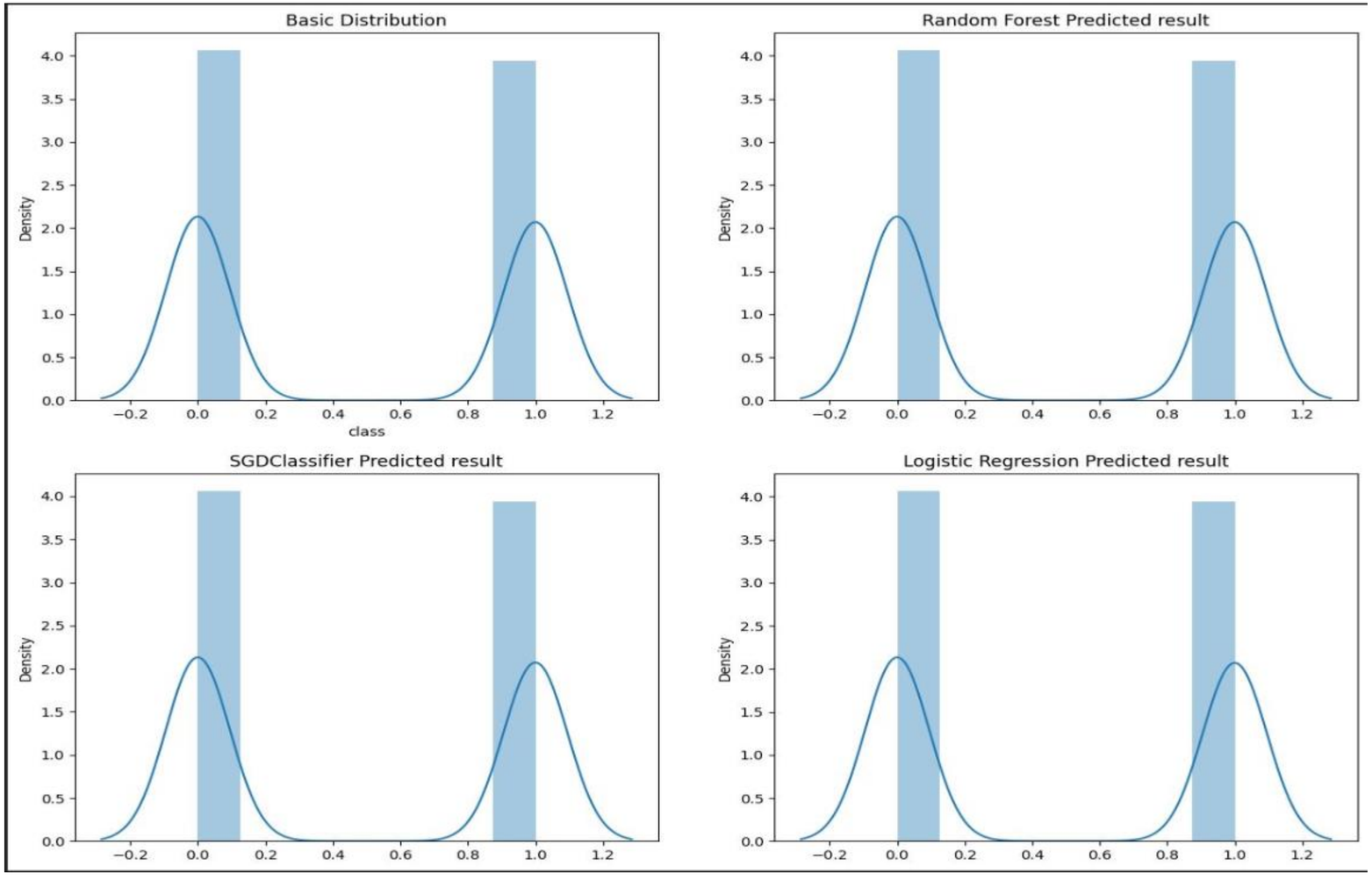
실제 타겟값과 모델의 예측값은 100퍼센트로 정확함

# 모델 분석 및 비교

```
## Distribution Comparison

f = plt.figure(figsize=(15,12))

# Basic Distribution
ax = f.add_subplot(221)
ax = sns.distplot(y_test)
ax.set_title('Basic Distribution')

# Random Forest Predicted result
ax = f.add_subplot(222)
xx = pd.DataFrame(RF_pred)
ax = sns.distplot(RF_pred, label="Predicted Values")
ax.set_title('Random Forest Predicted result')

# SGDClassifier Predicted result
ax = f.add_subplot(223)
ax = sns.distplot(SGD_pred, label="Predicted Values")
ax.set_title('SGDClassifier Predicted result')

# Logistic Regression Predicted result
ax = f.add_subplot(224)
ax = sns.distplot(LR_pred, label="Predicted Values")
ax.set_title('Logistic Regression Predicted result')
```

테스트 세트와 각 모델의 예측 결과에 대한 분포를 그래프로 시각화

# 모델 분석 및 비교


Basic Distribution
Random Forest Predicted result
SGDClassifier Predicted result
Logistic Regression Predicted result
Density
class
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2

# 06 | 결론

# 결론

테스트의 정확성을 바탕으로 데이터분석 결과를 정리해 보았을때

주름크기 : 주름이 큰 경우 '식용' 비율이 높고, 좁을 경우 '독성' 비율이 높다

주름 사이 간격 : 간격이 멀 수록 '식용' 비율이 높고, 좁을 수록 '독성' 비율이 높다

머리 표면 : 부드럽거나 비늘이 있는 경우에는 '독성' 비율이 '식용' 비율보다 높다.

턱받이 수 : 2개일 때 '식용' 비율이 높고, '1개'일 때 '독성' 비율이 높다

주름 부착 방식 : 자유로운 부착 방식일 때 '독성'와 '식용'의 분포가 비슷하며, 밀착되어 있을 경우 '식용' 비율이 낮다

부속물 색상 : 전체적으로 '식용'과 '독성'의 편차가 균일하다

줄기 모양 : 모양이 점점 커질 경우 '독성' 비율이 높고, 끝이 뾰족할 경우 '식용' 비율이 높다.

# 감사합니다