

# [Lab 4] 4주차 실습



## 과제 설명.

Q. 1-nearest neighbors classifier를 구현하고 iris 데이터를 이용하여 성능을 테스트 하시오.

Q. 5-cross validation 방법으로 성능을 평가하시오.

\* 자료실에 업로드 된 iris 데이터 파일을 이용하면 클래스 레이블이 정수로 되어 있어서 파일을 읽어 Numpy 배열에 저장하기 쉽습니다.

- 주의: 이번 실습에서는 지난 실습 과제와 같이 scikit\_learn 함수는 사용하지 않고 파이썬 함수를 사용하여 작성해야 합니다.
- 옆의 과제 예시 화면의 형태로 코드에서 결과값이 확인이 되어야합니다.

• 데이터 참고 : <https://archive.ics.uci.edu/ml/datasets/iris>

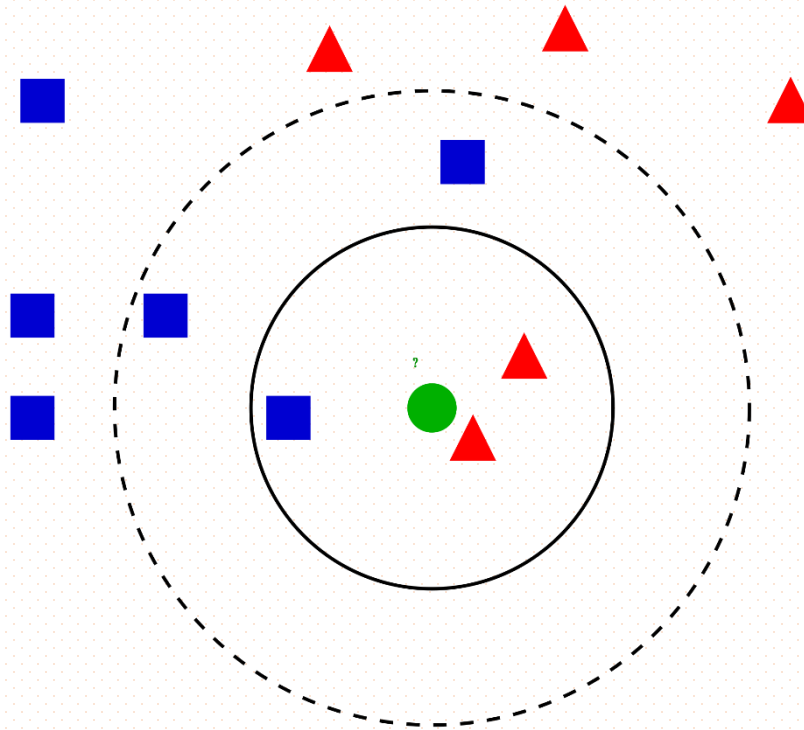
|    |   |     |
|----|---|-----|
| 1  | : | 94% |
| 2  | : | 93% |
| 3  | : | 92% |
| 4  | : | 91% |
| 5  | : | 90% |
| 평균 | : | 94% |



- 제출 기한
  - 1차 : 다음 실습수업시간 전일 자정까지 (10점 만점 채점)
  - 2차 : 2주 후 실습수업시간 전일 자정까지(7점 만점 채점)
  - 이후 제출은 미제출로 간주(점수 없음)
  
- 제출 방법
  - 1) Source 코드
  - 2) 코드에 대한 설명과 실행 결과 화면 캡처해서 작성한 보고서
  - 3) 1)과 2)를 폴더에 넣은 후 압축하여 사이버 캠퍼스에 제출.
  - 4) 폴더 및 파일명 : ML\_과제번호\_학번\_이름
  - 5) 과제 번호는 주차를 작성하시면 됩니다.

# KNN은 무엇인가?

- 패턴 인식에서,  **$k$ -최근접 이웃 알고리즘**(또는 줄여서  **$k$ -NN**)은 분류나 회귀에 사용되는 방식이다. (위키디피아, [https://ko.wikipedia.org/wiki/K-최근접\\_이웃\\_알고리즘](https://ko.wikipedia.org/wiki/K-최근접_이웃_알고리즘))
  - $k$ -NN 분류에서 출력은 소속된 항목이다. 객체는  $k$ 개의 최근접 이웃 사이에서 가장 공통적인 항목에 할당되는 객체로 과반수 의결에 의해 분류된다( $k$ 는 양의 정수이며 통상적으로 작은 수). 만약  $k = 1$  이라면 객체는 단순히 하나의 최근접 이웃의 항목에 할당된다.



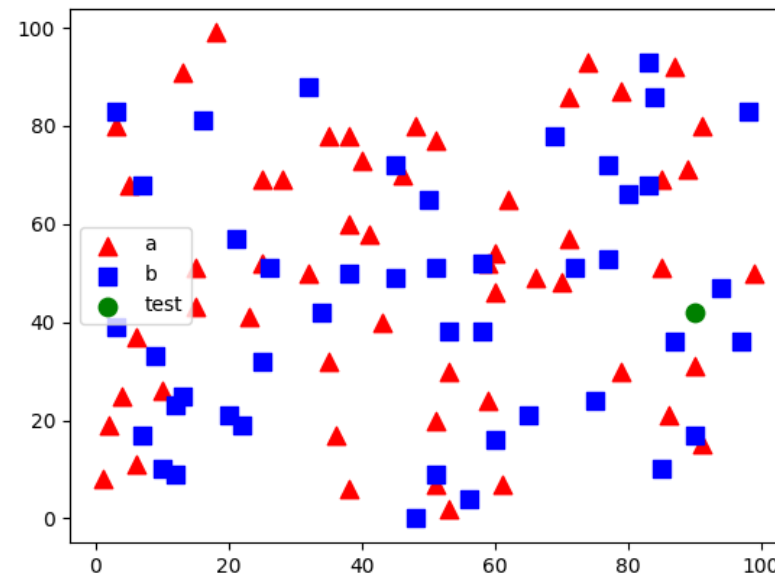
# - 가상 데이터 만들기



## - Numpy를 활용한 가상데이터 만들기

- ① 10\* 2 형태의 랜덤 트레이닝 데이터 생성
- ② Label(1 or 2)에 해당하는 10\*1 형태 랜덤 트레이닝 데이터 레이블 생성
- ③ KNN - Classifier을 수행 할 데이터 만들기

```
data_train = np.random.randint(0, 100, (10, 2))  
data_label = np.random.randint(0, 2, (10, 1))  
data_test = np.random.randint(0, 100, (1, 2))
```



# - KNN Classifier 함수



예시 ) 10개의 Training 데이터와 1개의 Test 데이터

|         |      |               |      |               |      |
|---------|------|---------------|------|---------------|------|
| [[1 0 ] | [[0] | [[8.54400375] | [[0] | [[1.]         | [[1] |
| [9 8 ]  | [1]  | [5.]          | [1]  | [2.23606798]  | [0]  |
| [8 7 ]  | [0]  | [4.12310563]  | [0]  | [2.23606798]  | [1]  |
| [4 7 ]  | [1]  | [1.]          | [1]  | [4.]          | [1]  |
| [2 7 ]  | [0]  | [2.23606798]  | [0]  | [4.12310563]  | [0]  |
| [9 1 ]  | [1]  | [8.60232527]  | [1]  | [4.12310563]  | [0]  |
| [0 8 ]  | [1]  | [4.]          | [1]  | [5.]          | [1]  |
| [3 6 ]  | [1]  | [2.23606798]  | [1]  | [7.07106781]  | [0]  |
| [0 9 ]  | [0]  | [4.12310563]  | [0]  | [8.54400375]  | [0]  |
| [3 1 ]] | [0]] | [7.07106781]] | [0]] | [8.60232527]] | [1]] |

-> Training Data

[[4 8]]

-> Test Data

-> Test데이터와  
Training 데이터들과의 거리

-> 데이터 정렬

1) k = 1 test 데이터 : b

2) K = 2 test 데이터 : ?

3) k = 3 test 데이터 : b

4) K = 5 test 데이터 : b

# K-Cross Validation은 무엇인가?

- K-Cross Validation
- **Cross-validation**, sometimes called **rotation estimation** or **out-of-sample testing**, is any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set.
- [https://en.wikipedia.org/wiki/Cross-validation\\_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))

