● Numpy (numerical python)

1. Numpy

: 넘파이(numpy)는 과학 계산을 위한 파이썬 데이터 분석 패키지이다. 다차원 배열을 처리하는데 필요한 여러 기능을 제공한다.

1. 다차원 배열(ndarray)

: 다차원배열이란 같은 자료형의 데이터를 담을 수 있는 포괄적인 그릇이라고 할 수 있다.

Ndarray의 모든 원소는 같은 자료형만 사용 가능하며 배열의 차원을 **rank**라 하고, 각 차원의 크기를 튜플로 표시하는 것을 **shape** 라고 한다.

1. Numpy 주요 함수

: 넘파이는 자료를 다루는 함수이기 때문에 다양한 연산 기능을 제공한다.

Numpy 뒤에 ‘.연산함수’를 붙이면 된다.

* Numpy 주요 연산 함수

|  |  |  |
| --- | --- | --- |
| 함수 | 설명 | 사용법 |
| abs, fabs | 절대값을 리턴, 복소수가 아닌 경우에는 빠른 연산을 위해 fabs 이용 | numpy.abs(arr) |
| sqrt | 제곱근(루트)을 계산 | numpy.sqrt(arr) |
| square | 제곱 계산 | numpy.square(arr) |
| exp | 지수 계산 | numpy.exp(arr) |
| Log | 로그 계산 | numpy.Log(arr) |
| add | 두 배열을 더한다 | numpy.add(arr1,arr2) |
| subtract | 첫 번째 배열에서 두 번째 배열을 뺀다 | numpy.subtract(arr1,arr2) |
| multiply | 두 배열을 곱한다 | numpy.multiply(arr1,arr2) |

* 명령어 : np.reshape(차원, 행, 렬, order = ‘C’ or ‘F’)
* Order=’C’ : 값을 행부터 채워 넣음 (기본값, 생략 가능), order = ’F’ : 값을 열부터 채워 넣음

1. Pandas

: 판다스(Pandas)는 자료구조 및 데이터 분석,처리를 위한 파이썬의 핵심 패키지이다.

1. Series

: 시리즈(Series)는 index와 value의 형태를 갖고 있는 pandas의 자료구조이다.

시리즈는 value외에 index도 갖기 때문에 value만 갖는 리스트와 다르다.

Index는 기본값으로 0,1,2,3,,으로 자동 생성된다.

1. DataFrame

* CSV파일 불러오기 : df = pd.read\_csv(‘파일명.csv’)
* CSV파일 불러오기(인코딩 옵션) : df = pd.read\_csv(‘파일명.csv’,encoding=’euc-kr’)
* Xls/xlsx 파일 불러오기 : pd.read\_excel(‘파일명.xlxs’)
* Xls/xlsx 파일 불러오기(인코딩 옵션) : pd.read\_excel(‘파일명.xlxs’,encoding=’euc-kr’)
* 처음부터 ~개의 행만 불러오기 : 변수명.head(불러올 행 개수, 공백시 5개)
* 뒤부터 ~개의 행만 불러오기 : 변수명.tail(불러올 행 개수, 공백시 5개)
* 특정한 연속적 행을 불러오기 : 변수명[처음 출력할 행 : 마지막으로 출력할 행 +1]
* 열 불러오는 방법 (1) : 데이터셋명[[‘열 이름’]]
* 열 불러오는 방법 (2) : 데이터셋명[데이터셋명.columns[[열 번호]]]
* 열 불러오는 방법 (3) : 데이터셋명.loc[:, 첫 열 이름 : 끝 열 이름]
* 만약 ‘[[ ]]’가 아니라 ‘[ ]’이면 데이터프레임이 아닌 시리즈형태로 불러와진다.
* 행과 열 불러오는 방법 : 데이터셋명.iloc[첫 행 : 끝 행, 첫 열 : 끝 열]
* 특정한 하나의 값을 불러오는 방법 : 데이터셋명.at[행 번호, ‘열 이름’]
* 데이터 프레임 전체를 복사하여 백업할 경우 : df\_columns = df.copy()
* 데이터 프레임의 열 이름을 확인할 경우 : df.columns
* 열들 중에서 일부 열(변수)만 필터링하여 데이터 셋을 구성할 경우

: df = df[[‘변수명’,’변수명’,’변수명’]]

* 변수이름 변경(period를 time으로 변경 예시)

: df.rename(columns={‘period’ : ‘time’}, inplace=True)

* 범주와 수 확인 : df[‘변수’].value\_counts()
* 함수 적용 방법

1. df.replace(‘함수명’)
2. df.apply(‘함수명’)
3. pandas와 numpy 사용 방법

: 머신러닝이나 딥러닝에서는 넘파이로 분석을 해야 빠른 연산이 가능하기 때문에 판다스로 기본적 분석을 한 후 넘파이로 변환하여 머신러닝, 딥러닝 분석을 수행한다. 반대로 넘파이 데이터는 눈으로 확인이 어려워 판다스로 변환하여 데이터를 살펴본다.