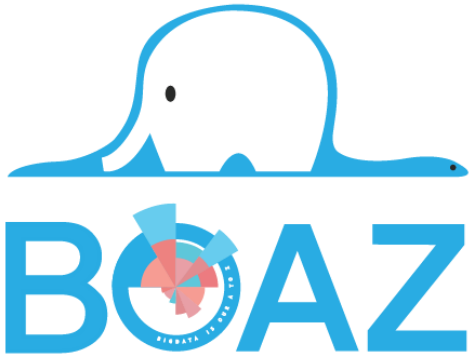


2017.09.21

K-nn

목 차

1. Scikit – Learn 의 전처리 기능

2. K – nearest neighbors 개념

2. 실습

Scikit – Learn의 전처리 기능

<Scikit – Learn의 전처리 기능>

- 스케일링(Scaling) : 자료의 크기 조정

- 인코딩 (Encoding) : 카테고리 값의 정수 표현

- Imputation : 결손 데이터(missing data) 처리

- Transform : 데이터 변환

Scikit – Learn의 전처리 기능

< 스케일링 (Scaling)>

- scale(x) : Standard Normal Gaussian 기본 스케일

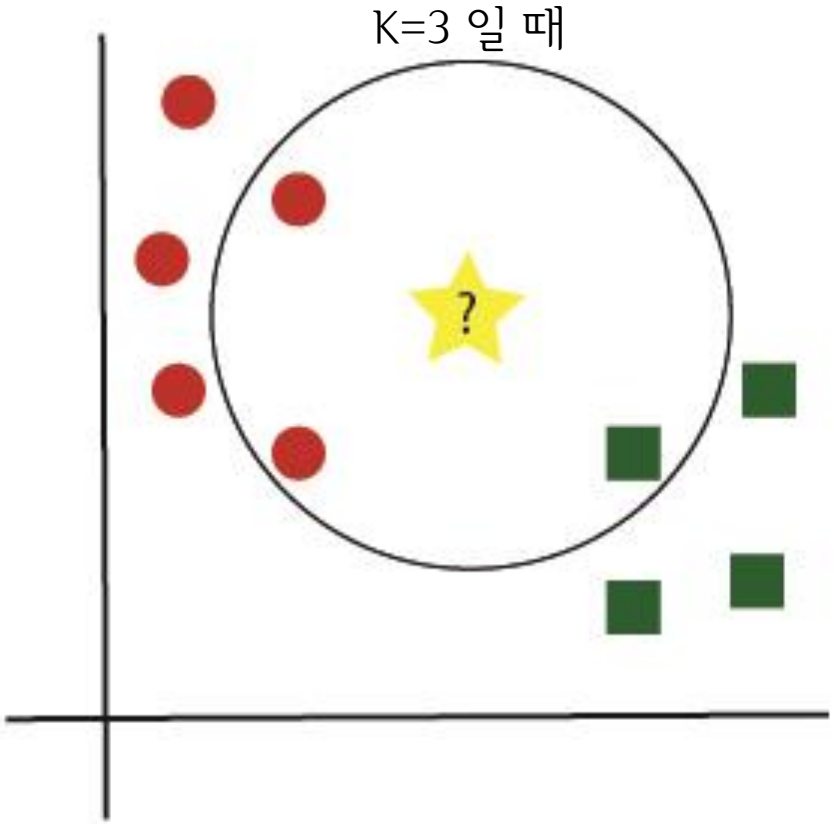
- normalize (x)

: 다차원 독립 변수 벡터가 있을 때 각 벡터 원소들의 상대적 크기만 중요한 경우에 사용

스케일링과 달리 개별 데이터의 크기를 모두 같게 만들기 위한 변환

따라서 개별 데이터에 대해 서로 다른 변환 계수가 적용

K – nearest neighbors

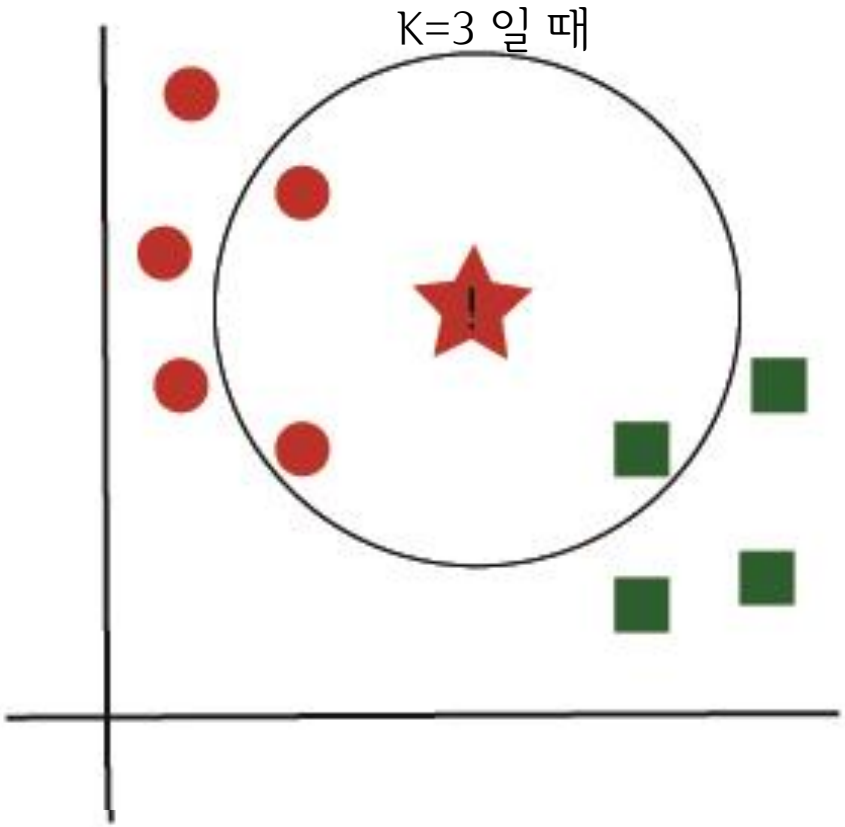


새로운 data인 별표가 들어왔을 때

유사도 측정방법을 통해

수가 많은 집단에 속하게 된다

K – nearest neighbors



새로운 data인 별표가 들어왔을 때

유사도 측정방법을 통해

수가 많은 집단에 속하게 된다

K – nearest neighbors

<유사도 측정방법>

- Euclidean 거리 ; p=2 일 때

$$\|p-q\| = \sqrt{(p-q)\cdot(p-q)} = \sqrt{\|p\|^2 + \|q\|^2 - 2p\cdot q}$$

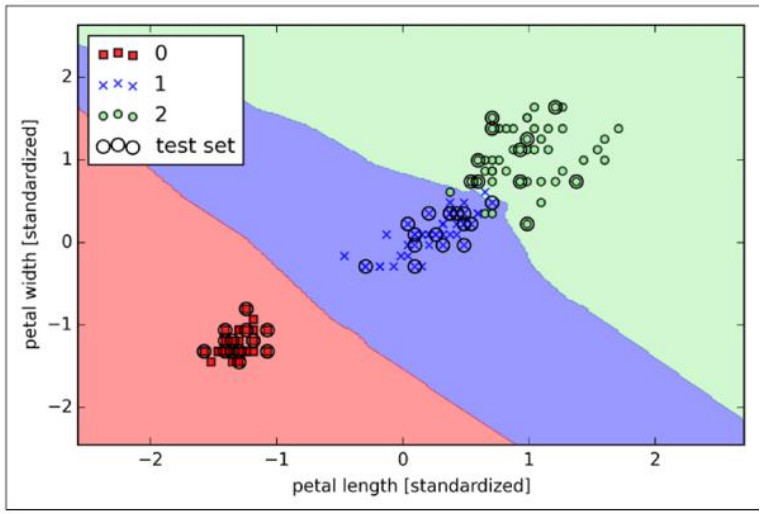
- Manhattan 거리 ; p=1 일 때
  : 두 개의 k-차원 실수 벡터 간 거리

$$d_1(p,q) = \|p-q\| = \sum_{i=1}^{n} |p_i - q_i|$$

K – nearest neighbors

<K – nn 사용 코드>

```
>>> from sklearn.neighbors import KNeighborsClassifier
>>> knn = KNeighborsClassifier(n_neighbors=5, p=2,
...                            metric='minkowski')
>>> knn.fit(X_train_std, y_train)
>>> plot_decision_regions(X_combined_std, y_combined,
...                       classifier=knn, test_idx=range(105,150))
>>> plt.xlabel('petal length [standardized]')
>>> plt.ylabel('petal width [standardized]')
>>> plt.show()
```



K – nearest neighbors

<실습!!!>