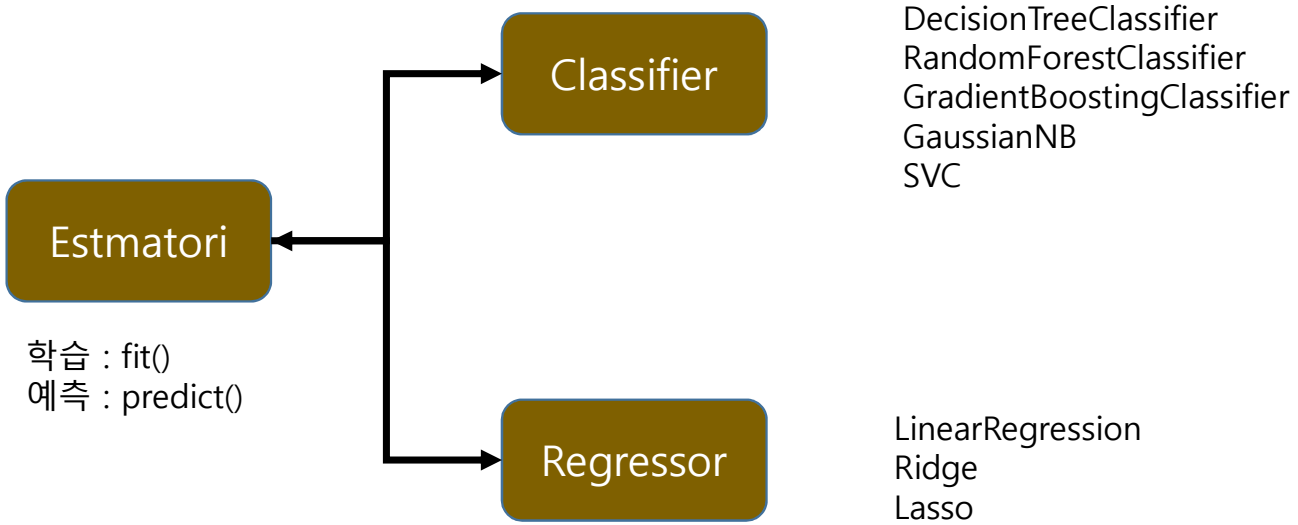
Scikit-learn 기반 프레임워크

- Estimator
  - 지도 학습의 모든 알고리즘을 구현한 클래스
  - fit()과 predict()를 내부에서 구현, 알고리즘을 평가하가너 하이퍼 파라미터 튜닝을 수행



하이퍼 파라미터 – 머신러닝 알고리즘별로 최적의 학습을 위해 직접 입력하는 파라미터

Scikit-learn 기반 프레임워크

- 사이킷런이 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(문자열을 숫자형 코드 값으로 인코딩, 정규화, 스케일링 등)
	sklearn.feature_selection	알고리즘에 큰 영향을 미치는 피처를 우선순위대로 셀렉션 작업을 수행하는 다양한 기능 제공
	sklearn.feature_extraction	텍스트 데이터나 이미지 데이터의 벡터화된 피처를 추출하는데 사용됨 예) 텍스트 데이터에서 Count Vectorizer나 Tf-Idf Vectorier 등을 생성하는 기능 제공 텍스트 데이터의 피처 추출은 sklearn.feature-extraction.text 모듈에 이미지 데이터의 피처 추출은 sklearn.feature_extraction.image 모듈에 지원 API 있음
피처 처리 & 차원 축소	sklearn.decomposition	차원 축소와 관련한 알고리즘을 지원하는 모듈임. PCA, NMF, Truncated SVD 등을 통해 차원 축소 기능을 수행할 수 있음
데이터 분리, 검증 & 파라미터 튜닝	sklearn.model_selection	교차 검증을 위한 학습용/테스트용 분리, 그리드 서치(Grid Search)로 최적 파라미터 추출 등의 API 제공

Scikit-learn 기반 프레임워크

- 사이킷런이 주요 모듈

분류	모듈명	설명
평가	sklearn.metrics	분류, 회귀, 클러스터링, 페어와이즈(Pairwise)에 대한 다양한 성능 측정 방법 제공 Accuracy, Precision, Recall, ROC-AUC, RMSE 등 제공
ML 알고리즘	sklearn.ensemble	앙상블 알고리즘 제공 랜덤 포레스트, 에이다 부스트, 그레디언트 부스팅 등을 제공
	sklearn.linear_model	선형 회귀, 릿지(Ridge), 라쏘(Lasso) 및 로지스틱 회귀 등 회귀 관련 알고리즘을 지원, 또한 SGD(Stochastic Gradient Descent) 관련 알로리즘도 제공
	sklearn.naïve_bayes	나이브베이즈 알고리즘 제공, 가우시안 NB, 다항분포 NB등
	skearn.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-계층, 계층형, DBSCAN 등)
유틸리티	sklearn.pipeline	차원 축소와 관련한 알고리즘을 지원하는 모듈임. PCA, NMF, Truncated SVD 등을 통해 차원 축소 기능을 수행할 수 있음

Scikit-learn 기반 프레임워크

➢ 분류나 회귀 연습용 예제 데이터

API명	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()	회귀용도이며, 당뇨 데이터 세트
datasets.load_digits()	분류 용도이며, 0에서 9까지 숫자의 이미지 픽셀 데이터 세트
datasets.load_iris()	분류 용도, 붓꽃에 대한 피처를 가진 데이터 세트

Fetch 계열의 명령 – 데이터의 크기가 커서 패키지에 처음부터 저장돼 있지 않고 인터넷에서 내려 받아 홈 디렉토리 아래의 scikit_learn_data라는 서브 디렉토리에 저장한 후 추후 불러들이는 데이터

- fetch_covtype() : 회귀 분석용 토지 조사 자료
- fetch_20newsgroups() : 뉴스 그룹 텍스트자료
- fetch_olivetti_faces() : 얼굴 이미지 자료
- fetch_lfw_people() : 얼굴 이미지 자료
- fetch_lfw_pairs() : 얼굴 이미지 자료
- fetch_rcv1() : 로이터 뉴스 말뭉치
- fetch_mldata() : ML 웹사이트에서 다운로드

Scikit-learn 기반 프레임워크

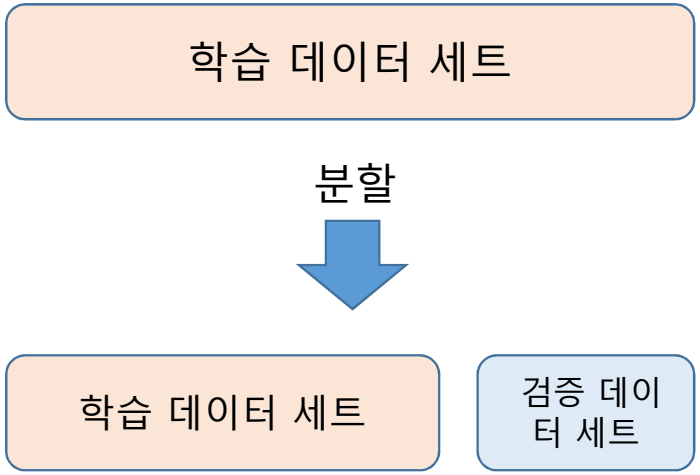
- 분류와 클러스터링을 위한 표본 데이터 생성기

API명	설명
datasets.make_classification()	분류를 위한 데이터 세트를 만듭니다. 특히 높은 상관도, 불필요한 속성 등의 노이즈 효과를 위한 데이터를 무작위로 생성해 줍니다.
datasets.make_blobs()	클러스터링을 위한 데이터 세트를 무작위로 생성해 줍니다. 군집 지정 개수에 따라 여러 가지 클러스터링을 위한 데이터 세트를 쉽게 만들어 줍니다.

Scikit-learn 기반 프레임워크

- 교차 검증
  - 과적합 : 모델이 학습 데이터에만 과도하게 최적화되어, 실제 예측은 다른 데이터로 수행할 경우에는 예측 성능이 떨어지는 것
  - 교차 검증 : 데이터 편중을 막기 위해서 별도의 여러 세트로 구성된 학습 데이터 세트와 검증 데이터 세트에서 학습과 평가를 수행하는 것

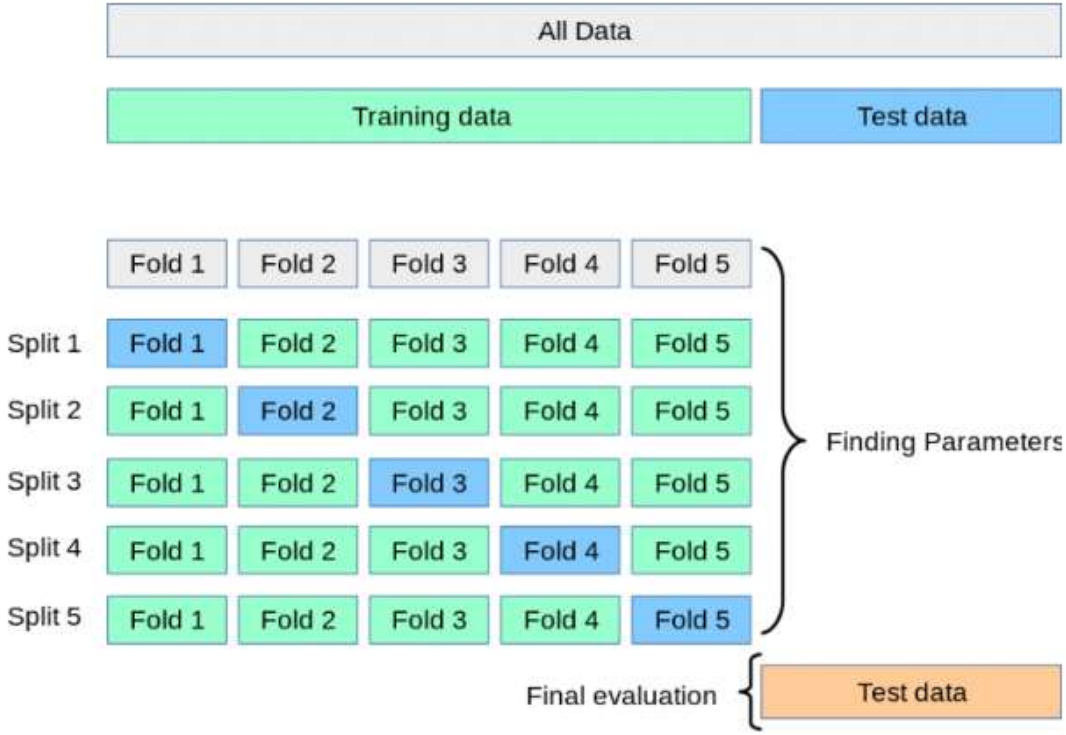
학습 데이터를 다시 분할하여 학습 데이터와 학습된 모델의 성능을 일차 평가하는 검증 데이터로 나눔



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

Scikit-learn 기반 프레임워크

- K 폴드 교차 검증
  - K개의 데이터 폴드 세트를 만들어서 K번만큼 각 폴드 세트에 학습과 검증 평가를 반복적으로 수행하는 방법
  - K 폴드 교차 검증 프로세스를 구현하기 위해 Kfold와 StratifiedKFold 클래스를 제공
  - Stratified K 폴드는 불균형한 분포도를 가진 레이블(결정 클래스) 데이터 집합을 위한 K 폴드 방식 : 레이블 데이터 분포도에 따라 학습/검증 데이터를 나누기 때문에 split()에 인자로 피처 데이터 세트뿐만 아니라 레이블 데이터 세트도 반드시 필요



1. 폴드 세트 설정
2. For루프에서 반복으로 학습 및 테스트 데이터의 인덱스 추출
3. 반복적으로 학습과 예측을 수행하고 예측 성능을 반환

Scikit-learn 기반 프레임워크

- cross_val_score()
  - 교차 검증 프로세스 (폴드 세트 설정, For루프에서 반복으로 학습 및 테스트 데이터의 인덱스 추출, 반복적으로 학습과 예측을 수행하고 예측 성능을 반환)를 한번에 수행할 수 있게 해주는 API

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

- Estimator : 사이킷런의 분류 알고리즘 클래스인 Classifier 또는 회귀 알고리즘 클래스인 Regressor
- X : 피처 데이터 세트
- Y : 레이블 데이터 세트
- scoring : 예측 성능 평가 지표
- cv : 교차 검증 폴드 수
- cv로 지정된 횟수만큼 scoring 파라미터로 지정된 평가 지표로 평가 결과값을 배열로 반환

Scikit-learn 기반 프레임워크

- GridSearchCV
  - 교차 검증과 최적 하이퍼 파리미터 튜닝을 수행 알고리즘의 예측 성능을 개선
  - Classifier나 Regressor 와 같은 알고리즘에 사용되는 하이퍼 파라미터를 순차적으로 입력하면서 편리하게 최적의 파라미터를 도출할 수 있는 방안을 제공
  - 교차 검증을 기반으로 하이퍼 파라미터의 최적 값을 찾게 해줌
  - Cross-validatio을 위한 학습/텟트 세트로 자동으로 분할한 뒤에 하이퍼 파라미터 그리드에 기술된 모든 파라미터를 순차적으로 적용해 최적의 파라미터를 찾을 수 있게 해줍니다.

GridSerchCV 생성자 파라미터

- Estimator : Classifier ,Regressor, pipelin이 사용될 수 있습니다.
- param_grid : key + 리스트 값을 가지는 딕셔너리가 주어집니다. Estimator의 튜닝을 위해 파라미터명과 사용될 여러 파라미터 값을 지정합니다.
- scoring : 예측 성능을 측정할 평가 방법을 지정합니다.
- cv : 교차 검증을 위해 분할되는 학습/테스트 세트의 개수를 지정합니다.
- refit : 디폴트가 True이며 True로 생성 시 가장 최적의 하이퍼 파라미터를 찾은 뒤 입력된 estimator 객체를 해당 하이퍼 파라미터로 재학습시킵니다.

- GridSearchCV 객체의 fit()을 수행하면 최고 성능을 나타낸 하이퍼 파라미터 값과 그때의 평가 결과 값이 각각 best_params_, best_score_ 속성에 기록됩니다.
- refit =True이면 GridSearchCV가 최적 성능을 나타내는 하이퍼 파라미터로 Estimator를 학습해 best_estimator로 저장합니다.

Scikit-learn 기반 프레임워크

- 데이터 전처리 - 인코딩
  - 피처 값 중 Null 값이 얼마 되지 않는다면 피처의 평균값 등으로 간단히 대체할 수 있습니다.
  - Null값이 대부분이라면 해당 피처는 드롭하는 것이 좋습니다.
  - 사이킷런의 머신러닝 알고리즘은 문자열 값을 입력 값으로 허용하지 않습니다.
  - 모든 문자열 값으로 인코딩돼서 숫자 형으로 변환해야 합니다.
  - 주민번호나 단순 문자열 아이디와 같은 식별자 피처는 단순히 데이터 로우를 식별하는 용도로 사용되기 때문에 예측에 중요한 요소가 될 수 없으며 알고리즘을 오히려 복잡하게 만들고 예측 성능을 떨어뜨립니다.

Scikit-learn 기반 프레임워크

- LabelEncoder
  - 카테고리 피처를 코드형 숫자 값으로 변환
  - fit()과 transform()을 호출해 레이블 인코딩을 수행
  - 인코딩 수행 후 LabelEncoder객체의 classes_ 속성값은 0번부터 순서대로 변환된 인코딩 값에 대한 원본값을 가지고 있습니다.
  - Inverse_transform()을 통해 인코딩된 값을 다시 디코딩할 수 있습니다.
  - 레이블 인코딩이 일괄적인 숫자 값으로 변환되면서 단순 코드값이 숫자 값에 따른 순서나 중요도로 인식돼어 예측 성능이 떨어지는 경우가 발생하므로 레이블 인코딩은 선형 회귀와 같은 알고리즘에는 적용하지 않아야 합니다
  - 트리 계열의 알고리즘은 숫자의 특성을 반영하지 않으므로 레이블 인코딩 값을 사용해도 문제가 없습니다.

Scikit-learn 기반 프레임워크

- One-Hot Encoder
  - 피처 값의 유형에 따라 새로운 피처를 추가해 고유 값에 해당하는 컬럼에만 1을 표시하고 나머지 컬럼에는 0을 표시하는 방식입니다.
  - OneHotEncoder로 변환하기 전에 모든 문자열 값이 숫자형 값으로 변환되어야 하며, 입력값으로 2차원 데이터 형태로 만들어 주어야 합니다.
  - Pandas에는 원-핫 인코딩을 더 쉽게 지원하는 API가 있습니다. get_dummies()는 문자열 카테고리 값을 숫자 형으로 변환할 필요 없이 바로 변환할 수 있습니다.

Scikit-learn 기반 프레임워크

- 피처 스케일링과 정규화
  - Feature scaling : 서로 다른 변수의 값 범위를 일정한 수준으로 맞추는 작업
  - Standardization : 데이터의 피처 각각이 평균이 0이고 분산이 1인 가우시안 정규 분포를 가진 값으로 변환
  - Normalization : 선형대수에서의 정규화 개념이 적용됐으며, 개별 벡터의 크기를 맞추기 위해 변환하는 것을 의미
  - 사이킷런에서 구현한 RBF 커널을 이용하는 서포트 벡터 머신(Support Vector Machine)이나 선형 회귀(Linear Regression), 로지스틱 회귀(Logistic Regression)는 데이터가 가우시안 분포를 가지고 있다고 가정하고 구현됐기 때문에 사전에 표준화를 적용하는 것은 예측 성능 향상에 중요한 요소가 될 수 있습니다
  - StarndarScaler : 개별 피처를 평균이 0이고 분산이 1인 값으로 변환
  - MinMxScaler : 데이터값을 0과 1사이의 범위 값으로 변환
  - Scaler 객체를 이용해 학습 데이터 세트로 fit()과 transform()을 적용하면 테스트 데이터 세트로는 다시 fit()을 수행하지 않고 학습 데이터 세트로 fit()을 수행한 결과를 이용해 transform() 변환을 적용해야 한다