

Machine Learning in Python

사이킷런 소개

사이킷런 소개

- 파이썬 기반의 다른 머신러닝 패키지도 사이킷런 스타일의 API를 지향할 정도로 쉽고 가장 파이썬스러운 API를 제공
- 머신러닝을 위한 매우 다양한 알고리즘과 개발을 위한 편리한 프레임워크와 API를 제공
- 오랜기간 실전 환경에서 검증됐으며, 매우 많은 환경에서 사용되는 성숙한 라이브러리
- 주로 Numpy와 Scipy기반 위에서 구축된 라이브러리

사이킷런 소개

사이킷런 소개

Machine Learning in Python

The screenshot shows the official website for scikit-learn (<https://scikit-learn.org/stable/>). The top navigation bar includes links for 'Install', 'User Guide', 'API', 'Examples', and 'More'. The main content area features a large orange header with the text 'Simple and efficient tools for predictive data analysis' and a bulleted list of features:

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

The page is divided into three main sections:

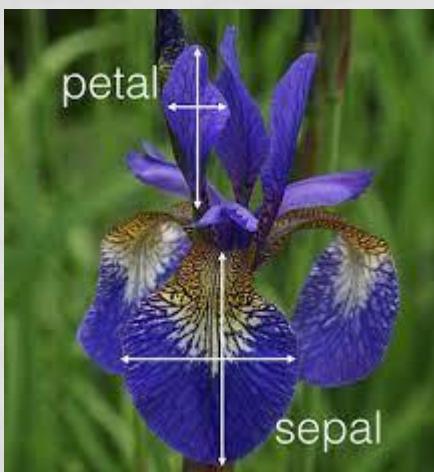
- Classification**: Describes identifying which category an object belongs to. It lists applications like spam detection and image recognition, and algorithms like SVM, nearest neighbors, random forest, and more. It includes a grid of 9x3 images showing classification results for various datasets.
- Regression**: Describes predicting a continuous-valued attribute associated with an object. It lists applications like drug response and stock prices, and algorithms like SVR, nearest neighbors, random forest, and more. It includes a line plot titled 'Boosted Decision Tree Regression' showing a fitted curve against training samples.
- Clustering**: Describes automatic grouping of similar objects into sets. It lists applications like customer segmentation and grouping experiment outcomes, and algorithms like k-Means, spectral clustering, mean-shift, and more. It includes a scatter plot titled 'K-means clustering on the digits dataset (PCA-reduced data)' showing data points grouped into five clusters with their respective centroids marked by 'x' symbols.

Below each main section are 'Examples' buttons, and at the bottom are additional sections: 'Dimensionality reduction', 'Model selection', and 'Preprocessing'.

사이킷런을 이용한 붓꽃 데이터 분류

사이킷런을 이용한 붓꽃 데이터 분류

- 사이킷런을 통해 첫 번째로 만들어볼 머신러닝 모델은 붓꽃 데이터 세트로 붓꽃의 품종을 분류(Classification)하는 것
- 붓꽃 데이터 세트는 꽃잎의 길이와 너비, 꽃받침의 길이와 너비 피처(Feature)을 기반으로 꽃의 품종을 예측하기 위한 것



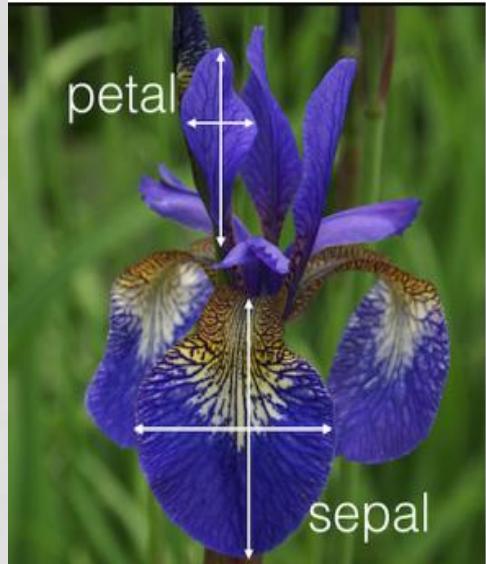
- Sepal length
- Sepal width
- Petal length
- Petal width

사이킷런을 이용한 붓꽃 데이터 분류

사이킷런을 이용한 붓꽃 데이터 분류

- 붓꽃 데이터 피처와 붓꽃 품종 레이블(Label)

피처(Feature)



붓꽃 품종 레이블(Label)

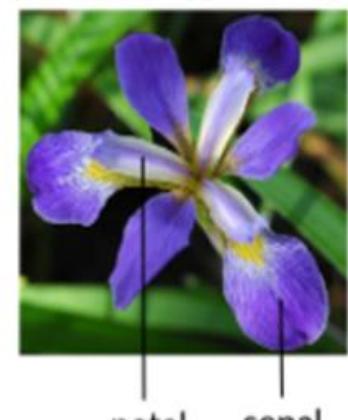
iris setosa



iris versicolor



iris virginica



머신러닝을 위한 용어 정리

머신러닝을 위한 용어 정리

- 피처(Feature)? 속성?

- 피처는 데이터 세트의 일반 속성임
- 머신러닝은 2차원 이상의 다차원 데이터에서도 많이 사용되므로 타겟값을 제외한 나머지 속성을 모두 피처로 지칭

- 레이블, 클래스, 타겟(값), 결정(값)

- 타겟값 또는 결정값은 지도 학습 시 데이터의 학습을 위해 주어지는 정답 데이터
- 지도 학습 중 분류의 경우에는 이 결정값을 레이블 또는 클래스로 지칭

지도학습 - 분류

분류(Classification)

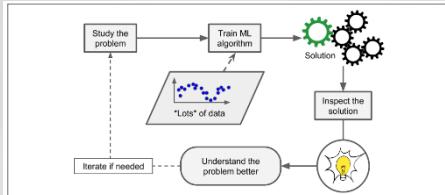
- 대표적인 지도학습(Supervised Learning) 방법의 하나
- 지도학습은 학습을 위한 다양한 피처와 분류 결정값인 레이블(Label) 데이터로 모델을 학습한 뒤, 별도의 테스트 세트에서 미지의 레이블을 예측
- 즉 지도학습은 명확한 정답이 주어진 데이터를 먼저 학습한 뒤 미지의 정답을 예측하는 방식
- 이 때 학습을 위해 주어진 데이터 세트를 학습데이터 세트, 머신러닝 모델의 예측 성능을 평가하기 위해 별도로 주어진 데이터 세트를 테스트 데이터 세트

지도학습 - 분류

분류(Classification)

붓꽃 데이터 분류

학습 데이터



꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	Iris 꽃 종류
5.3	3.5	1.4	0.2	Setora
4.9	3.0	1.4	0.2	Setora
6.4	3.5	4.5	1.2	Versicolor
.

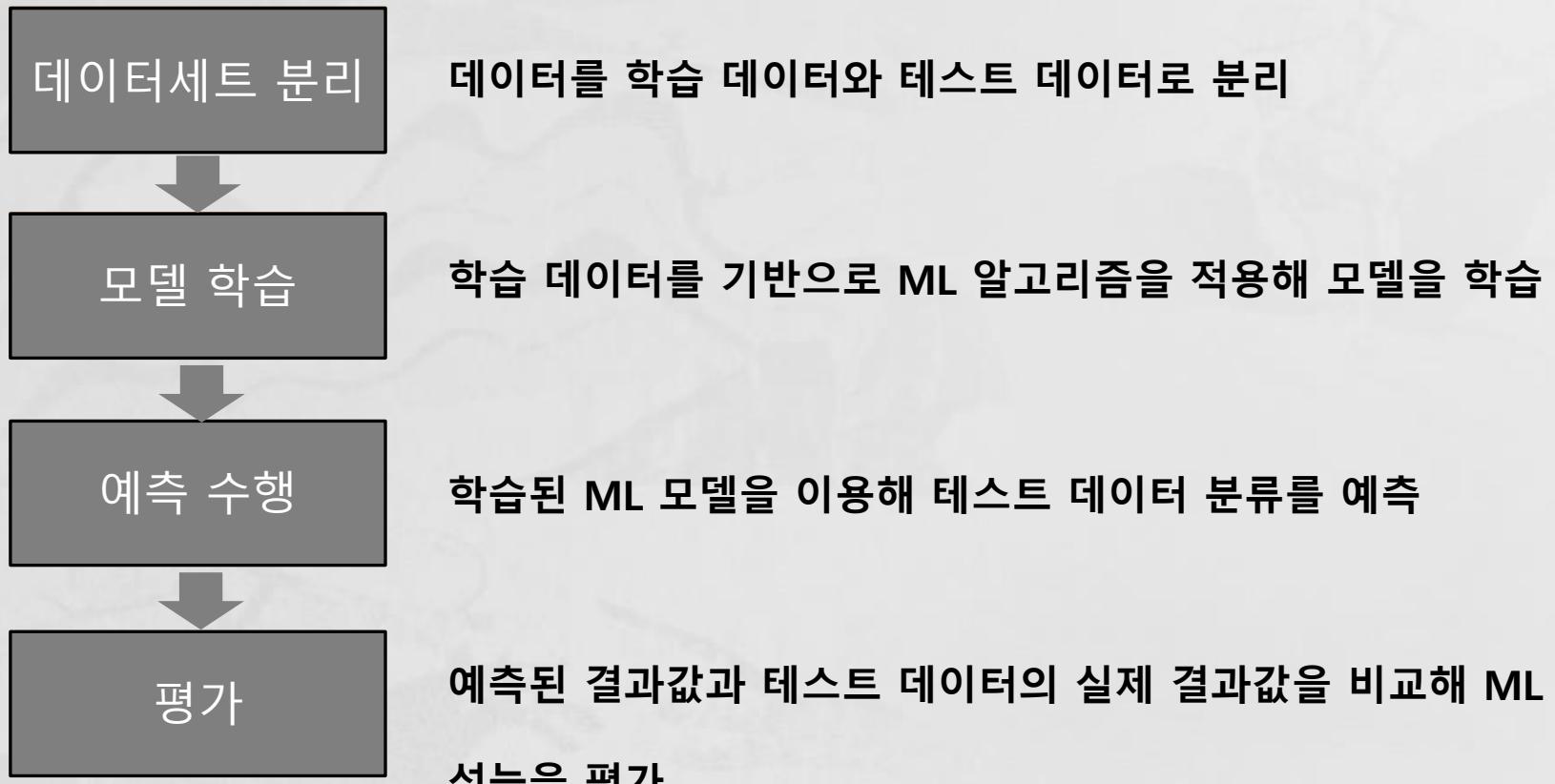
테스트 데이터

꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	Iris 꽃 종류
5.3	3.2	1.1	0.3	?
4.2	2.0	2.4	0.4	?
6.5	3.8	5.5	1.1	?
.

붓꽃 데이터 분류 예측 프로세스

붓꽃 데이터 분류 예측 프로세스

- 분류 예측 프로세스



붓꽃 데이터 분류 예측 프로세스

붓꽃 데이터 분류 예측 프로세스

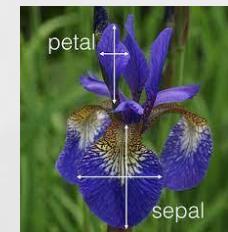
• 분류 예측 프로세스

학습
데이터

번호	꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	Iris 꽃 종류
1	5.3	3.5	1.4	0.2	Setora
2	4.9	3.0	1.4	0.2	Setora
....				
50	6.4	3.5	4.5	1.2	Versicolor
....				
150	5.9	3.0	5.0	1.8	Virginica

학습 데이터로
모델 학습

모델학습



테스트
데이터

번호	꽃받침 길이	꽃받침 너비	꽃잎 길이	꽃잎 너비	Iris 꽃 종류
1	5.3	3.2	1.1	0.3	?
....					?
50	6.5=4	3.5	4.5	1.2	?

학습 모델을 통해
테스트 데이터의
레이블 값 예측

학습된 모델로
테스트 데이터
예측

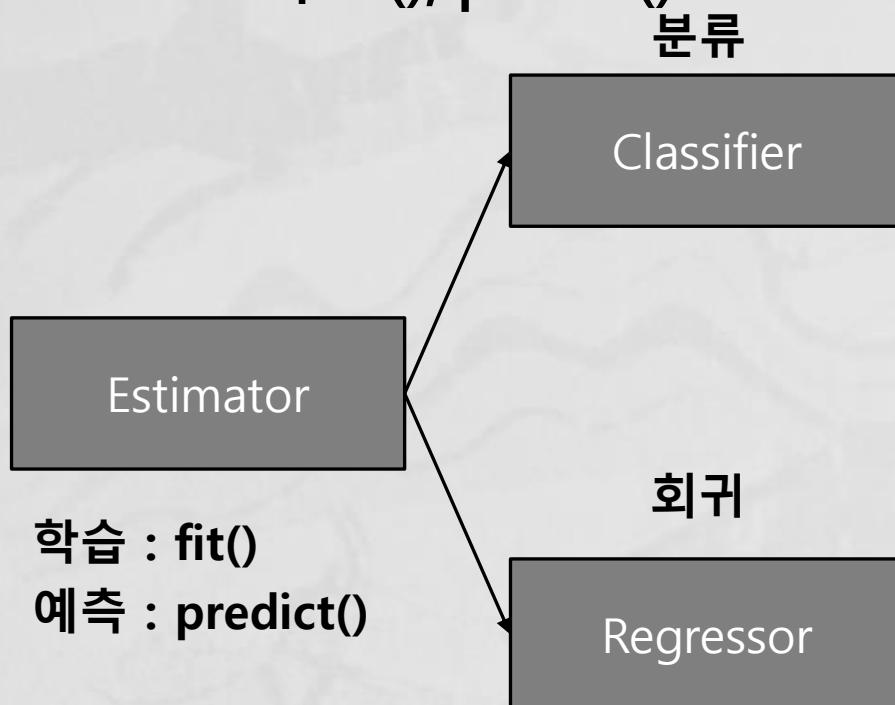
예측된 레이블 값
과 실제 레이블
예측 정확도 평가

평가

사이킷런 기반 프레임워크

Estimator와 fit(), predict()

- Estimator와 fit(), predict()



학습 : fit()

예측 : predict()

분류 구현 클래스

DecisionTreeClassifier

RandomForestClassifier

GradientBoostingClassifier

GaussianNB

SVC

회귀 구현 클래스

LinearRegression

Ridge

Lasso

RandomForestRegressor

GradientBoostingRegressor

사이킷런의 주요 모듈

사이킷런의 주요 모듈

분류	모듈명	설명
예제 데이터	sklearn.datasets	사이킷런에 내장되어 예제로 제공하는 데이터 세트
피처 처리	sklearn.preprocessing	데이터 전처리에 필요한 다양한 가공 기능 제공 (인코딩, 정규화, 스케일링 등)
	sklearn.feature_selection	알고리즘에 큰 영향을 미치는 피처를 우선순위대로 선택 작업을 수행하는 다양한 기능 제공
	sklearn.feature_extraction sklearn.feature_extraction.text (텍스트데이터) sklearn.feature_extraction.image (이미지 데이터)	텍스트 데이터나 이미지 데이터의 벡터화된 피처를 추출하는데 사용함. (예를 들어 텍스트 데이터에서 Count Vectorizer나 Tf-IDF Vectorizer 등을 생성하는 기능 제공.)
피처 처리 & 차원 축소	sklearn.decomposition	차원 축소와 관련한 알고리즘을 지원하는 모듈. (PCA, NMF, Truncated SVD)

사이킷런의 주요 모듈

사이킷런의 주요 모듈

분류	모듈명	설명
터 분리, 검증 & 파라미터 튜닝	sklearn.model_selection	교차 검증을 위한 train/test 데이터 분리, 그리드 서치(GridSearch)로 최적 파라미터 추출 등의 API 제공
평가	sklearn.metrics	분류, 회귀, 클러스터링, 페어와이즈(Pairwise)에 대한 다양한 성능 측정 방법 제공 (Accuracy, Precision, Recall, ROC-AUC, RMSE 등)

사이킷런의 주요 모듈

사이킷런의 주요 모듈

분류	모듈명	설명
ML 알고리즘	sklearn.ensemble	앙상블 알고리즘 제공. (랜덤포레스트, 에이다부스트, 그래디언트 부스팅 등)
	sklearn.linear_model	주로 선형회귀, 릿지(Ridge), 라쏘(Lasso) 및 로지스틱 회귀 등 회귀 관련 알고리즘을 지원. 또한 SGD(Stochastic Gradient Descent) 관련 알고리즘도 제공
	sklearn.naive_bayes	나이브 베이즈 알고리즘 제공. 가우시안 NB, 다항분포 NB 등.
	sklearn.neighbors	최근접 이웃 알고리즘 제공, K-NN 등
	sklearn.svm	서포트 벡터 머신 알고리즘 제공
	sklearn.tree	의사 결정 트리 알고리즘 제공
	sklearn.cluster	비지도 클러스터링 알고리즘 제공 (K-평균, 계층형,DBSCAN 등)
유틸리티	sklearn.pipeline	피처 처리 등의 변환과 ML알고리즘 학습, 예측 등을 함께 묶어서 실행할 수 있는 유틸리티 제공

사이킷런 내장 예제 데이터 셋

분류 및 회귀용

API 명	설명
datasets.load_boston()	회귀 용도, 미국 보스턴의 집 피처들과 가격에 대한 데이터 세트
datasets.load_breast_cancer()	분류 용도. 위스콘신 유방암 피처들과 악성/음성 레이블 데이터 세트
datasets.load_diabetes()	회귀 용도. 당뇨 데이터 세트
datasets.load_digits()	분류 용도. 0~9까지 숫자의 이미지 픽셀 데이터 세트
datasets.load_iris()	분류 용도. 붓꽃 데이터 피처를 가진 데이터 세트

내장 예제 데이터 셋 구성

내장 예제 데이터 셋 구성

feature_names	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target_names
	5.1	3.5	1.4	0.2	setosa, versicolor, virginica
	4.9	3.0	1.4	0.2	(0 , 1 , 2)
data	
	4.6	3.1	1.5	0.2	0
	5.0	3.6	1.4	0.2	1
				
					2
					0

붓꽃 데이터 세트

붓꽃 데이터 세트

- 꽃잎의 길이와 너비, 꽃받침의 길이와 너비 feature을 기반으로 품종을 예측하기 위한 것
- 지도학습 중 분류
- 독립변수(x): 꽃잎의 길이, 너비, 꽃받침의 길이, 너비
- 종속변수(y): 품종(Setosa, Vesicolor, Virginica)

붓꽃 데이터 분류 예측 프로세스

붓꽃 데이터 분류 예측 프로세스

1. **데이터 세트 분리**: 데이터를 training data와 test data로 분리
2. **모델 학습**: training data를 기반으로 ML 알고리즘을 적용하여 모델을 학습
3. **예측 수행**: 학습된 ML 모델을 이용해 test data의 분류를 예측
4. **평가**: 이렇게 예측된 결과값과 test data의 실제 결괏값을 비교해 ML 모델 성능을 평가

Model Selection 소개

학습 데이터와 테스트 데이터

- 학습 데이터 세트
- 머신러닝 알고리즘의 학습을 위해 사용
- 데이터의 속성들과 결정값(레이블) 값 모두를 가지고 있음
- 학습 데이터를 기반으로 머신러닝 알고리즘이 데이터 속성과 결정값의 패턴을 인지하고 학습

Model Selection 소개

학습 데이터와 테스트 데이터

- 테스트 데이터 세트
- 테스트 데이터 세트에서 학습된 머신러닝 알고리즘을 테스트
- 테스트 데이터는 속성 데이터만 머신러닝 알고리즘에 제공하며, 머신러닝 알고리즘은 제공된 데이터를 기반으로 결정값을 예측
- 테스트 데이터는 학습 데이터와 별도의 데이터 세트로 제공되어야 함

학습 데이터와 테스트 데이터 분리

train_test_split()

- sklearn.model_selection의 train_test_split() 함수

```
X_train, X_test, y_train, y_test = train_test_split(iris_data.data,  
                                                iris_data.target, test_size=0.3, random_state=121)
```

- test_size : 전체 데이터에서 테스트 데이터 세트 크기를 얼마로 샘플링 할 것인가를 결정. 디폴트는 0.25, 즉 25%
- train_size : 전체 데이터에서 학습용 데이터 세트 크기를 얼마로 샘플링 할 것인가를 결정
- shuffle : 데이터를 분리하기 전에 데이터를 미리 섞을지를 결정. 디폴트는 True. 데이터를 분산시켜서 더 효율적인 학습 및 테스트 데이터 세트를 만드는데 사용
- random_state : 호출할 때마다 동일한 학습/테스트용 데이터 세트를 생성하기 위해 주어지는 난수 값

교차 검증(Cross Validation)

교차 검증(Cross Validation)

학습 데이터를 다시 분할하여 학습 데이터와 학습된 모델의 성능을 일차 평가하는 검증 데이터로 나눔

모든 학습/검증 과정이 완료된 후 최종적으로 성능을 평가하기 위한 데이터 세트

학습 데이터 세트

테스트 데이터 세트

분할



학습 데이터 세트

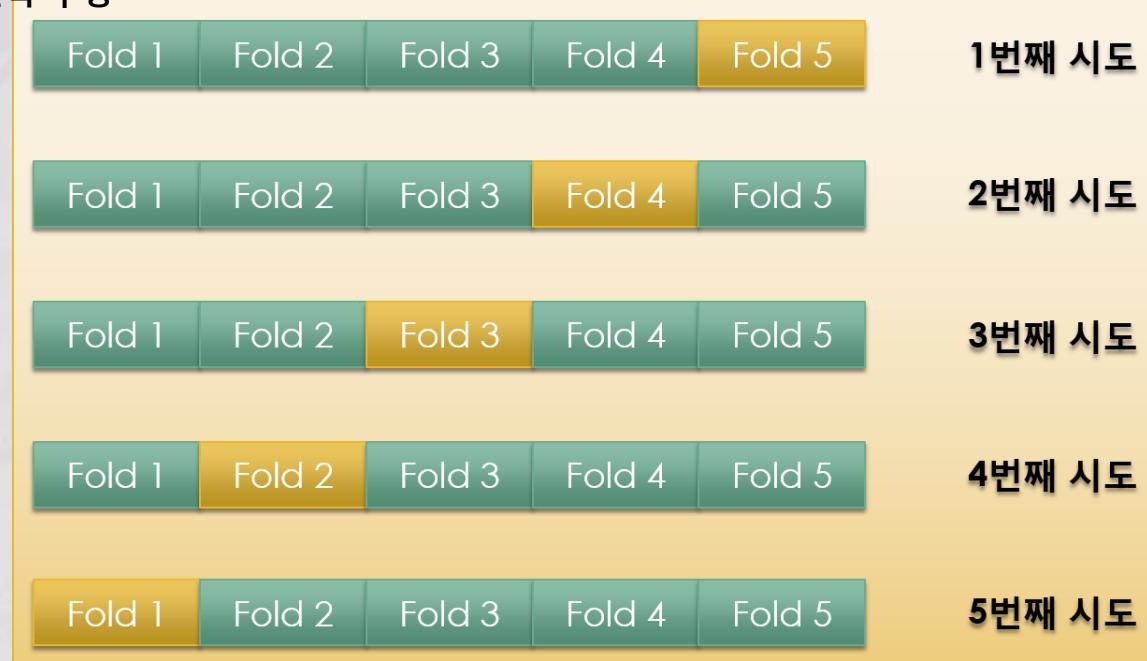
검증 데이터
세트

K 폴드(K-Fold) 교차 검증

K 폴드(K-Fold) 교차 검증

K=5일 경우

총 5개의 폴드 세트에 5번의 학습과 검증
평가 반복 수행



Test Data를 활용한 검증

K 폴드(k-fold) 교차 검증

K 폴드 교차 검증

K폴드 교차 검증

- 일반 K폴드
- Stratified K 폴드

- 불균형한(imbalanced) 분포도를 가진 레이블(결정 클래스) 데이터 집합을 위한 K 폴드 방식
- 학습데이터와 검증 데이터 세트가 가지는 레이블 분포도가 유사하도록 검증 데이터 추출

교차 검증을 보다 간편하게

Cross_val_score()

- Kfold 클래스를 이용한 교차 검증 방법

- 1. 폴드 세트 설정
- 2. For 루프에서 반복적으로 학습/검증 데이터 추출 및 학습과 예측 수행
- 3. 폴드 세트별로 예측 성능을 평균하여 최종성능 평가



cross_val_score()
함수로 폴드 세트
추출, 학습/예측, 평
가를 한번에 수행

```
cross_val_score(estimator, X, y=None, scoring=None,  
cv=None, n_jobs=1, verbose=0, fit_params=None,  
pre_dispatch='2*n_jobs')
```

교차 검증과 최적 하이퍼 파라미터 튜닝

GridSearchCV

- 사이킷런은 GridSearchCV를 이용해 Classifier나 Regressor와 같은 알고리즘에 사용되는 하이퍼 파라미터를 순차적으로 입력하면서 편리하게 최적의 파라미터를 도출할 수 있는 방안을 제공

```
Grid_parameters = {'max_depth':[1,2,3],  
'min_samples_split':[2,3]}  
}
```

CV 세트가 3이라면

파라미터 순차 적용 횟수	CV 세트 수	학습/검증 총 수행 횟수
6	X	18

순번	Max_depth	Min_samples_split
1	1	2
2	1	3
3	2	2
4	2	3
5	3	2
6	3	3

데이터 전처리(Preprocessing)

데이터 전처리

- 데이터 클린징
- 결손값 처리(Null/NaN 처리)
- 데이터 인코딩(레이블, 원-핫 인코딩)
- 데이터 스케일링
- 이상치 제거
- Feature 선택, 추출 및 가공

데이터 인코딩

데이터 인코딩

- 머신러닝 알고리즘은 문자열 데이터 속성을 입력 받지 않으며 모든 데이터는 숫자형으로 표현되어야 함
- 문자형 카테고리형 속성은 모두 숫자값으로 변환/인코딩되어야 함

레이블(Label) 인코딩

원 – 핫(One-Hot) 인코딩

레이블(Label) 인코딩

레이블 인코딩

원본 데이터

상품분류	가격
TV	1,000,000
냉장고	1,500,000
전자레인지	200,000
컴퓨터	800,000
선풍기	100,000
선풍기	100,000
믹서	50,000
믹서	50,000

상품 분류를 레이블 인코딩 한 데이터

상품분류	가격
0	1,000,000
1	1,500,000
4	200,000
5	800,000
3	100,000
3	100,000
2	50,000
2	50,000

[TV, 냉장고, 전자레인지, 컴퓨터, 선풍기, 믹서]

[0,1,4,5,3,2]

원-핫(One-Hot) 인코딩

원-핫(One-Hot) 인코딩

- 원-핫 인코딩은 피처 값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 컬럼에만 1을 표시하고 나머지 컬럼에는 0을 표시하는 방식

TV:0, 냉장고:1, 전자레인지:4, 컴퓨터:5, 선풍기:3, 믹서:2를 원-핫 인코딩

The diagram illustrates the process of one-hot encoding. On the left, a vertical table shows categories: TV, 냉장고 (refrigerator), 전자레인지 (microwave), 컴퓨터 (computer), 선풍기 (fan), 선풍기 (fan), 믹서 (mixer), and 믹서 (mixer). An arrow points from this table to a larger matrix on the right. The matrix has columns labeled '상품분류_TV', '상품분류_냉장고', '상품분류_믹서', '상품분류_선풍기', '상품분류_전자레인지', and '상품분류_컴퓨터'. The rows correspond to the categories in the table. The '냉장고' row is highlighted with a black box around the '냉장고' column, which contains a value of 1, indicating it is the selected category for that row.

상품분류	상품분류_TV	상품분류_냉장고	상품분류_믹서	상품분류_선풍기	상품분류_전자레인지	상품분류_컴퓨터
TV	1	0	0	0	0	0
냉장고	0	1	0	0	0	0
전자레인지	0	0	0	0	1	0
컴퓨터	0	0	0	0	0	1
선풍기	0	0	0	1	0	0
선풍기	0	0	0	1	0	0
믹서	0	0	1	0	0	0
믹서	0	0	1	0	0	0

원-핫(One-Hot) 인코딩

사이킷런 원-핫(One-Hot) 인코딩

원본 데이터

상품분류	가격
TV	1,000,000
냉장고	1,500,000
전자레인지	200,000
컴퓨터	800,000
선풍기	100,000
선풍기	100,000
믹서	50,000
믹서	50,000

숫자로 인코딩

상품분류	가격
0	1,000,000
1	1,500,000
4	200,000
5	800,000
3	100,000
3	100,000
2	50,000
2	50,000

TV	냉장고	믹서	선풍기	전자레인지	컴퓨터	가격
1	0	0	0	0	0	1,000
0	1	0	0	0	0	1,500
0	0	0	0	1	0	200
0	0	0	0	0	1	800
0	0	0	1	0	0	100
0	0	0	1	0	0	100
0	0	1	0	0	0	50
0	0	1	0	0	0	50

피처 스케일링

피처 스케일링

- 데이터 전처리(Data Preprocessing) - Scaling
 - 범위(Scale)가 다른 변수들의 범위(Scale)를 비슷하게 맞추기 위한 목적
 - 연속형 변수가 다양한 범위(Scale)로 존재할 때 제곱 오차 계산 시 왜곡 발생
Ex) x_1 은 1에서 10 사이 스케일, x_2 는 1000에서 100만 사이 스케일
 - 스케일이 더 큰 변수에 맞추어서 가중치를 최적화하는 문제 발생
- Scaling in Python
 - `from sklearn.preprocessing import MinMaxScaler`
 - `from sklearn.preprocessing import StandardScaler`

피처 스케일링

피처 스케일링

- 표준화는 데이터 피처 각각의 평균이 0이고 분산이 1인 가우시안 정규 분포를 가진 값으로 변환하는 것을 의미
- 정규화는 서로 다른 피처의 크기를 통일하기 위해 크기를 변환해주는 개념

피처 스케일링

피처 스케일링

- Normalization(정규화)

- 변수의 스케일을 0~1 사이 범위로 맞추는 것(min-max scaling)
- 정규화는 변수의 범위가 정해진 값이 필요할 때 유용하게 사용

$$X_{Normalization} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

```
[ ] from sklearn.preprocessing import MinMaxScaler
```

```
scaler1 = MinMaxScaler()  
X_Norm = scaler1.fit_transform(X)
```

피처 스케일링

피처 스케일링

- Standardization(표준화)

- 변수의 평균을 0, 표준편차를 1로 만들어 **표준정규분포의 특징**을 갖도록 함
- 표준화는 가중치(weight) 학습을 더 쉽게 할 수 있도록 함

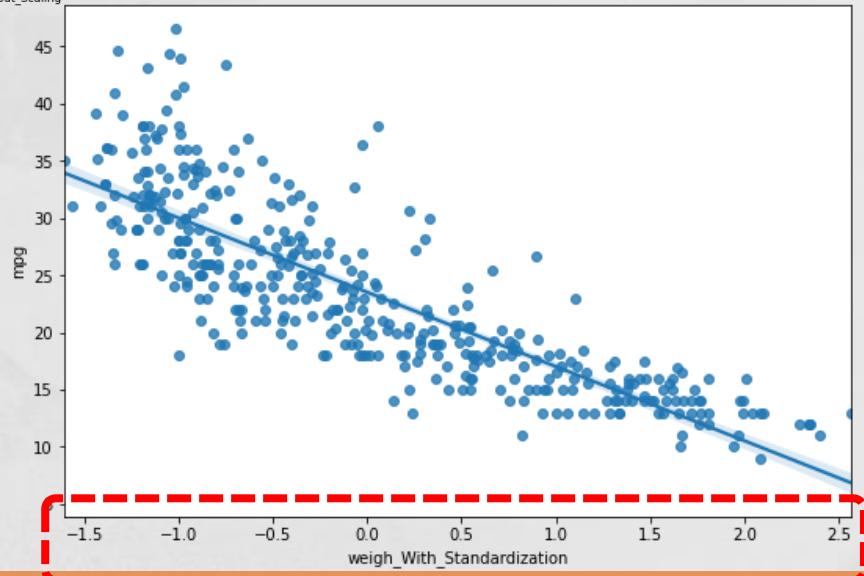
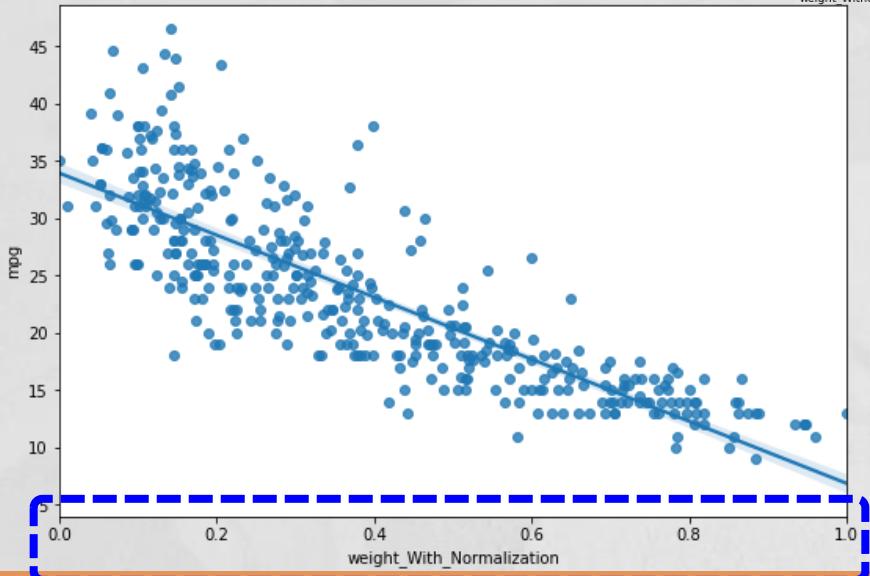
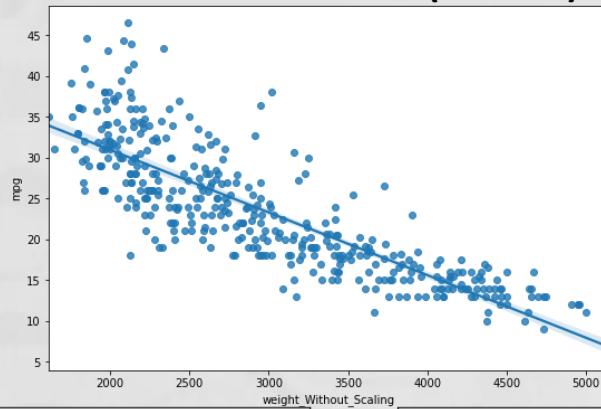
$$X_{Standardization} = \frac{X - mean(X)}{std (X)}$$

```
[ ] from sklearn.preprocessing import StandardScaler  
  
scaler2 = StandardScaler()  
X_Stan = scaler2.fit_transform(X)
```

피처 스케일링

피처 스케일링

- Normalization(정규화) vs. Standardization(표준화) 2/2



타이타닉 생존자 ML 예측 구현

타이타닉 생존자 ML 예측 구현

데이터 전처리

모델 학습 및
검증/예측/평가

- Null 처리
- 불필요한 속성 제거
- 인코딩 수행
- 결정 트리, 랜덤포레스트, 로지스틱 회귀 학습 비교
- K 폴더 교차 검증
- `cross_val_score()`와 `GridSearchCV()` 수행



머신 러닝 지도 학습 프로세스

머신러닝 지도 학습 프로세스

데이터 전처리

데이터 세트 분리

모델 학습 및
검증 평가

예측 수행

평가

- 데이터 클린징
- 결손값 처리
(Null/Nan 처리)
- 데이터 인코딩
(레이블, 원-한인
코딩)
- 데이터 스케일링
- 이상치 제거
- Feature 선택, 추
출 및 가공

- 학습데이터/
테스트
- 데이터 분리

- 알고리즘
학습

- 테스트 데이
터로 예측
수행

- 예측 평가

교차 검증
`cross_val_score()`
`GridSearchCV`

정리

정리

- 사이킷런을 이용한 붓꽃 데이터 분류
- Model Selection 소개
- 교차 검증
- 데이터 전처리

[부록] Google Colab

Google Colab

Google Colaboratory 1/5

- 설치가 필요 없는 클라우드 기반의 무료 Jupyter Notebook 환경
- 크롬브라우저 환경을 통하여 코드를 작성, 실행하여 분석 수행
- Colab에서 작성된 스크립트는 Google Drive에 자동 저장
- 구글 계정으로 Colab에 로그인하여 사용
- GPU 및 TPU 컴퓨팅 환경 지원



Google Colab

Google Colaboratory 2/5

- <https://colab.research.google.com>

The screenshot shows the Google Colaboratory interface. At the top, there's a navigation bar with the 'Colaboratory' logo, file menu items (파일, 수정, 보기, 삽입, 런타임, 도구, 도움말), and a '로그인' button (circled in red). Below the navigation bar is a toolbar with icons for code (+ 코드), text (+ 텍스트), and clipboard (드라이브로 복사). On the left, a sidebar contains links like 'Colaboratory 소개', '시작하기', '추가 리소스', '머신러닝 예제: Seedbank', and '섹션'. The main content area has a title '시작하기' and a sub-section about Jupyter notebooks. It shows a code cell with the Python code `seconds_in_a_day = 24 * 60 * 60` and its output `86400`. Below this, another code cell shows `seconds_in_a_week = 7 * seconds_in_a_day` and its output `604800`. A note at the bottom says to refer to the 'Colaboratory 가요' for more information. A '추가 리소스' section is also present.

Colaboratory에 오신 것을 환영합니다

파일 수정 보기 삽입 런타임 도구 도움말

+ 코드 + 텍스트 드라이브로 복사

로그인

공유

연결 수정 가능

목차 코드 스니펫 파일 X

시작하기

Colaboratory 소개

머신러닝 예제: Seedbank

추가 리소스

섹션

시작하기

현재 문서는 Colaboratory에 호스팅된 [Jupyter 노트](#)입니다. 정적인 페이지가 아닌, Python 등의 언어로 코드를 작성하고 실행할 수 있는 대화형 환경입니다.

예를 들어 다음은 값을 계산하여 변수로 저장하고 결과를 출력하는 간단한 Python 스크립트가 포함된 **코드 셀**입니다.

```
[ ] seconds_in_a_day = 24 * 60 * 60  
seconds_in_a_day
```

86400

위 셀의 코드를 실행하려면 셀을 클릭하여 선택한 후, 코드 왼쪽의 실행 버튼을 누르거나 단축키 'Command/Ctrl+Enter'를 사용합니다.

모든 셀은 동일한 전역 상태를 수정하므로 셀을 실행하여 정의되는 변수는 다른 셀에서도 사용할 수 있습니다.

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day  
seconds_in_a_week
```

604800

Colaboratory 노트 사용 방법을 자세히 알아보려면 [Colaboratory 가요](#)를 참조하세요.

추가 리소스

다음 리소스를 통해 Python, Jupyter, Colaboratory 및 관련 도구를 최대한 활용하는 방법을 알아보세요.

Colaboratory에서 노트로 작업하기

Google Colab

Google Colaboratory 3/5

- <https://colab.research.google.com>

The screenshot shows the Google Colaboratory interface. At the top, there's a navigation bar with tabs for '예' (Example), '최근 사용' (Recent Usage), 'Google 드라이브' (Google Drive), 'GitHub', and '업로드' (Upload). The 'Recent Usage' tab is selected. Below the navigation bar is a search bar labeled '노트 필터링' (Note filtering) with a dropdown arrow icon. Underneath the search bar is a table with columns for '제목' (Title), '처음 엔 시간' (First run time), and '마지막 엔 시간' (Last run time). The table lists five notebooks:

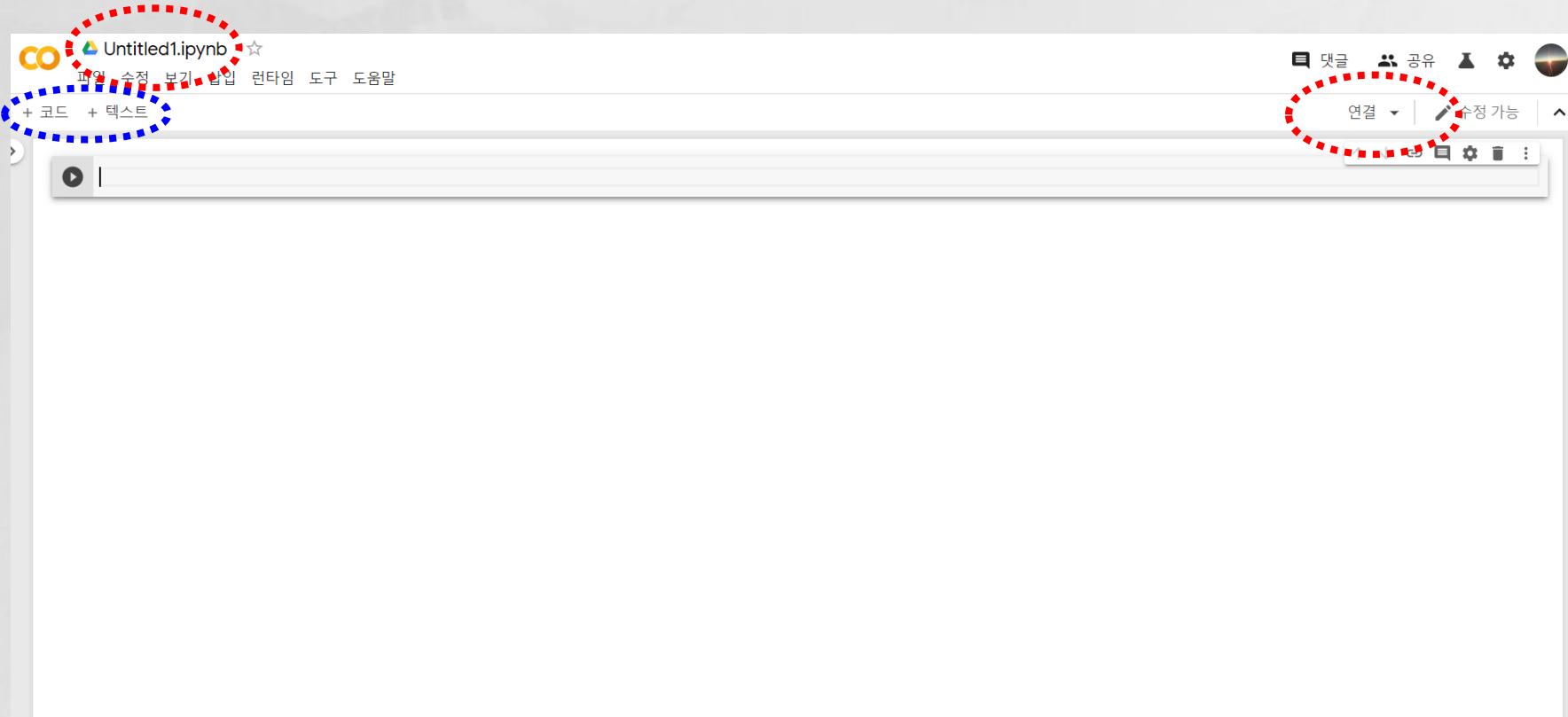
제목	처음 엔 시간	마지막 엔 시간
Colaboratory에 오신 것을 환영합니다	2020년 2월 5일	0분 전
401_Keras_AE.ipynb	7일 전	7일 전
402_Keras_VAE.ipynb	7일 전	7일 전
304_Stacked_CuDNNLSTM_Time_Series.ipynb	7일 전	7일 전
302_CuDNNLSTM_Time_Series.ipynb	7일 전	7일 전

At the bottom right of the interface, there are two buttons: '새 노트' (New Note) and '취소' (Cancel), with a red circle highlighting the '새 노트' button.

Google Colab

Google Colaboratory 4/5

- <https://colab.research.google.com>



Google Colab

Google Colaboratory 5/5

- <https://drive.google.com>

