1. 머신러닝

(1) 개념

애플리케이션을 수정하지 않고도 데이터를 기반으로 패턴을 학습하고 결과를 예측하는 알고리즘 기법을 통칭

(2) 분류

(ㄱ) 지도학습

-분류/회귀

(ㄴ)비지도

-클러스터링/차원 축소/강화학습

2. 파이썬 머신러닝 생태계를 구성하는 주요 패키지

(1) 머신러닝 패키지

-사이킷런(데이터 마이닝 기반의 머신러닝에서 독보적인 위치를 차지하고 있음)

-케라스

-텐설플로우

(2) 행렬/선형대수/통계 패키지

-넘파이: 선형대수를 다룸

-사이파이: 자연과학과 통계를 위한 다양한 패키지

(3) 데이터 핸들링

-판다스

(4) 시각화

-맷플롯립

-시본: 맷플롯립을 보완하는 패키지

(5) 아이파이썬

: 대화형 파이썬 툴(프로그래밍과 이에 대한 설명적인 요소를 결합했다는 뜻)

-주피터 노트북(중요 코드 단위로 설명을 적고 코드를 수해해 그 결과를 볼 수 있게 만들어서 직관적으로 어떤 코드가 어떤 역할을 하는지 매우 쉽게 이해할 수 있도록 지원)

3. 파이썬 머신러닝을 위한 S/W 설치

(1) 아나콘다 설치

\*스터디 시간에 같이 설치하기

(2) Microsoft Visual Studio Build Tools 설치

+넘파이/판다스: 대부분의 데이터 처리 부분

+사이킷런의 API 코드: 넘파이를 기반으로 작성됨

4. 넘파이(Numerical Python)

: 파이썬에서 선형대수 기반의 프로그램을 쉽게 만들 수 있도록 지원하는 대표적인 패키지

(1) ndarray

: 다차원의 배열을 쉽게 형성하고 다양한 연상을 수행할 수 있습니다.

#Array1=np.array([1,2,3])

#format(array1.ndim)

(2) ndarray의 데이터 타입

-int형(8/16/32)

-unsigned int형

-float형

#array\_float=array\_int.astype(‘float64’)

🡪데이터 타입 변형

(3) ndarray 편리하게 생성하기

(ㄱ) arange

#sequence\_array=np.arange(10)

(ㄴ) zeros

#zero\_array=np.zeros((3,2), dtype=’int32’)

(ㄷ) ones

#one\_array=np.ones((3,2), dtype=’int32’)

(4) 차원과 크기를 변경하는 reshape

#array3=array1.reshape(5,2)

#array3=array1.reshape(5,-1) 🡪똑같은 결과값 나옴

#array3=array1.reshape(1,-1)🡪2차원이고 여러 개의 로우를 가지지만 1개의 칼럼을 가진ndarray로 변형

(5) 넘파이의 ndarray의 데이터 세트 선택-indexing

(1) 단일 값 추출

(ㄱ) 단일 값 추출

#value=array1[2]

#value=array1[-1] : 맨 뒤의 값

(ㄴ) 인덱스 수정

#array1[2]:5

(ㄷ) 다차원에서의 단일 값 추출

#value=array1[0,3]

\*indexing에서는 row 와 column의 개념이 아니라 axis0,1,2 이런 개념으로 사용되는 것이 맞음

(2) 슬라이싱

(ㄱ)

#array3=array1[0:3]

(ㄴ) 다차원의 슬라이싱

#array4=array2d[0:2, 1:3]

(3) 펜시 인덱싱

#array4=array2d[[0,1], 0:2]

#array4.tolist()

🡪

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

(4) 불린 인덱싱

: 필터링과 검색을 동시에 할 수 있음

#array3=array1d[array1d>5] : 5보다 큰 값만 출력하라는 뜻

#array 1>5 🡪array([F,F,F,F,F,T,T,T,T]) 이렇게 나온다

5. 행렬의 정렬

(ㄱ) 1차원

#sort\_array1=np.sort(org\_array): 원본 행렬이 유지됨

#sort\_array2=org\_array.sort() : 원본 행렬이 바뀌는 것

\*오름차순이 기본임

\*내림차순으로 하기 위해서는

#sort\_array3=np.sort(org\_array)[::-1]

(ㄴ) 2차원

#sort\_array2d\_axis0=np.sort(array2d, axis=0) : 로우 방향으로 정렬

# sort\_array2d\_axis0=np.sort(array2d, axis=1) : 칼럼 방향으로 정렬

(ㄷ) 정렬된 행렬의 인덱스를 반환하기

#sort\_indices=np.argsort(org\_array) : 행렬 정렬 시 원본 행렬의 인덱스

#sort\_indices=np.argsort(org\_array)[::-1] : 내림 차원의 경우

Ex) 데이터 활용 방안

학생들의 성적을 내림차순으로 정렬하고 name index를 나오게 할 수 있음

6. 선형대수 연산

#dot\_product=np.dot(A,B)

#transpose\_mat=np.transpose(A)