| 교육 제목 | 데이터 기반 인공지능 시스템 엔지니어 양성 과정\_판다스 |
| --- | --- |
| 교육 일시 | 2021년 9월 29일 |
| 교육 장소 | YGL C-6 학과장 & 자택(디스코드 이용한 온라인) |
| **교육 내용** | |
| 오전 | 파이썬 머신러닝 판다스 데이터 분석 1장 \_ 판다스 입문   * 데이터 과학자가 판다스를 배우는 이유 * 판다스 자료 구조   + 판다스의 구조화된 데이터 형식 - 시리즈(series) : 1차원 배열, 인덱스(index)와 데이터값(value)이 일대일 대응 ! 딕셔너리 → 시리즈 변환 : pandas.Series(딕셔너리) ! 인덱스 배열 : Series객체.index ! 데이터 값 배열 : Series객체.values - 데이터프레임(dataframe) : 2차원 배열, 열(column)은 공통의 속성을 갖는 일련의 데이터를 나타내고 행(row)은 개별 관측대상에 대한 다양한 속성 데이터들의 모음인 레코드(record)가 된다.  ! 딕셔너리 → 데이터프레임 변환 : pandas.DataFrame(딕셔너리 객체) ! 행 인덱스/열 이름 설정 : pandas.DataFrame(2차원 배열, index=행 인덱스 배열, columns=열 이름 배열) !! 행 인덱스 변경 : DataFrame 객체.index = 새로운 행 인덱스 배열 !! 열 이름 변경 : DataFrame 객체.columns = 새로운 열 이름 배열  ! rename() 메소드 적용법 : 행 인덱스 또는 열 이름 일부를 선택하여 변경할 수 있다. 새로운 데이터프레임 객체를 반환함으로 원본 객체를 변경하기 위해서는 inplace=True 옵션을 사용한다.  - 행 인덱스 변경: DataFrame 객체.rename(index={기존 인덱스:새 인덱스, … })  - 열 이름 변경: DataFrame 객체.rename(columns={기존 이름:새 이름, … })  ! drop() 메소드 적용법 : 데이터 프레임의 행 또는 열을 삭제하는 명령. 새로운 데이터프레임 객체를 반환함으로 원본 객체를 변경하기 위해서는 inplace=True 옵션을 사용한다.  - 행 삭제: DataFrame 객체.drop(행 인덱스 또는 배열, axis=0, inplace=True) - 열 삭제: DataFrame 객체.drop(열 이름 또는 배열, axis=1, inplace=True)  ! 행 선택 : loc[‘인덱스 이름’] vs iloc[정수형 위치 인덱스] : 데이터프레임의 행 데이터를 선택하기 위해 사용되는 인덱서  c.f> 범위 슬라이싱 : DataFrame 객체.iloc[시작 인덱스 : 끝 인덱스 : 슬라이싱 간격]  ! 열 1개 선택 : DataFrame 객체[‘열 이름'] or DataFrame 객체.열 이름 ! 열 n개 선택 : DataFrame 객체[[열1, 열2, … , 열n]] → 데이터프레임 생성됨   ! 원소 선택 : 2개 이상 선택할 때는 리스트 형식으로 입력하거나 슬라이싱을 이용한다.  - 이름 인덱스 : DataFrame 객체.loc[행 인덱스, 열 이름] - 정수 위치 인덱스 : DataFrame 객체.iloc[행 번호, 열 번호]  ! 열 추가 : DataFrame 객체[‘추가하려는 열 이름'] = 데이터 값 ! 행 추가 : DataFrame 객체.loc[‘새로운 행 인덱스'] = 데이터 값 (또는 배열)  ! 원소 값 변경 : DataFrame 객체의 일부분 또는 원소를 선택 = 새로운 값   + 행, 열 위치 바꾸기 : DataFrame 객체.transpose() 또는 DataFrame 객체.T   + 인덱스의 활용  - 특정 열을 행 인덱스로 설정 : DataFrame 객체.set\_index([‘열이름'] 또는 ‘열이름) ! 기존의 행 인덱스는 삭제  - 행 인덱스 재배열 : DataFrame 객체.reindex[새로운 인덱스 배열] - 행 인덱스 초기화 : DataFrame 객체.reset\_index() ! 기존의 인덱스는 열로 이동 - 행 인덱스를 기준으로 데이터프레임 정렬 : DataFrame 객체.sort\_index() ! ascending=True or False 옵션으로 정렬방식 지정 ~! 특정 열의 데이터를 기준으로 정렬 : DataFrame 객체.sort\_values()   + 산술연산 : 3단계 (행/열 인덱스를 기준으로 모든 원소 정렬 → 동일한 위치에 있는 원소끼리 일대일 대응 → 일대일 대응이 되는 원소끼리 연산을 처리) - 시리즈 vs 숫자 (또는 시리즈) ! 한쪽에만 인덱스가 존재하고 짝이 없는 경우 유효한 값이 존재하지 않는다는 의미로 NaN 처리 - 연산 메소드 : NaN 발생을 피하기 위해 fill\_value 옵션을 설정 ! ex) Series1.add(Series2, fill\_value=()) - 데이터프레임 연산 : 시리즈와 동일 |
| 오후 | 파이썬 머신러닝 판다스 데이터 분석 2장 \_ 데이터 입출력   | df = pandas.DataFrame ..   * 행 df.loc[인덱스명], df.iloc[인덱스첨자] 여러개 start:end, [ , ... , ] * 열 df.열명 df[열명] * 원소 df.loc[[인덱스명],[컬럼명]] df.iloc[[인덱스첨자],[컬럼의 순서 정수]] | | --- |  1. 외부 파일 읽어오기 2. 웹(web)에서 가져오기 3. API 활용하여 데이터 수집하기 4. 데이터 저장하기  | * 읽기 : read\_csv , read\_html, read\_excel, read\_json * 저장 : df.to\_csv(파일명), df.to\_html, df.to\_excel, df.to\_json * json 파일은 구조가 DataFrame과 동일 csv 파일은 2차원 배열로 --> DataFrame으로 변경해서 사용 | | --- | |