scikit-learn-2

김 종 우

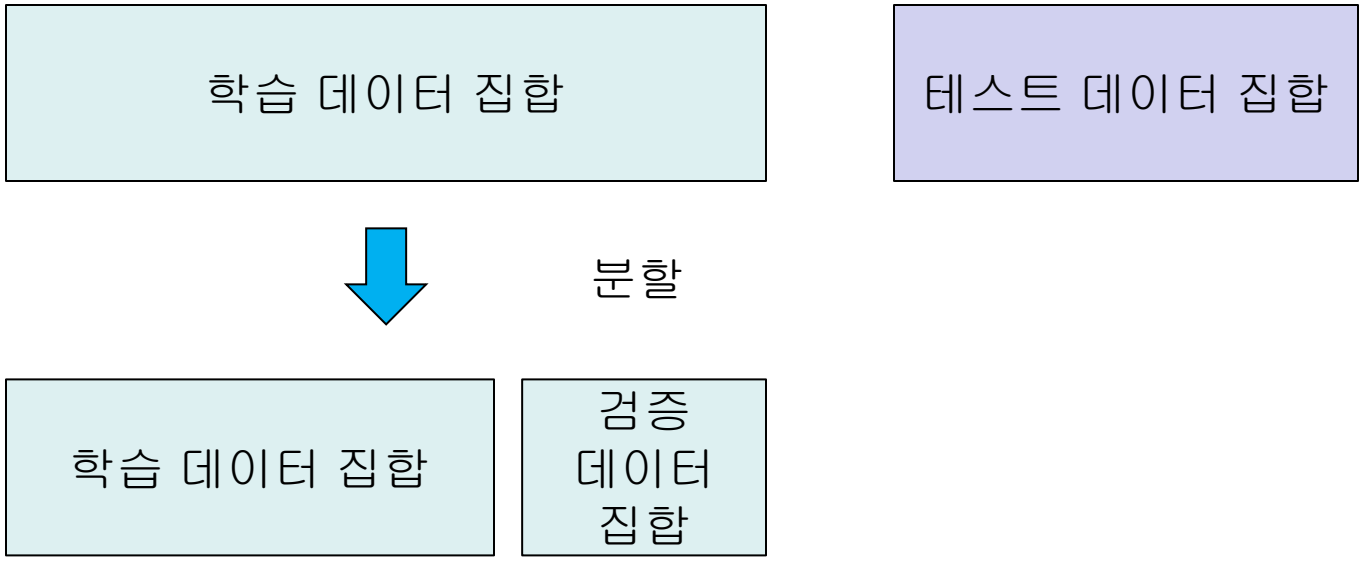
한양대학교

목차

- 개요
- 첫 번째 머신러닝 만들어보기
- 사이킷런의 기반 프레임워크
- Model Selection 모듈
- 데이터 전처리

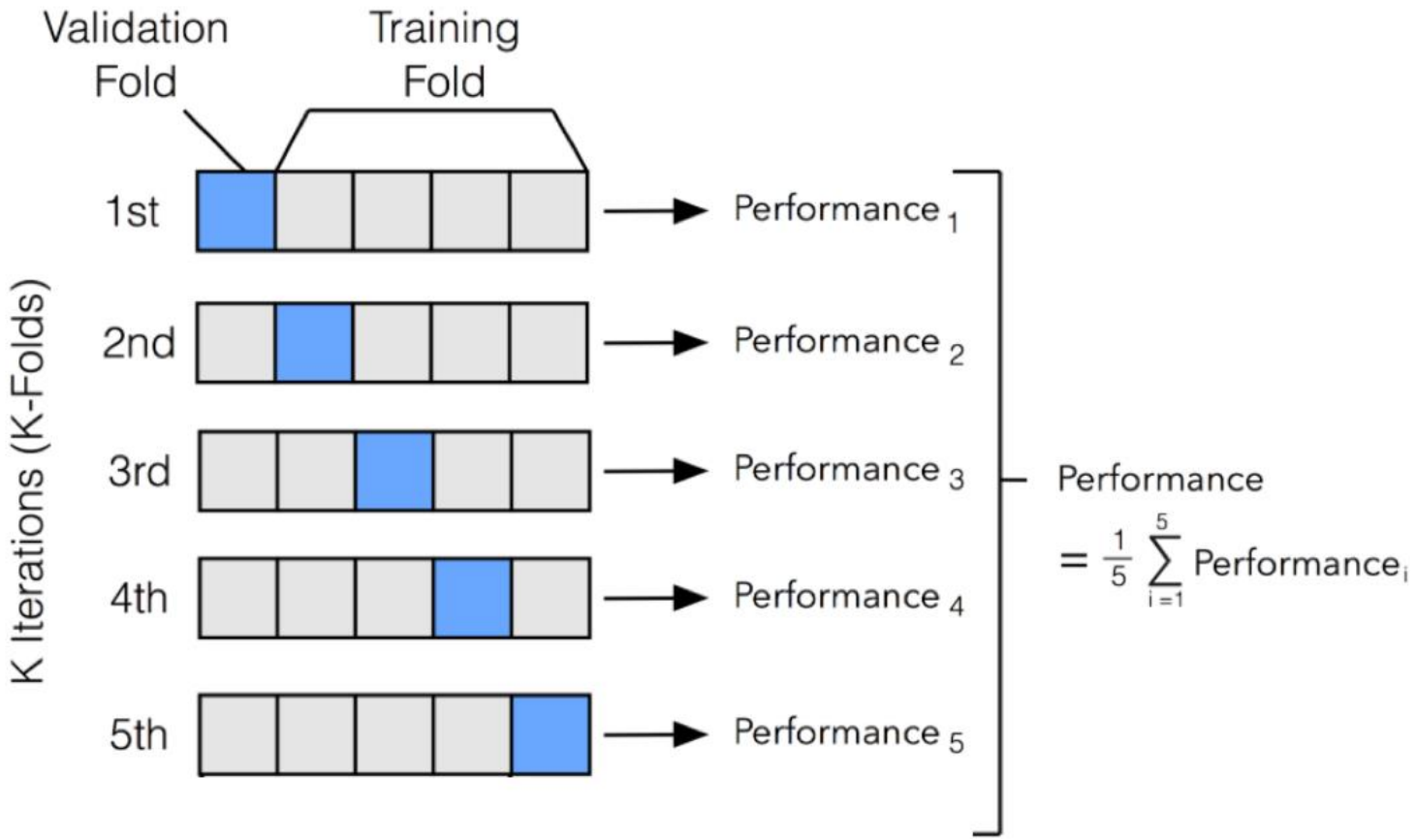
Model Selection 모듈

- 많은 기계학습 모델들이 검증 데이터 집합을 활용하여 하이퍼 파라미터 튜닝 등의 모델 최적화



Model Selection 모듈

- K 폴드 교차 검증



Model Selection 모듈

- K-폴드 교차 검증

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import KFold
import numpy as np

iris = load_iris()
features = iris.data
label = iris.target
dt_clf = DecisionTreeClassifier(random_state=156)

kfold = KFold(n_splits=5)
cv_accuracy = []
print('붓꽃 데이터 세트 크기:',features.shape[0])
```

Model Selection 모듈

- K-폴드 교차 검증

```
n_iter = 0

for train_index, test_index  in kfold.split(features):
    X_train, X_test = features[train_index], features[test_index]
    y_train, y_test = label[train_index], label[test_index]
    #학습 및 예측
    dt_clf.fit(X_train , y_train)
    pred = dt_clf.predict(X_test)
    n_iter += 1
    accuracy = np.round(accuracy_score(y_test,pred), 4)
    train_size = X_train.shape[0]
    test_size = X_test.shape[0]
    print('\n#{0} 교차 검증 정확도 :{1}, 학습 데이터 크기: {2}, 검증 데이터 크기: {3}'
          .format(n_iter, accuracy, train_size, test_size))
    print('#{0} 검증 세트 인덱스:{1}'.format(n_iter,test_index))
    cv_accuracy.append(accuracy)
```

Model Selection 모듈

- K-폴드 교차 검증

#1 교차 검증 정확도 :1.0, 학습 데이터 크기: 120, 검증 데이터 크기: 30
#1 검증 세트 인덱스:[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26 27 28 29]

#2 교차 검증 정확도 :0.9667, 학습 데이터 크기: 120, 검증 데이터 크기: 30
#2 검증 세트 인덱스:[30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
48 49 50 51 52 53
 54 55 56 57 58 59]
…

Model Selection 모듈

- K-폴드 교차 검증

```
# 개별 iteration별 정확도를 합하여 평균 정확도 계산
print('\n## 평균 검증 정확도:', np.mean(cv_accuracy))
```

```
## 평균 검증 정확도: 0.9
```

Model Selection 모듈

- Stratified K-폴드 교차 검증
  - 층화 K-폴드

```
import pandas as pd

iris = load_iris()

iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
iris_df['label']=iris.target
iris_df['label'].value_counts()
```

```
2    50
1    50
0    50
Name: label, dtype: int64
```

Model Selection 모듈

- Stratified K-폴드 교차 검증

```
kfold = KFold(n_splits=3)

n_iter =0
for train_index, test_index  in kfold.split(iris_df):
    n_iter += 1
    label_train= iris_df['label'].iloc[train_index]
    label_test= iris_df['label'].iloc[test_index]
    print('## 교차 검증: {0}'.format(n_iter))
    print('학습 레이블 데이터 분포:\n', label_train.value_counts())
    print('검증 레이블 데이터 분포:\n', label_test.value_counts())
```

Model Selection 모듈

- Stratified K-폴드 교차 검증

```
## 교차 검증: 1
학습 레이블 데이터 분포:
 2    50
1    50
Name: label, dtype: int64
검증 레이블 데이터 분포:
 0    50
Name: label, dtype: int64
....
```

Model Selection 모듈

- Stratified K-폴드 교차 검증

```
from sklearn.model_selection import StratifiedKFold

skf = StratifiedKFold(n_splits=3)
n_iter=0

for train_index, test_index in skf.split(iris_df, iris_df['label']):
    n_iter += 1
    label_train= iris_df['label'].iloc[train_index]
    label_test= iris_df['label'].iloc[test_index]
    print('## 교차 검증: {0}'.format(n_iter))
    print('학습 레이블 데이터 분포:\n', label_train.value_counts())
    print('검증 레이블 데이터 분포:\n', label_test.value_counts())
```

Model Selection 모듈

- Stratified K-폴드 교차 검증

```
## 교차 검증: 1
학습 레이블 데이터 분포:
 2    34
1    33
0    33
Name: label, dtype: int64
검증 레이블 데이터 분포:
 1    17
0    17
2    16
```

Model Selection 모듈

- Stratified K-폴드 교차 검증

```
dt_clf = DecisionTreeClassifier(random_state=156)

skfold = StratifiedKFold(n_splits=3)
n_iter=0
cv_accuracy=[]

for train_index, test_index  in skfold.split(features, label):
    X_train, X_test = features[train_index], features[test_index
    y_train, y_test = label[train_index], label[test_index]
    dt_clf.fit(X_train , y_train)
    pred = dt_clf.predict(X_test)
```

Model Selection 모듈

- Stratified K-폴드 교차 검증

```
    n_iter += 1
    accuracy = np.round(accuracy_score(y_test,pred), 4)
    train_size = X_train.shape[0]
    test_size = X_test.shape[0]
    print('\n#{0} 교차 검증 정확도 :{1}, 학습 데이터 크기: {2}, 검증 데이터 크기: {3}'
          .format(n_iter, accuracy, train_size, test_size))
    print('#{0} 검증 세트 인덱스:{1}'.format(n_iter,test_index))
    cv_accuracy.append(accuracy)
```

```
print('\n## 교차 검증별 정확도:', np.round(cv_accuracy, 4))
print('## 평균 검증 정확도:', np.mean(cv_accuracy))
```

```
## 교차 검증별 정확도: [0.98 0.94 0.98]
## 평균 검증 정확도: 0.9666666666666667
```

Model Selection 모듈

- cross_val_score()
  - 교차 검증을 보다 간편하게
  - (1) 폴드 집합 설정, (2) for 루프를 통한 반복 추출 과 학습, 정확도 예측, (3) 정확도 평균

Model Selection 모듈

- cross_val_score()

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
from sklearn.datasets import load_iris

iris_data = load_iris()
dt_clf = DecisionTreeClassifier(random_state=156)

data = iris_data.data
label = iris_data.target

scores = cross_val_score(dt_clf , data , label , scoring='accuracy',cv=3)
print('교차 검증별 정확도:',np.round(scores, 4))
print('평균 검증 정확도:', np.round(np.mean(scores), 4))
```

```
교차 검증별 정확도: [0.98 0.94 0.98]
평균 검증 정확도: 0.9667
```

** 분류의 경우 Straified K-폴드 방식으로 분할*

scikit-learn Lab 1

1. scikit-learn에서 제공하는 breast_cancer 데이터 집합을 load_breast_cancer()를 이용하여 읽어들이시오.
2. 목표 변수값이 어떻게 배정되었는지, 또 각 목표변수의 이름이 무엇인지 확인하시오.
3. 입력변수와 목표변수로 이루어진 데이터프레임 breast_df를 생성하시오.

scikit-learn Lab 1

4. train_test_split() 함수를 이용하여 데이터집합을 훈련 데이터집합과 테스트 데이터집합으로 나누고, 성능을 평가하시오.
   - 단, 테스트 데이터 집합은 전체의 30%로 하시오.

scikit-learn Lab 1

5. Kfold 클래스를 이용하여, k-fold 교차 검증을 통해서 성능을 평가하시오.
   - 단, k=5로 하시오

6. StratifiedKfold 클래스를 이용하여, Stratified k-fold 교차 검증을 통해서 성능을 평가하시오.
   - 단, k=5로 하시오

scikit-learn Lab 1

7. cross_val_score() 함수를 이용하여 교차 검증을 통해서 성능을 평가하시오.
   - 단, k=5로 하시오