데이터 과학

L10.1: kNN Practice

Kookmin University

Iris dataset

- 아이리스(붓꽃) 데이터
  - 붓꽃 종류별로 꽃받침과 꽃잎의 길이 및 너비를 측정한 데이터

Iris Setosa

Iris Versicolour

Iris Virginica

Iris dataset

- 아이리스(붓꽃) 데이터
  - 붓꽃 종류별로 꽃받침과 꽃잎의 길이 및 너비를 측정한 데이터
  - https://archive.ics.uci.edu/ml/datasets/Iris

```
4.6,3.2,1.4,0.2,Iris-setosa
5.3,3.7,1.5,0.2,Iris-setosa
5.0,3.3,1.4,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
5.5,2.3,4.0,1.3,Iris-versicolor
```

데이터 다운로드

```
!wget https://archive.ics.uci.edu/static/public/53/iris.zip

!unzip iris.zip
```

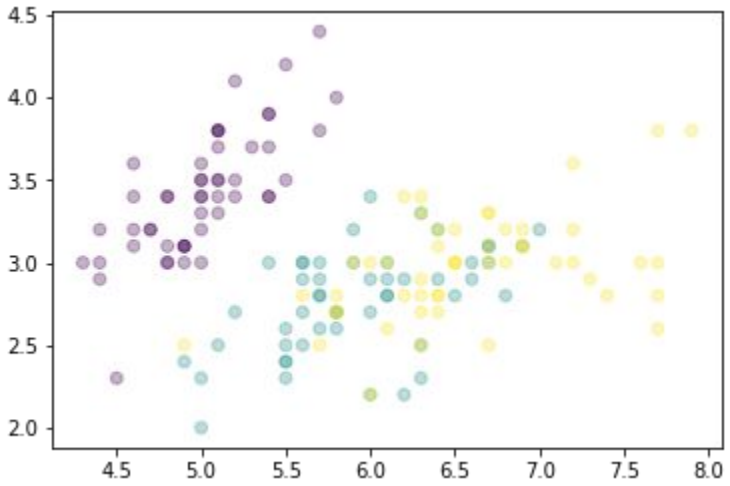
- **wget**: url로부터 파일을 다운로드 받는 쉘 명령어
- **unzip**: zip 압축 파일을 해제하는 쉘 명령어

데이터 불러오기

```
import numpy as np
X = []
y = []
for line in open("iris.data", "r"):
   line = line.strip()
   if line != "":
       tokens = line.split(",")
       X.append([float(t) for t in tokens[:4]])
       y.append(tokens[4])

y_labels = list(set(y))
y = [y_labels.index(a) for a in y]

X = np.array(X)
y = np.array(y)
```

데이터 살펴보기

```
import matplotlib.pyplot as plt

plt.scatter(X[:,0], X[:,1], c=y, alpha=0.3)
plt.show()
```



데이터 분리하기

- train data와 test data로 분리

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=41)
```

KNeighborsClassifier

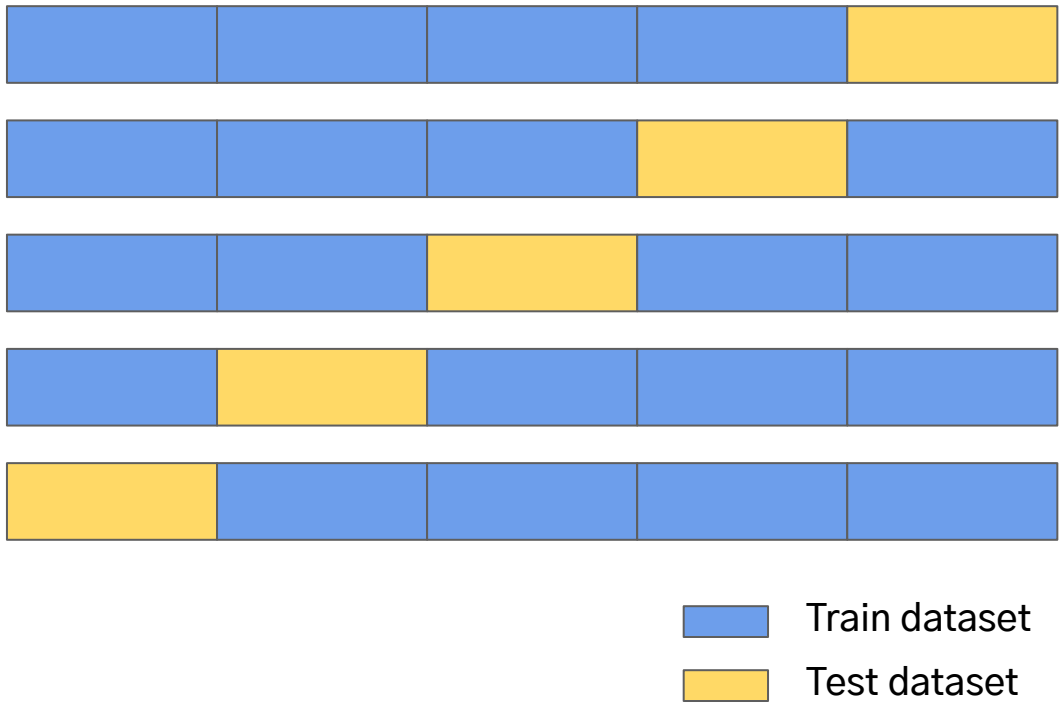
- sklearn의 KNeighborsClassifier 사용해보기

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

y_train_pred = knn.predict(X_train)
y_test_pred = knn.predict(X_test)

print("train accuracy:", accuracy_score(y_train_pred, y_train))
print("test accuracy:", accuracy_score(y_test_pred, y_test))
```

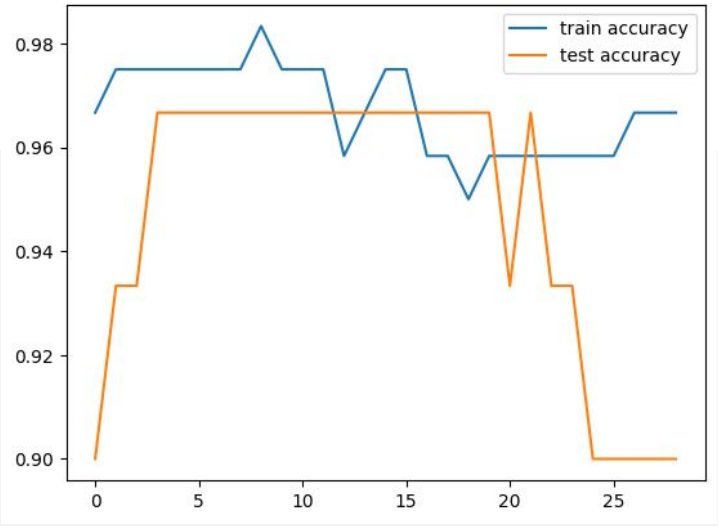
```
train accuracy: 0.9833333333333333
test accuracy: 0.9666666666666667
```

최적의 k 찾기

- n-fold cross validation: 데이터셋을 **n개의 서브데이터셋으로 분할**하고, 각 서브데이터셋을 test 데이터셋으로 사용하여 성능 확인



최적의 k 찾기




```
from sklearn.model_selection import cross_val_score
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt

scores = []
test_scores = []

for k in range(1,30):
    knn = KNeighborsClassifier(n_neighbors=k)
    score = cross_val_score(knn, X_train, y_train, cv=10, scoring="accuracy")
    scores.append(score.mean())

    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    test_scores.append(accuracy_score(y_test, y_pred))

plt.plot(scores, label="train accuracy")
plt.plot(test_scores, label="test accuracy")
plt.legend()

plt.show()
```

Questions?