분류 알고리즘 이론 및 실습

김지선

Contents

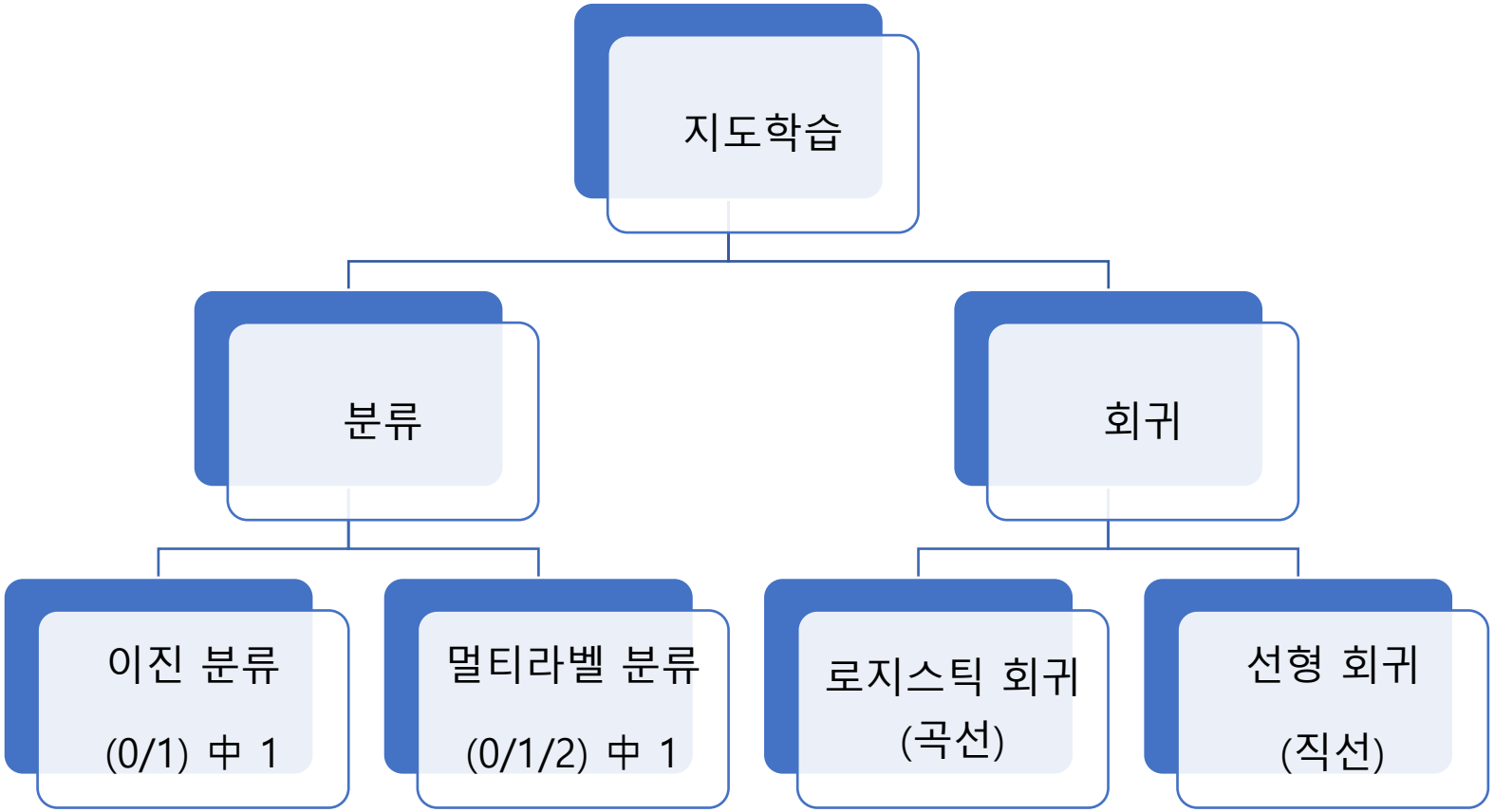
- 사이킷런
- 머신러닝 Background

복습

- Feature(X) – 정답을 제외한 나머지 칼럼(data in model)
- Target(y) - 예측하려는 목표(class, label)
  - Y=Wx+b(선형회귀)

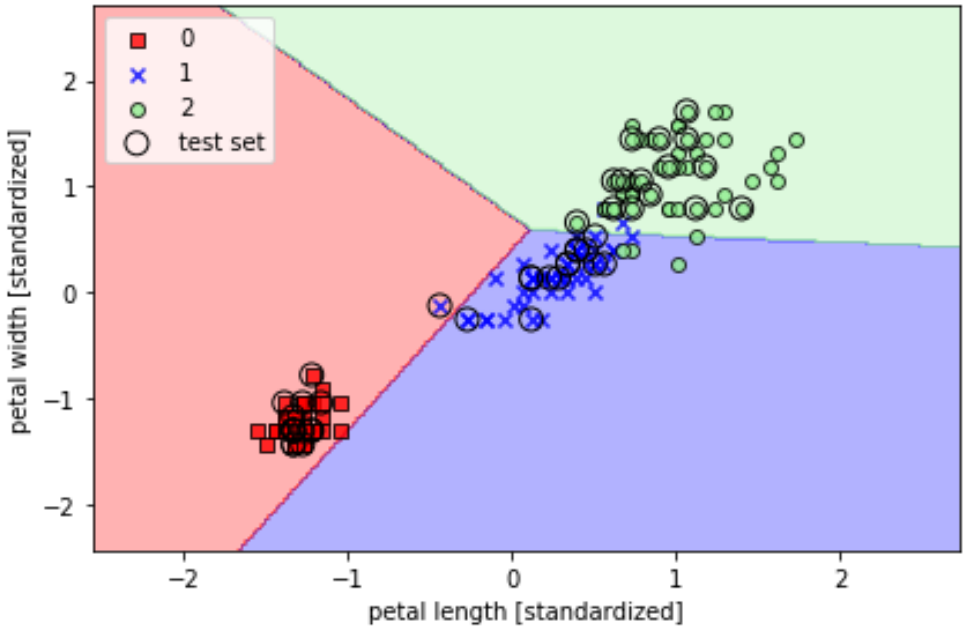
	나이	체중(kg)	유전병 여부	당뇨 여부
0	22	54	0	1
1	43	68	1	0
2	34	80	1	1
3	64	43	1	1

복습



사이킷런(Scikit-learn)이란?

- 머신러닝에서 주로 사용되는 라이브러리
  - conda install scikit-learn
  - pip install scikit-learn



https://benghak.github.io/2020-01-28-ML_chaper_03/

사이킷런 기본 모듈/함수

- From 모듈 import 이름
  - ➪ sklearn.datasets –자체 제공 데이터
  - ➪ sklearn.tree - 트리 관련 클래스들
  - ➪ sklearn.model_selection – 데이터 분리 및 평가

- ML 모델 사용
  - 객체=모델
  - 객체.fit(train data, train label)
  - 객체.predict(test data)
  - accuracy_score(test label, predict label)

사이킷런 예측 프로세스- fit, predict

1. 데이터 세트 분리
2. 모델 학습(fit)
3. 테스트 데이터 예측(predict)
4. 정확도 평가

사이킷런 분류 문제 예시

- 붓꽃 데이터 레이블 결정트리 알고리즘으로 예측

```
# 학습과 테스트 데이터 분리 - X: data,y: label
X_train,X_test,y_train,y_test=train_test_split(iris.data,iris.target,test_size=0.3,random_state=121)

# 객체에 모델 할당
DTmodel=DecisionTreeClassifier(random_state=11)

# 학습 수행 - (학습 데이터, 학습 데이터의 라벨)
DTmodel.fit(X_train,y_train)
```

```
DecisionTreeClassifier(random_state=11)
```

```
# 테스트 세트로 예측
predict_label=DTmodel.predict(X_test)

# 모델이 예측한 데이터의 정확도 출력 -by.y_test(실제 라벨)
print('정확도: ',accuracy_score(y_test,predict_label))
```

```
정확도:  0.9555555555555556
```

사이킷런 주요 모듈 (1)

- 지도학습에 분류와 회귀의 다양한 알고리즘 - fit, predict 사용

분류	모듈명	설명
예제데이터	sklearn.datasets	예제 데이터
피처처리	sklearn.preprocessing	전처리 기능 인코딩,정규화,스케일링
	sklearn.feature_selection	Feature를 우선순위대로 selection
	sklearn.feature_extraction	벡터화된 피처 추출하는데 사용
피처처리 & 차원축소	sklearn.decomposition	차원축소 관련 알고리즘
데이터 분리, 검증 & 파라미터 튜닝	sklearn.model_selection	교차검증을 위한 데이터 분리, 그리드 서치/최적 파라미터 추출
평가	sklearn.metrics	성능 측정 방법 제공

사이킷런 주요 모듈 (2)

분류	모듈명	설명
ML 알고5리즘	sklearn.ensemble	앙상블 알고리즘 -랜덤 포레스트,에이다/그래디언트
	sklearn.linear_model	회귀 관련 알고리즘/SGD - 선형/로지스틱 회귀, 라쏘, 릿지
	sklearn.naïve_bayes	나이브 베이즈 알고리즘 - 가우시안 NB, 다항 분포 NB
	sklearn.neighbors	최근접 이웃 알고리즘(KNN)
	sklearn.svm	서포트 벡터 머신 알고리즘
	sklearn.tree	의사 결정 트리 알고리즘
	sklearn.cluster	비지도 클러스터링 알고리즘 - kmeans, 계층형, DBSCAN 등
유틸리티	sklearn.pipeline	여러 기능 묶인 유틸리티

교차 검증

- K 폴드 교차 검증 – 과적합 예방
- Stratified K 폴드 – 레이블값 분포 반영

K 폴드 교차 검증

- 교차검증으로 과적합을 방지
- 교차검증한 스코어의 평균으로 score 측정

- 코드
  - `From sklearn.model_selection import Kfold`
  - `Kfold = Kfold(n_splits=n) #n개의 덩어리로 나눠줌`
  - `Kfold.split(features) #교차검증시 인덱스 설정해줌`

Stratified K 폴드

- 불균형한 분포도의 레이블을 가진 데이터를 위한 k 폴드 방식
- 레이블 분포도 유지시켜줌
- 분류 문제에서 교차검증으로 주로 사용(회귀X)

EX) 대출사기 이진 분류 (대출사기 -1, 정상 대출– 0)
대출사기 비율이 0.001%와 같이 편향 되어 있을 때

Stratified K 폴드

- 불균형한 분포도의 레이블을 가진 데이터를 위한 k 폴드 방식
- 레이블 분포도 유지시켜줌
- 분류 문제에서 교차검증으로 주로 사용(회귀X)

EX) 대출사기 이진 분류 (대출사기 -1, 정상 대출– 0)
대출사기 비율이 0.001%와 같이 편향 되어 있을 때

- 코드
  - from sklearn.model_selection import Kfold
  - kfold = KFold(n_splits=n) // 동일
  - kfold.split(feature,label) // label 값만 추가로 줌

교차검증 API - cross_val_score

- Cross_val_score(estimator,X,y,cv)
  - Cv – fold number(반복 횟수)
  - Estimator – classifier, regressor

- classifier ⇨ K 폴드
- Regressor ⇨ Stratified K 폴드
  +회귀는 레이블 분포 불균형에 영향을 받지 않기 때문

GridSearch CV

- 촘촘하게 파라미터를 입력하면서 테스트
- 파라미터 집합을 만들고 순차적 적용해주는 API

  - 객체=GridSearchCV(알고리즘, 파라미터 집합, refit=True)
  - 객체.cv_results_ : 결과 세트
  - 객체.best_estimator_ : 최적의 하이퍼 파라미터 저장

과제

- 타이타닉 데이터 분석
  https://www.kaggle.com/competitions/titanic