

# 버섯의 물리적 특성을 이용한 식용 / 독성 분류

---

버섯의 갓 직경, 모양, 색상 등 다양한 물리적  
특성을 기반으로 해당 버섯이  
식용(e)인지 독성(p)인지 분류합니다.



01.

프로젝트 개요

02.

탐색적 데이터 분석

03.

데이터 전처리

04.

모델링 및 하이퍼파라미터 최적화

05.

최적화 된 모델 학습 및 검증

06.

결과 분석 및 해석

# OUTLINE

## 프로젝트 개요



# Binary Prediction of Poisonous Mushrooms

Playground Series - Season 4, Episode 8

<https://www.kaggle.com/competitions/playground-series-s4e8>



UCI Machine Learning Repository의 Mushroom 데이터셋을 활용한 버섯 분류

- 버섯의 물리적 특성을 기반으로 식용 또는 독성 여부를 분류함
- 이 데이터셋은 8,124개의 샘플과 22개의 특성을 가진 다변량, 범주형 데이터로 구성

데이터셋 설명

이 데이터셋은 Agaricus 및 Lepiota 과(Family)에 속하는 23종의 주름버섯에 대한 가상의 표본 설명을 포함합니다. 각 종은 다음과 같이 분류됩니다:

- \* 확실히 식용 가능
- \* 확실히 독성
- \* 식용 여부 불명 및 비관장 (이 분류는 독성 분류와 통합됨)

# EDA

# 탐색적 데이터분석



## 버섯의 주요 특성 요약

### 줄기(Stem)

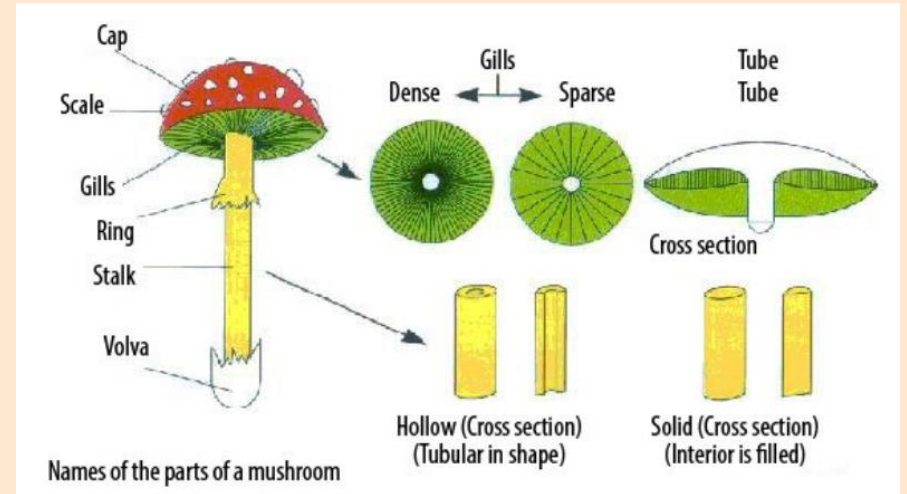
- Stem Height(줄기 높이) : 지면에서 갓 까지의 길이
- Stem Width(줄기 너비) : 직경(좁음, 중간, 넓은)
- Stem Surface(표면) : 매끄러움, 섬유질, 비늘 모양 등
- Stem Root(줄기) : 구근 모양, 뿔죽함 등
- Stem Color(줄기 색상) : 균일하거나, 길이에 따라 다양성을 보임

### 주름살(gill)

- Gill Attachment(주름살 부착) : 자유형, 부착형, 내린형 등
- Gill Spacing(주름살 간격) : 조밀, 간격 있음, 중간
- Gill Color(주름살 색상) : 종 구별에 중요, 나이에 따라 변화

### 갓(Cap)

- Cap Diameter(갓 직경) : 크기 측정 (mm~cm)
- Cap Shape(갓 모양) : 원뿔형, 종 모양, 평평한 형태 등
- Cap Surface(갓 표면) : 매끄러움, 비늘 모양, 끈적거림 등
- Cap Color(갓 색상) : 다양하며 성숙도에 따라 변화 가능



### 베일(veil) & 고리(ring)

- Veil Type(베일 종류) : 부분 베일, 전체 베일
- Has Ring(고리 유무) : 있음 또는 없음
- Ring Type(고리 종류) : 단일, 이중, 벌어짐 등

### 기타

- Does Bruise or Bleed(멍들거나 출혈) : 색 변화 또는 유색 액체 방출
- Spore Print Color(포자 흔적 색상) : 중요한 식별 특징
- Habitat(서식지) : 삼림지대, 초원, 도시 지역 등
- Season(계절) : 나타나는 시기

전반적인 데이터 확인하기

÷ 피쳐 ÷	÷ 타입 ÷	123 결측값 ÷	123 유니크값 ÷	첫번째값 ÷	두번째값 ÷	세번째값 ÷
0 id	int64	0	3116945	0	1	2
1 class	object	0	2	e	p	e
2 cap-diameter	float64	4	3913	8.8	4.51	6.94
3 cap-shape	object	40	74	f	x	f
4 cap-surface	object	671023	83	s	h	s
5 cap-color	object	12	78	u	o	b
6 does-bruise-or-bleed	object	8	26	f	f	f
7 gill-attachment	object	523936	78	a	a	x
8 gill-spacing	object	1258435	48	c	c	c
9 gill-color	object	57	63	w	n	w
10 stem-height	float64	0	2749	4.51	4.79	6.85
11 stem-width	float64	0	5836	15.39	6.48	9.93
12 stem-root	object	2757023	38	NaN	NaN	NaN
13 stem-surface	object	1980861	60	NaN	y	s
14 stem-color	object	38	59	w	o	n
15 veil-type	object	2957493	22	NaN	NaN	NaN
16 veil-color	object	2740947	24	NaN	NaN	NaN
17 has-ring	object	24	23	f	t	f
18 ring-type	object	128880	40	f	z	f
19 spore-print-color	object	2849682	32	NaN	NaN	NaN
20 habitat	object	45	52	d	d	l
21 season	object	0	4	a	w	w

- 피쳐의 타입을 보면 3가지(int, object, float)로 분류되는 모습을 볼 수 있는데 이는 범주형과 연속형으로 나눌 수 있어 보입니다.
- 결측값과 유니크값의 범위들이 꽤나 차이가 나는 모습을 볼 수 있는 것으로 보아 범위를 제한할 필요가 있어 보입니다.
- 샘플링 된 값들은 적절한 인코딩을 할 필요가 있어 보입니다.

23종의 주름버섯에 대한 가상의 표본은 식용/독성 두가지 클래스로 분류되어 있습니다.

Class P	Class E
1,705,396	1,411,549
3,116,945	
Total Class	

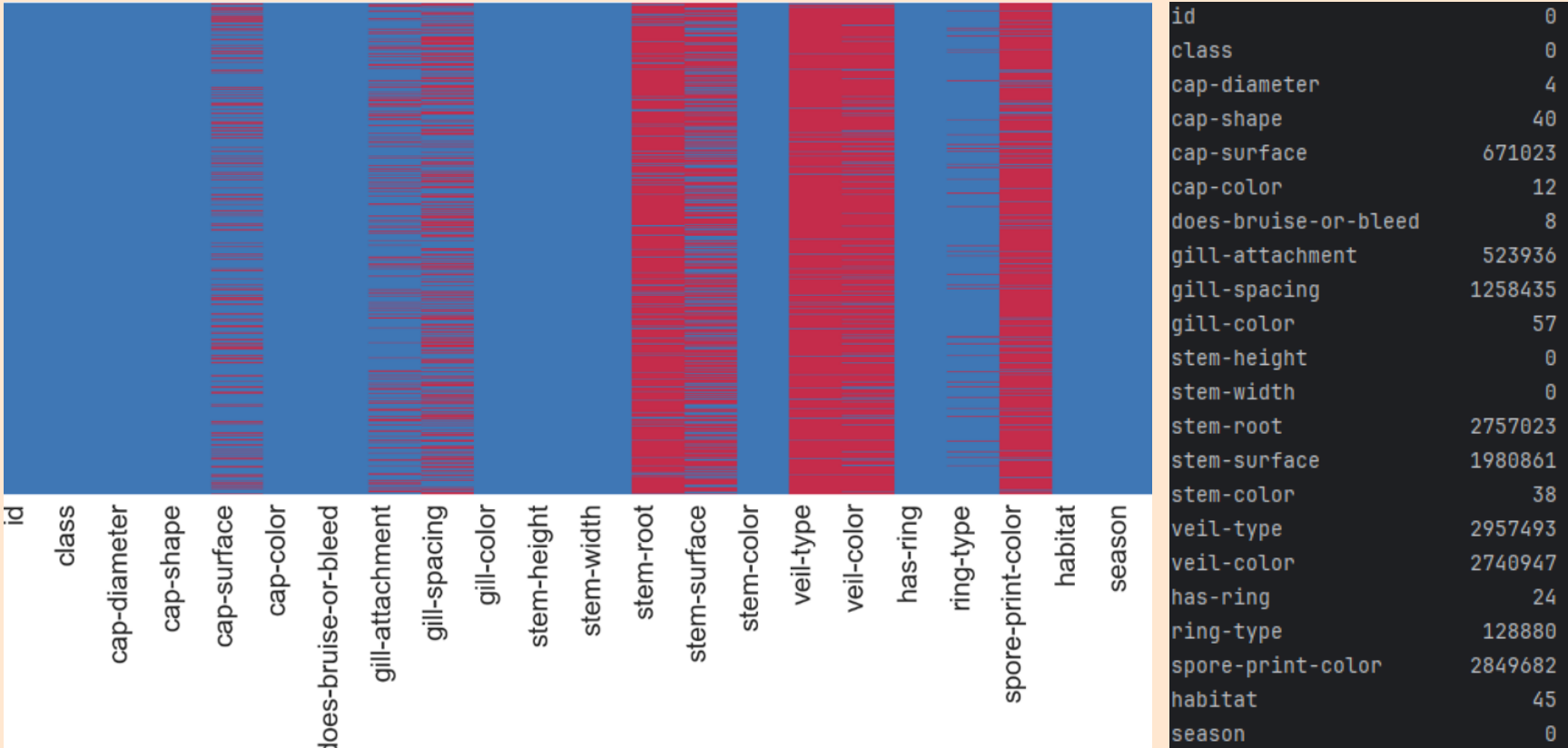
Train File



02. 탐색적 데이터 분석

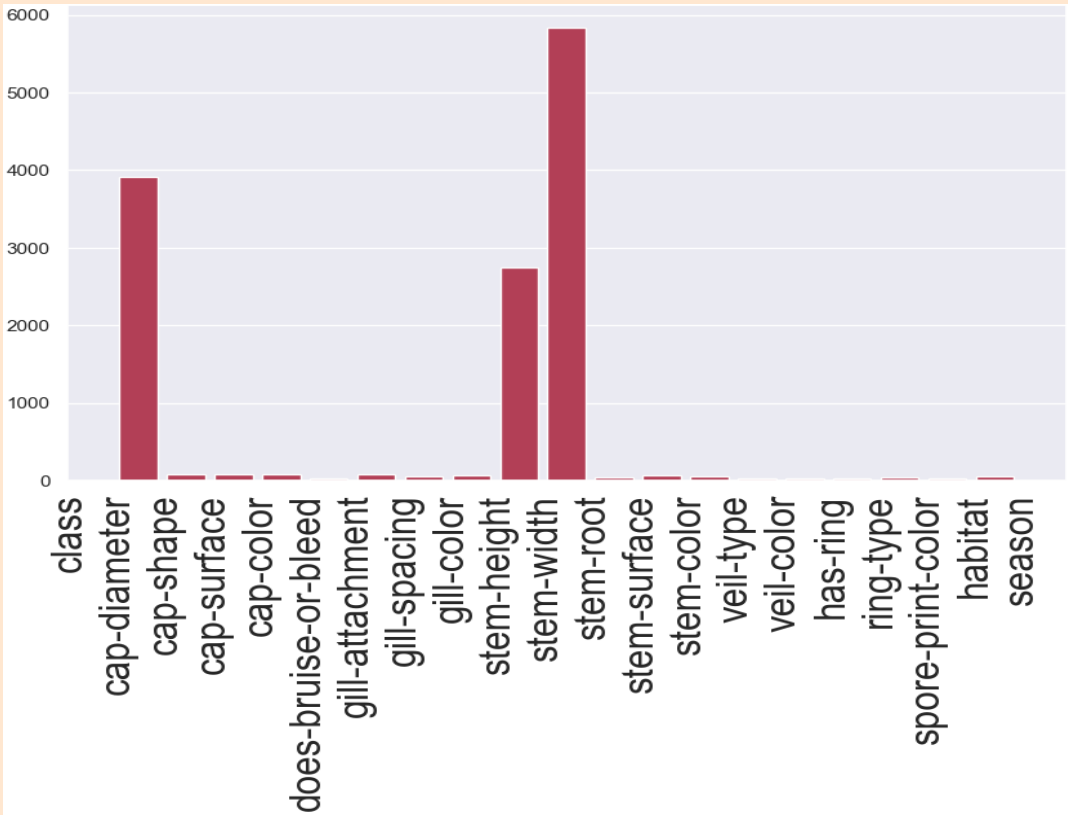
데이터 톨아보기

8개의 피처에서 결측치가 존재합니다.



# 유니크 값(\*id 컬럼 제외)

유니크 값의 전체적인 범위가 꽤나 차이가 나는 모습을 보이고 있습니다.



피쳐	유니크값
class	2
cap-diameter	3913
cap-shape	74
cap-surface	83
cap-color	78
does-bruise-or-bleed	26
gill-attachment	78
gill-spacing	48
gill-color	63
stem-height	2749
stem-width	5836
stem-root	38
stem-surface	60
stem-color	59
veil-type	22
veil-color	24
has-ring	23
ring-type	40
spore-print-color	32
habitat	52
season	4

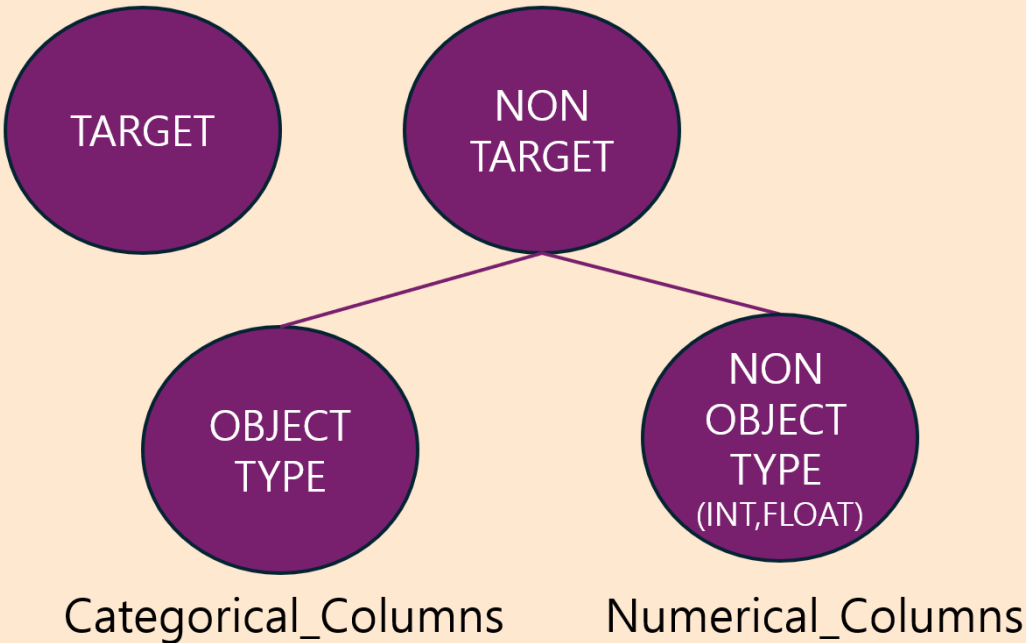
# PREPROCESSING

## 데이터 전처리



# 변수 데이터 분리하기

타겟열과 타겟열을 제외한 열들의 오브젝트 타입과 비 오브젝트 타입으로 데이터 분리



## Categorical\_Columns

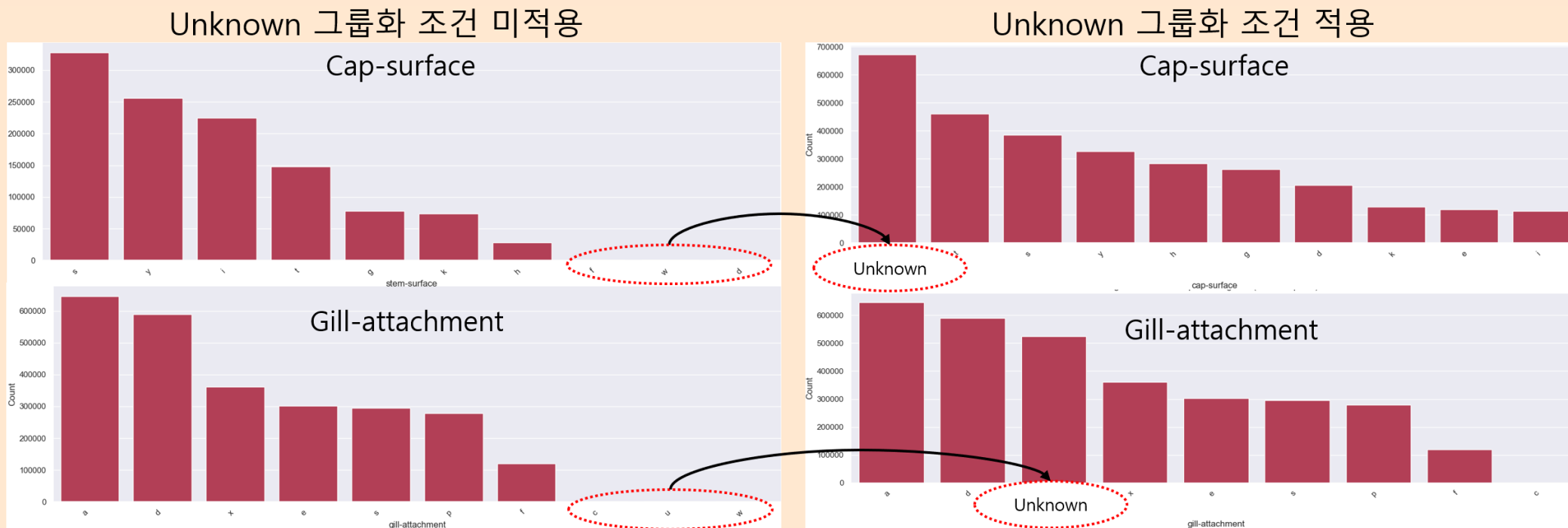
- 'cap-shape'
- 'cap-surface'
- 'cap-color'
- 'does-bruise-or-bleed'
- 'gill-attachment'
- 'gill-spacing'
- 'gill-color'
- 'stem-root'
- 'stem-surface'
- 'stem-color'
- 'veil-type'
- 'veil-color'
- 'has-ring'
- 'ring-type'
- 'spore-print-color'
- 'habitat'
- 'season'

## Numerical\_Columns

- 'cap-diameter'
- 'stem-height'
- 'stem-width'

## 범주형 변수 처리하기

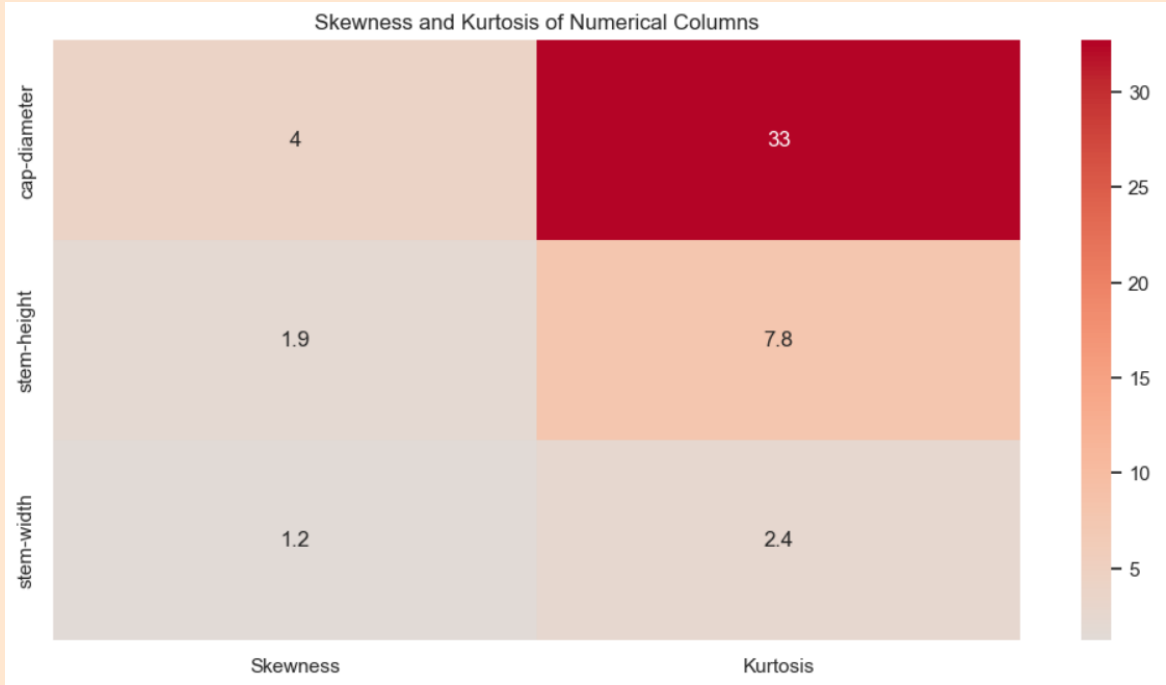
범주형 변수의 카디널리티에 기반하여 빈도수가 낮은 카테고리들을 특정 조건에 의해 새로운 범주 'UNKNOWN'으로 그룹화 처리



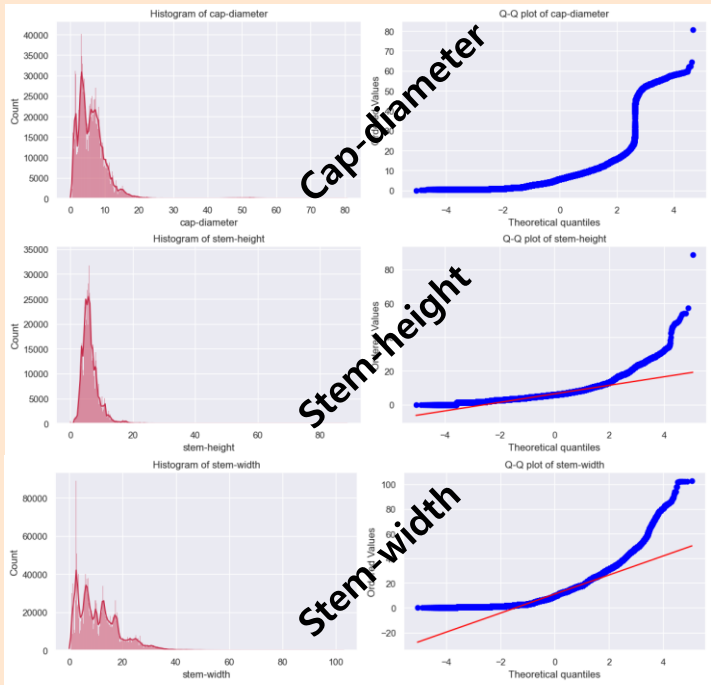
### 03. 데이터 전처리

## 연속형 변수 처리하기

왜도-첨도 분석, 히스토그램, QQ-plot을 통한 연속형 변수의 분포 특성 분석 및 정규성 검토 결과 정규성과 극단적인 값 존재



모든 특성이 높은 비대칭(왜도>1)과 뾰족한 분포(첨도>3)을 보이며, 특히 cap-diameter는 극단적(왜도4.0, 첨도33.0)이며 정규성 가정이 크게 위배 될 가능성을 나타냅니다.



QQ-plot: 직선에서 벗어난 정도가 클수록 정규 분포에서 벗어남을 나타냅니다. 특히 cap-diameter의 경우 큰 값에서 정규성에서 크게 벗어나는 것을 볼 수 있습니다.

히스토그램: 모든 변수가 오른쪽으로 치우친(right-skewed) 분포를 보이며, 특히 cap-diameter의 경우 극단적인 값들이 많이 관찰됩니다.

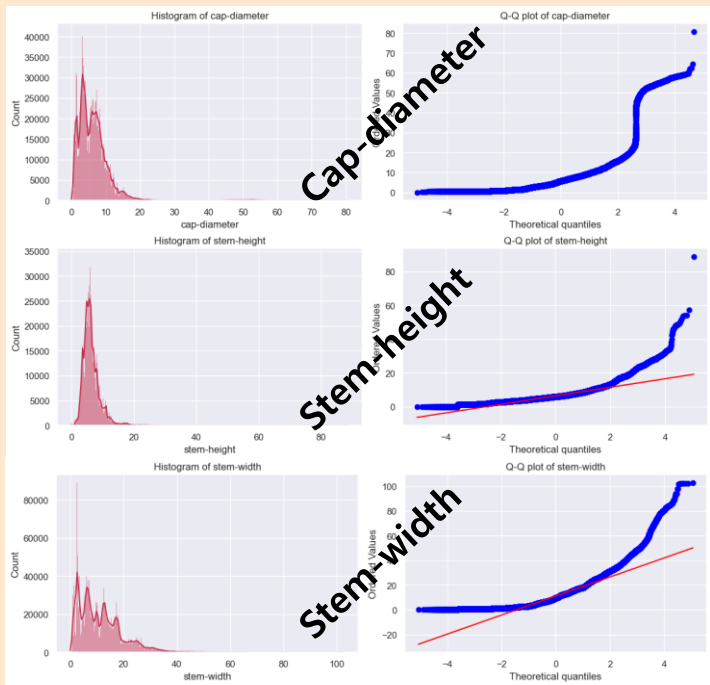
첨도(Kurtosis):

- 3에 가까우면 정규 분포와 유사한 뾰족함
- 3보다 크면 정규 분포보다 뾰족함 (heavy-tailed)
- 3보다 작으면 정규 분포보다 평평함 (light-tailed)

왜도(Skewness):

- |왜도| < 0.5: 거의 대칭
- 0.5 < |왜도| < 1: 중간 정도의 비대칭
- |왜도| > 1: 매우 비대칭

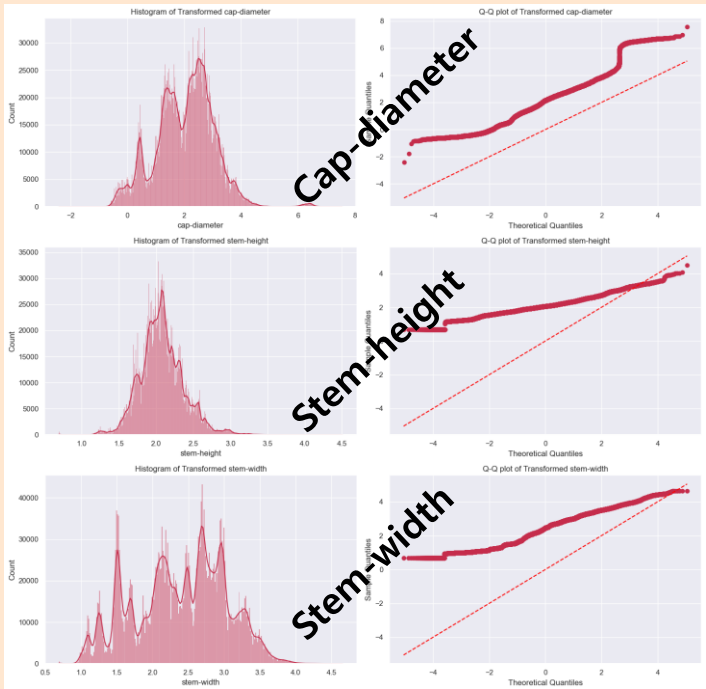
Stem-height, Stem-width는 로그 변환, Cap-diameter는 Box-Cox 변환



Box-Cox 변환

로그 변환

로그 변환

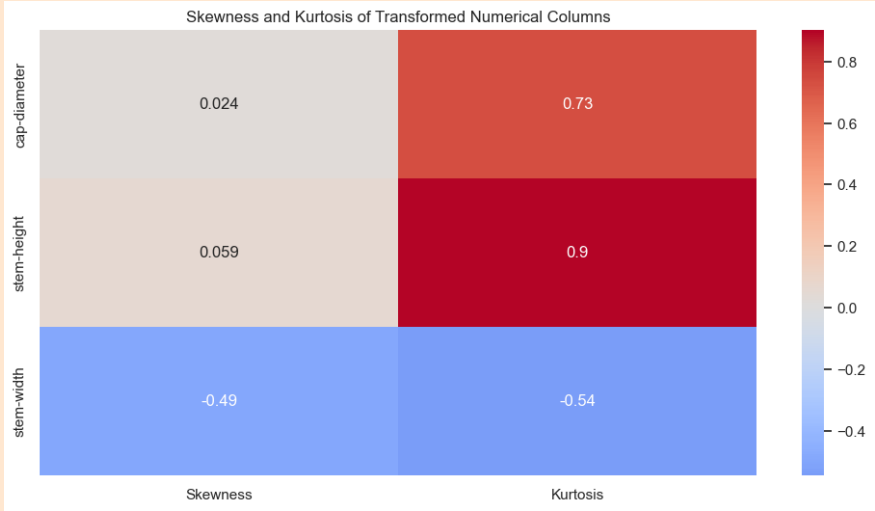
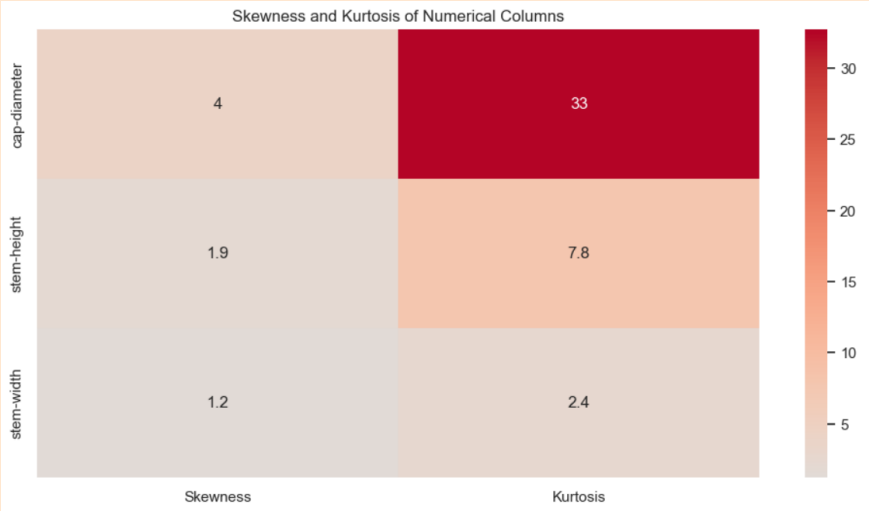


Cap-diameter의 경우 Box-Cox 변환 후 히스토그램이 더 대칭적인 형태로 바뀌고 Q-Q plot이 직선에 가까워진 것을 볼 수 있어 정규성이 크게 개선되었다고 할 수 있습니다. Stem-height, Stem-width도 로그변환 후 미세하게 개선이 되었음을 알 수 있습니다.

\* 로그 변환 : 데이터의 스케일을 조정하고 치우친 분포를 대칭적으로 만드는데 효과적이며, 양의 값을 가진 오른쪽으로 치우친 데이터에 주로 사용됩니다.  
\* Box-Cox 변환 : 로그변환을 포함한 더 일반화된 변환 방법으로, 데이터에 맞는 최적의 변환 파라미터를 찾아 정규성을 향상시키며, 특히 복잡한 비대칭 분포를 다루는데 유용합니다.

# 연속형 변수 처리하기

왜도-첨도 분석, 히스토그램, QQ-plot을 통한 연속형 변수의 분포 특성 분석 및 정규성 검토 결과 정규성과 극단적인 값 존재



첨도(Kurtosis):

- 3에 가까우면 정규 분포와 유사한 뾰족함
- 3보다 크면 정규 분포보다 뾰족함 (heavy-tailed)
- 3보다 작으면 정규 분포보다 평평함 (light-tailed)

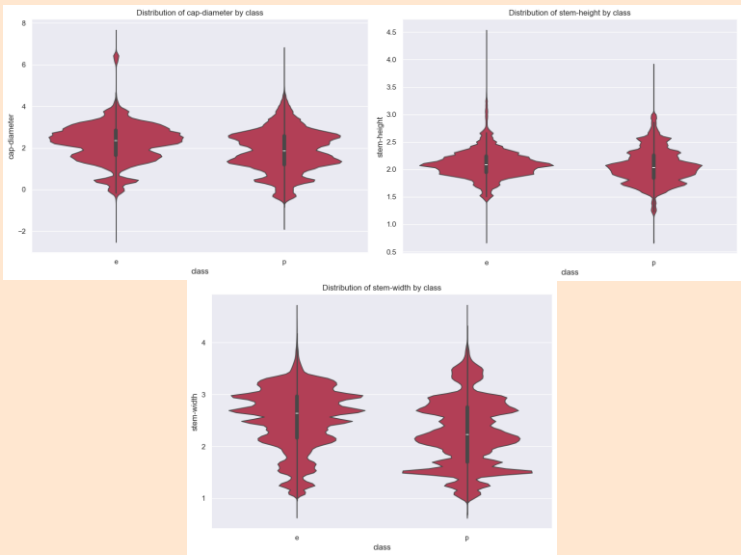
왜도(Skewness):

- |왜도| < 0.5: 거의 대칭
- 0.5 < |왜도| < 1: 중간 정도의 비대칭
- |왜도| > 1: 매우 비대칭

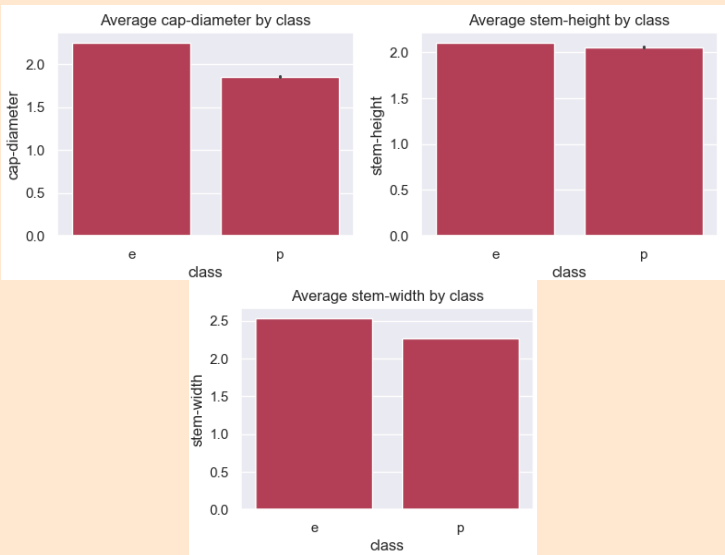


### 03. 데이터 전처리

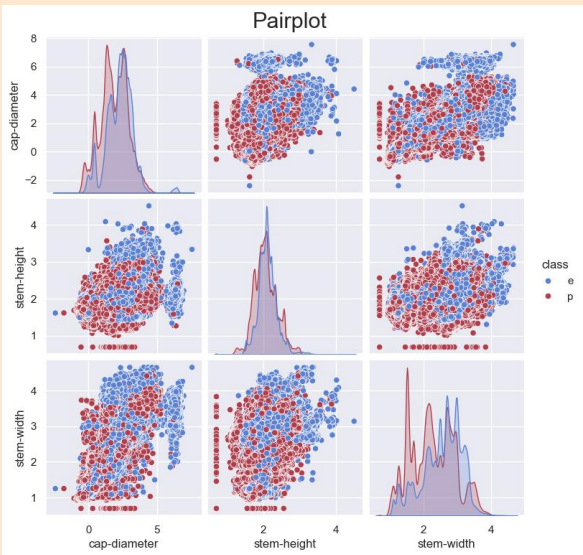
## 수치적 특성들 간의 상관관계 탐색



Stem-width에서 'e' 클래스가 'p'클래스보다 넓은 분포를 가지는 것이 뚜렷하게 보이긴 하지만 이러한 특성만으로는 두 클래스를 구분하기는 어려울 것으로 보입니다.



Cap-diameter, Stem-heigh에서는 클래스 간 평균 차이가 미미하지만, stem-width에서는 'e' 클래스와 평균이 'p' 클래스보다 높은 것을 볼 수 있습니다만, 분포의 복잡성이나 변동성은 크게 보이지 않습니다.



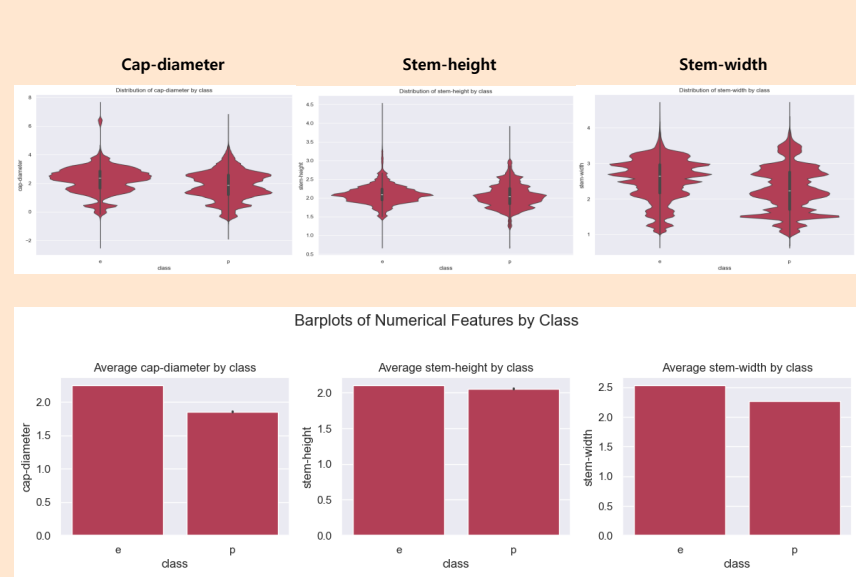
Stem-width와 다른 특성들 간의 관계에서 클래스 간 구분이 가장 뚜렷하게 나타나는 것을 확인 할 수 있습니다. 이는 Stem-width가 클래스 예측('e' 'p' 구분)에 중요한 역할(특성)을 할 수 있음을 시사합니다.

\* 로그 변환 : 데이터의 스케일을 조정하고 치우친 분포를 대칭적으로 만드는데 효과적이며, 양의 값을 가진 오른쪽으로 치우친 데이터에 주로 사용 됩니다.

\* Box-Cox 변환 : 로그변환을 포함한 더 일반화된 변환 방법으로, 데이터에 맞는 최적의 변환 파라미터를 찾아 정규성을 향상시키며, 특히 복잡한 비대칭 분포를 다루는데 유용합니다.

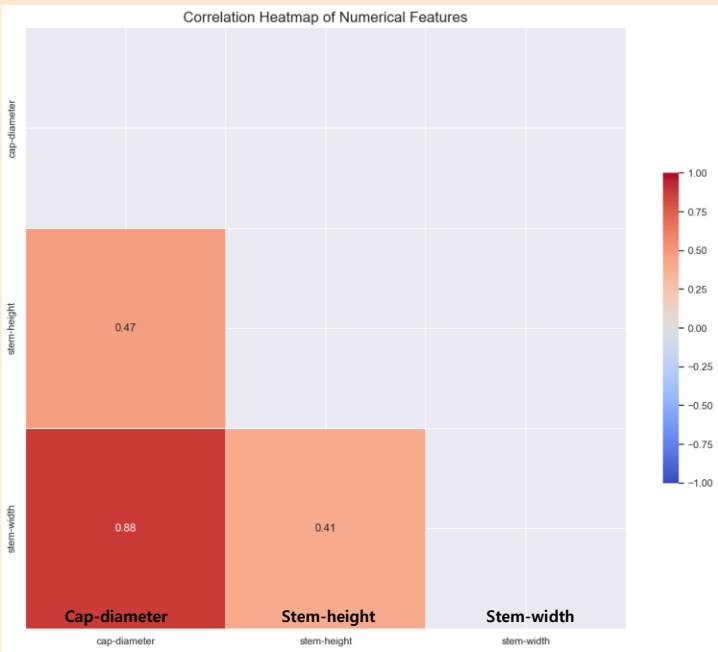
### 03. 데이터 전처리

## 수치적 특성들 간의 상관관계 탐색

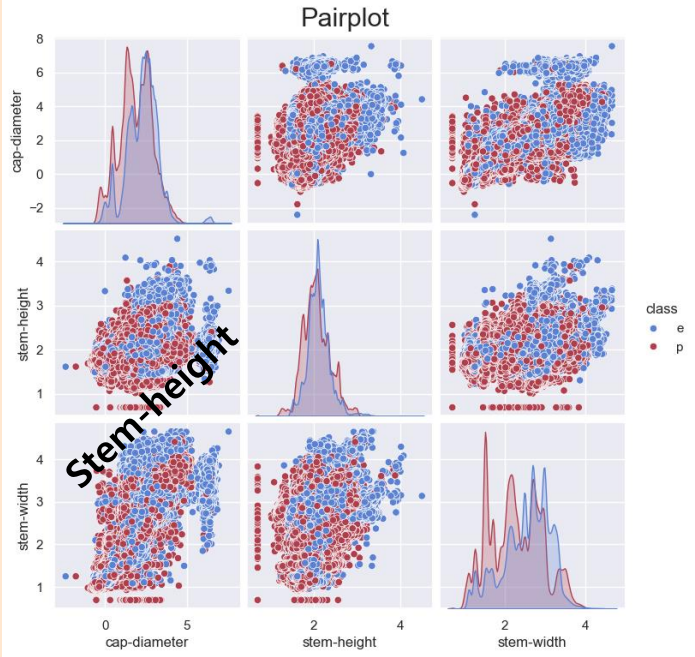


Violinplot : Stem-width에서 'e' 클래스가 'p'클래스보다 넓은 분포를 가지는 것이 뚜렷하게 보이긴 하지만 이러한 특성만으로는 두 클래스를 구분하기는 어려울 것으로 보입니다.

Barplot : Cap-diameter, Stem-height에서는 클래스 간 평균 차이가 미미하지만, stem-width에서는 'e' 클래스와 평균이 'p' 클래스보다 높은 것을 볼 수 있습니다만, 분포의 복잡성이나 변동성은 크게 보이지 않습니다.

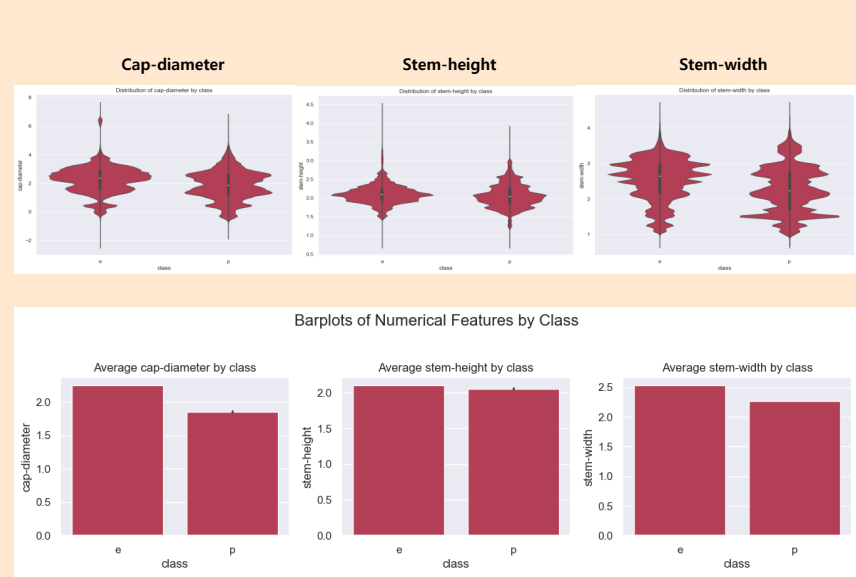


Correlation : 0.88의 높은 상관계수를 보이고 있어, 이 두 특성이 매우 강한 양의 상관관계를 가지고 있음을 알 수 있습니다. 이는 버섯의 갓 지름이 커질수록 줄기의 너비도 함께 증가하는 경향이 있다는 것을 의미합니다.



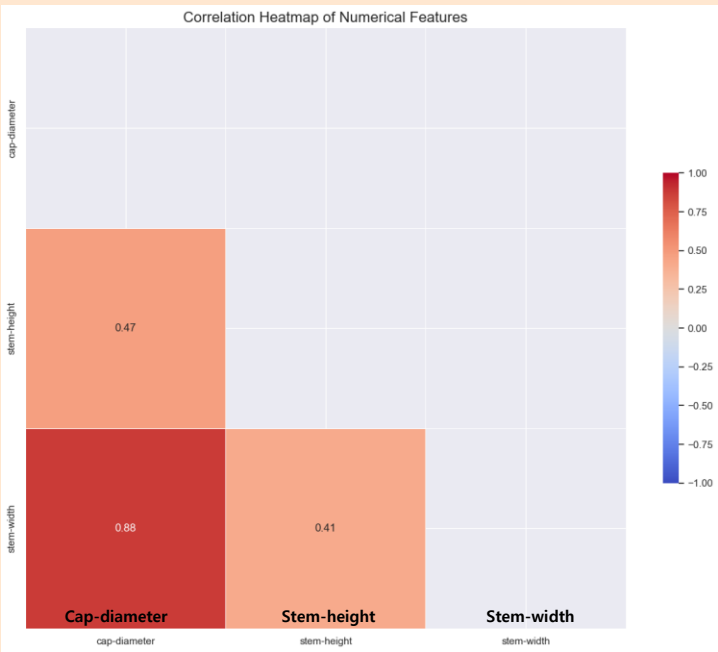
Pairplot : Stem-width와 다른 특성들 간의 관계에서 클래스 간 구분이 가장 뚜렷하게 나타나는 것을 확인 할 수 있습니다. 이는 Stem-width가 클래스 예측('e' 'p' 구분)에 중요한 역할 (특성)을 할 수 있음을 시사합니다.

\* 로그 변환 : 데이터의 스케일을 조정하고 치우친 분포를 대칭적으로 만드는데 효과적이며, 양의 값을 가진 오른쪽으로 치우친 데이터에 주로 사용 됩니다.  
\* Box-Cox 변환 : 로그변환을 포함한 더 일반화된 변환 방법으로, 데이터에 맞는 최적의 변환 파라미터를 찾아 정규성을 향상시키며, 특히 복잡한 비대칭 분포를 다루는데 유용합니다.

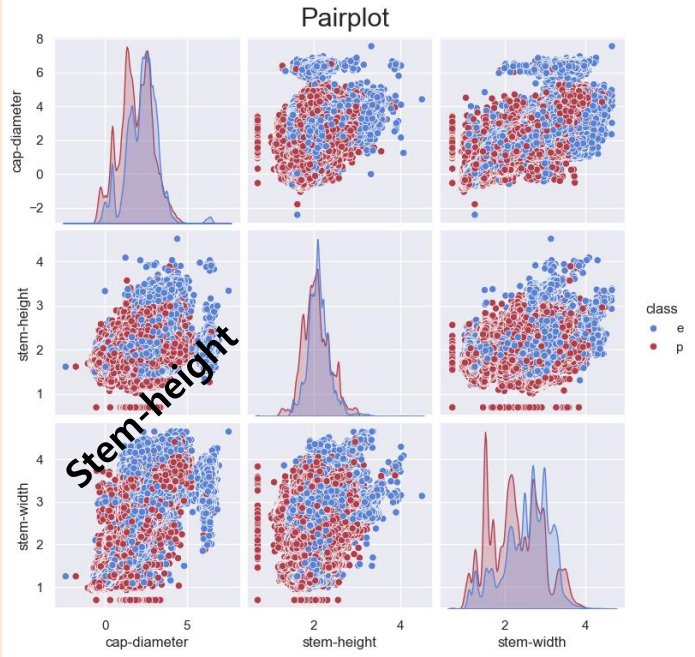


Violinplot : Stem-width에서 'e' 클래스가 'p'클래스보다 넓은 분포를 가지는 것이 뚜렷하게 보이긴 하지만 이러한 특성만으로는 두 클래스를 구분하기는 어려울 것으로 보입니다.

Barplot : Cap-diameter, Stem-heigh에서는 클래스 간 평균 차이가 미미하지만, stem-width에서는 'e' 클래스와 평균이 'p' 클래스보다 높은 것을 볼 수 있습니다만, 분포의 복잡성이나 변동성은 크게 보이지 않습니다.



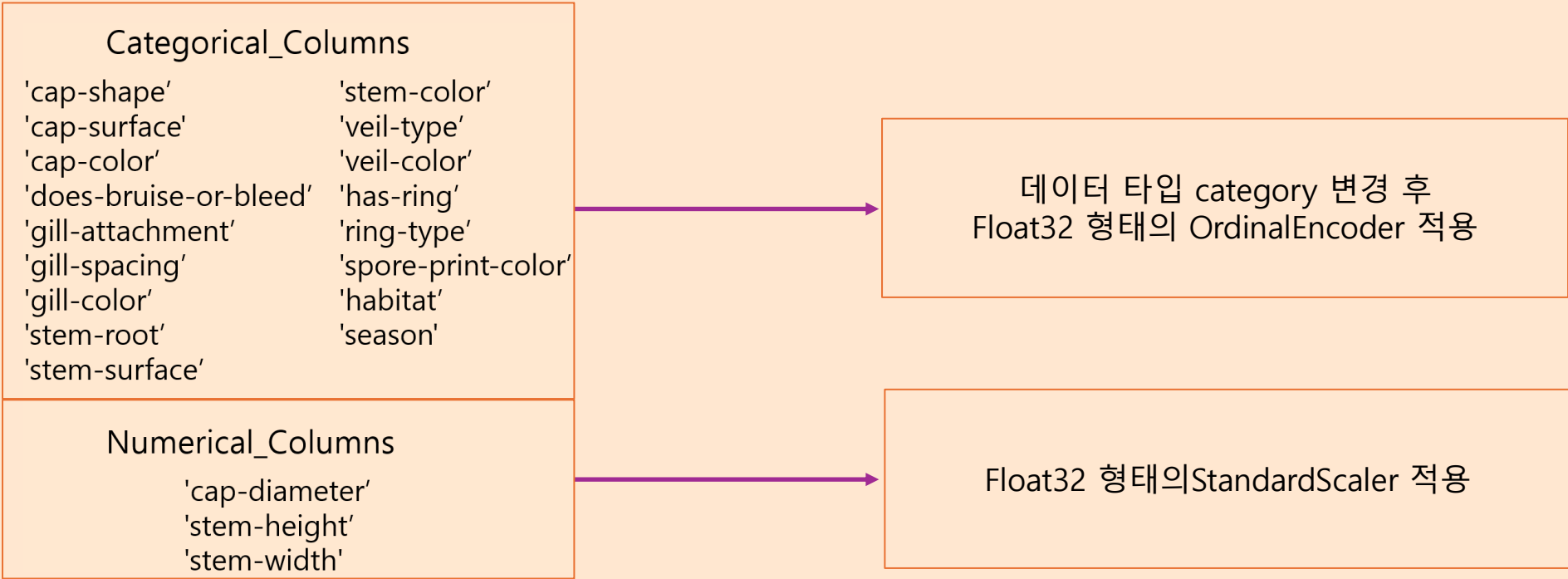
Correlation : 0.88의 높은 상관계수를 보이고 있어, 이 두 특성이 매우 강한 양의 상관관계를 가지고 있음을 알 수 있습니다. 이는 버섯의 갓 지름이 커질수록 줄기의 너비도 함께 증가하는 경향이 있다는 것을 의미합니다.



Pairplot : Stem-width와 다른 특성들 간의 관계에서 클래스 간 구분이 가장 뚜렷하게 나타나는 것을 확인 할 수 있습니다. 이는 Stem-width가 클래스 예측('e' 'p' 구분)에 중요한 역할 (특성)을 할 수 있음을 시사합니다.

\* 로그 변환 : 데이터의 스케일을 조정하고 치우친 분포를 대칭적으로 만드는데 효과적이며, 양의 값을 가진 오른쪽으로 치우친 데이터에 주로 사용 됩니다.  
\* Box-Cox 변환 : 로그변환을 포함한 더 일반화된 변환 방법으로, 데이터에 맞는 최적의 변환 파라미터를 찾아 정규성을 향상시키며, 특히 복잡한 비대칭 분포를 다루는데 유용합니다.

변수 인코딩, 스케일링 및 모델 메모리 최적화



\* OrdinalEncoder : 범주형 데이터를 수치형으로 변환. 범주 간 관계를 수치로 표현하며, NaN값과 같이 알 수 없는 범주의 처리가 가능  
\* StandardScaler : 수치형 특성을 정규화함. 특성 간 스케일 제거, 모특성 간 가중치 균형 유지, 이상치 영향 감소의 기능

**MODELING**

**OPTIMIZATION TUNING**

**모델링 최적화 튜닝**



XGBoost와 CatBoost 모델의 하이퍼 파라미터 최적화 :  
Grid SearchCV, Randomized SearchCV, Optuna

XGBClassifier  
+  
GridSearchCV

XGBClassifier  
+  
RandomizedSearchCV

CatBoost  
+  
Optuna

### XGBoost, Extreme Gradient Boost

- 기존 Gradient Tree Boosting 알고리즘을 개선한 고성능 지도학습 프레임워크입니다. 트리 기반 앙상블 모델로, 약한 학습기를 순차적으로 결합하여 강력한 예측 모델을 구축합니다.
- 주요 특징으로는 높은 예측 정확도, 과적합 방지 기능, 빠른 실행 속도가 있으며, 특징이며 Binary Prediction of Poisonous Mushrooms과 같은 대규모 데이터셋 처리에 효과적입니다. 효율적인 누락 값 처리, 병렬 처리 지원, 정규화 기법 등을 통해 성능을 최적화합니다.
- 단점으로는 계산 복잡도가 높으며, 하이퍼파라미터에 전문성이 요구되고, 대규모 데이터셋 처리시 높은 메모리 요구사항이 필요합니다.

\* OrdinalEncoder : 범주형 데이터를 수치형으로 변환. 범주 간 관계를 수치로 표현하며, NaN값과 같이 알 수 없는 범주의 처리가 가능

\* StandardScaler : 수치형 특성을 정규화함. 특성 간 스케일 제거, 모특성 간 가중치 균형 유지, 이상치 영향 감소의 기능

### XGBoost에 GridSearchCV, RandomizedSearchCV 적용

XGBClassifier  
+  
GridSearchCV

XGBClassifier  
+  
RandomizedSearchCV

#### XGBClassifier + GridSearchCV

- 모든 가능한 하이퍼파라미터 조합을 철저히 탐색하여 주어진 파라미터 공간 내에서 최적의 조합을 보장함
- 계산 비용이 높고 시간이 많이 소요되다 보니 파라미터 공간이 넓은 경우 비효율적임

#### XGBClassifier + RRandomizedSearchCV

- GridSearchCV 보다 적은 계산 리소스로 더 넓은 파라미터 공간을 빠르게 탐색이 가능함
- 무작위 특성으로 인해 최적의 조합을 놓칠 가능성과 적절한 반복 횟수의 설정이 필요함



“Catboost is a high-performance open source library for gradient boosting on decision trees”

### CatBoost: Ordered Boosting 기반의 고성능 그래디언트 부스팅 라이브러리

#### 기존의 Boosting 모델

기존 Boosting 모델들은 모든 훈련 데이터를 대상으로 잔차계산함

#### Catboost

Catboost는 Ordered Boosting 방식으로 일부 학습데이터를 사용해 잔차계산의 결과로 모델을 생성하고 다시 일부 학습 데이터로 잔차계산을 점진적으로 반복하여 모델을 꾸준히 개선해 나감

Ordered Boosting 방식 :

1. 학습데이터 무작위 정렬
2. 소수의 데이터로 초기 모델 학습
3. 모델 예측, 잔차 계산, 모델 개선
4. 데이터 포인트 점진적 확대
5. 전체 데이터셋을 사용할 때까지 3~4회 반복

Ordered Boosting 방식의 이점

점진적 학습으로 과적합을 감소

데이터의 순서를 무작위로 섞어 특정 순서에 의한 편향 감소  
계산 코스트 감소

“Optuna is an open-source hyperparameter optimization framework to automate hyperparameter search”

다양한 파라미터를 지원하고, 베이지안 최적화를 활용한 자동화된 하이퍼파라미터 튜닝 프레임 워크

### CatBoost + Optuna의 최적화 프로세스

CatBoost  
+  
Optuna

#### CatBoost와 Optuna의 조합

- Optuna는 많은 하이퍼파라미터를 가진 복잡한 모델(CatBoost)의 최적화에 유용함
- 모델의 성능을 자동화된 최적화 루프를 통해 지속적인 개선 가능성 증가
- 초기 설정 및 구현에 노력이 필요하고 계산 코스트가 증가할 가능성이 있음

1. 초기 설정:  
초기 파라미터 탐색 범위 정의
2. 목적 함수 정의 (objective):
  - CatBoost 모델 생성 및 학습
  - Matthews Correlation Coefficient (MCC) 계산 및 반환(\*MCC는 이진 분류 모델의 성능을 평가하는 지표)
3. 파라미터 범위 업데이트 함수 (update\_param\_ranges):
  - 현재 최적 파라미터 주변으로 탐색 범위 조정
4. Optuna 생성:
  - TPESampler를 사용하여 최대화 방향으로 스터디 설정
5. 최적화 루프:
  - 설정된 횟수(n\_trials)만큼 반복
  - 각 시도마다 목적 함수 실행 및 결과 기록
  - 일정 횟수 동안 성능 향상이 없으면 파라미터 범위 업데이트해 효과적인 탐색 수행
6. 최종 결과 출력:
  - 최적의 파라미터 및 최고 MCC 점수 출력

OPTIMIZATION MODEL  
LEARNING VALIDATION

최적화 된 모델 학습, 검증

ing

## 최적화된 모델 학습 및 검증

Optuna로 최적화된 파라미터의 CatBoost, Kfold 학습, 검증제출 Submission 파일 생성


CatBoost와 Optuna의 조합의 최적의 파라미터

```
'iterations': 2185  
'learning_rate': 0.026383282299563434,  
'depth': 10,  
'l2_leaf_reg': 5.850828606362642,  
'bootstrap_type': 'MVS',
```

모델 학습 &amp; 검증

CatBoost(best parameter)  
Kfold=15

Submission 파일



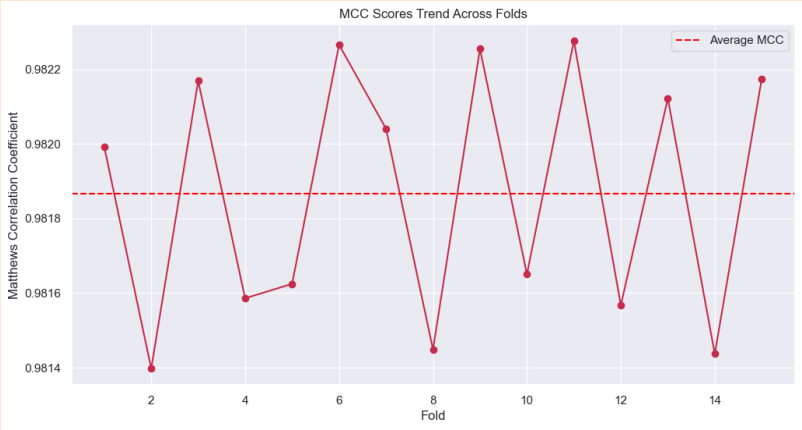
	id	class
0	3116945	e
1	3116946	p
2	3116947	e
3	3116948	p
4	3116949	e

# ANALYSIS AND INTERPRETATION OF RESULTS

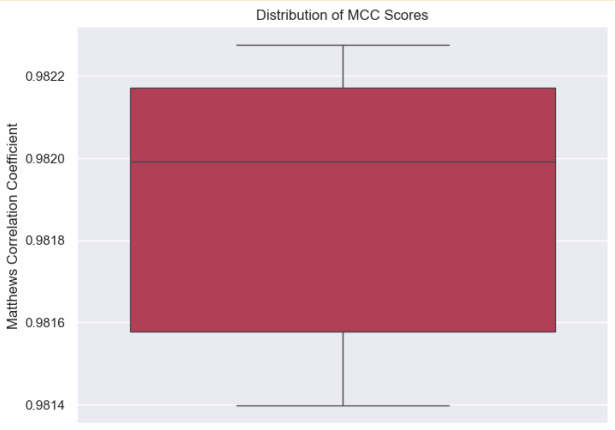
결과 분석 및 해석



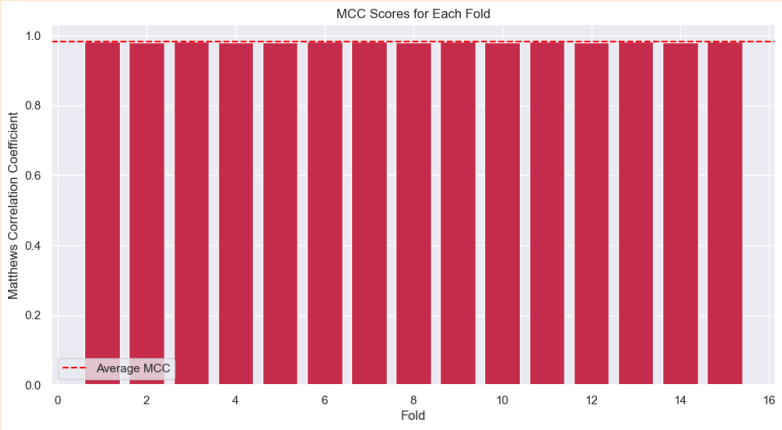
학습결과 MCC평가



MCC Scores Trend Across Folds:  
MCC 점수가 0.9814에서 0.9823 사이로 매우 안정적인 모습을 보여줌.  
최대 변동 폭이 약 0.0009로 변동성이 매우 낮음.  
모델이 다양한 데이터 분할에 대해 매우 안정적이고 높은 성능을 보이며,  
과적합의 징후가 없음을 알 수 있음



Distribution of MCC Scores:  
중앙값이 약 0.9820으로 높음.  
눈에 띄는 이상치는 없고 모델 성능이 일관적이고 안정적임



MCC Scores for Each Fold:  
모든 폴드에서 0.92 이상의 높은 일관성을 가진 MCC 점수를 보임  
최고 성능과 최저 성능의 폴드 간 차이가 작아 모든 폴드에서 일관되  
게 우수한 성능을 보이며 특정 데이터 분할에 대해서 과도하게 의존  
하지 않음을 알 수 있음

과적합 징후가 보이지 않으며 모델이 새로운 데이터에 대해서 일관된 성능을 보일 것으로 예상  
할 수 있으며, Optuna를 통해 최적화된 파라미터가 문제없이 효과적이었음을 시사함.

# Binary Prediction of Poisonous Mushrooms

Playground Series - Season 4, Episode 8



MCC 점수 통계(최소값:0.9814, 최대값:0.9823, 평균:0.9819, 표준편차:0.0003)

Fold 15 - Matthews Correlation Coefficient: 0.9821744047342607

Kaggle Score 651/2422