|  |  |
| --- | --- |
| 교육 제목 | **Pandas 데이터프레임 응용- 병합과 정렬 머신러닝의 기초** |
| 교육 일시 | 21년 10월 5일 |
| 교육 장소 | YGL - 비대면 수업 (자택) |
| **교육 내용** | |
| 오전 | 시계열 데이터(time series)  -객체.to\_datetime() 함수: 문자열이나 다른 자료형의 배열을을 판다스 Timestamp를 나타내는 datetime64 자료형으로 변환한다  \*객체.to\_period() 함수: Timestamp를 Periodf객체로 변환해 준다  \*연월일을 나타내는 데이터를 문자열로 변경 ex)day 5pandas 참고  df['연월일'] = df['연월일'].astype('str')  \*연월일을 '-'기준으로 split  dates = df['연월일'].str.split('-')  \* 3.데이터프레임에 '연', '월', '일' 컬럼을 추가  \*리스트에서 각 리스트의 동일 위치의 원소를 추출하고자 할 경우  리스트.str.get(위치인덱스)  df['연'] = dates.str.get(0) #dates의 원소 리스트의 0번째 인덱스 값을 가져온다  \*원래는 df['열'] = dates[0][0] ...dates[0][1]이런식으로 해야함  \*하지만 시리즈의 경우 str.get() 함수가 있어서 편히 할 수 있음  df['월'] = dates.str.get(1)  df['일'] = dates.str.get(2)  이런 식으로 데이터 프레임에 시간을 직관성 있게 추가 할 수 있다.  열 재구성  -객체.str.get(위치인덱스) : 객체 안의 리스트들의 동일 위치의 원소를 추출 할 수 있다  df['월'] = dates.str.get(1)  df['일'] = dates.str.get(2)  위와 같은 경우에는 df객체 안의 '월'과'일' 행에 해당되는 원소들을 추출한다 get()메서드 안의 정수들은 인덱스로 예제에는 '년월일'을 추출해서 월은 1번 인덱스에 일은 2번 인덱스에 있어서 그런 것이다.  필터링  데이터프레임 객체에서 원하는 데이터 만을 추출을 하기 위해서는 mask기법을 이용하여 연산이나 구분자를 통해 간단한 함수를 만들어서 데이터를 뽑았지만  mask2 = (titanic.sibsp == 3) | (titanic.sibsp == 4) | (titanic.sibsp == 5) 이것은 seaborn의 titanic 데이터세트로 함께 탑승한 형제 또는 배우자 수가 3,4,5인 승객만 따로 추출하는 mask기법이다  -객체['행'].isin(원소) 함수:  mask3 = titanic['sibsp'].isin([3,4,5]) 과 같이 titanic데이터 셋의 'sibsp'행의 3,4,5원소를 가진 데이터 만을 mask3에 간단히 넣어준다  데이터프레임 합치기  -객체.concat([데이터프레임1,데이터프레임2]) 함수: 데이터프레임들을 합쳐주는데 누락된 데이터들은 NaN으로 나타난다. 기본적으로 행을 기준으로 붙이게 되고 열로 붙이고 싶으면 객체.concat([데이터프레임1,데이터프레임2],axis=1)로 축을 1로 지정해서 열을 기준으로 데이터 프레임을 합치게 해준다.  \*join키워드: concat()메서드에서의 옵션으로 inner outer left right가 있고 형식은 객체.concat([데이터프레임1,데이터프레임2],axis=1,join='inner') 와 같은것이 있으며 inner는 데이터 프레임들의 교집합 부분만을 보여주며 기본으로 지정된 outer는 합집합에 해당되는 모든 부분을 보여준다.  -객체.merge() :데이터프레임을 합쳐주는 메서드로 concat()메서드와 비슷한 방식으로 어떤 기준에 의해 두 데이터프레임을 병합해준다. 하지만 이때 기준이 되는 열이나 인덱스인 키(key)를 두 데이터프레임 모두 가지고 있어야한다. |
| 오후 | \*how 키워드: concat()메서드의 join옵션과 같이 어떻게 병합을 하는지를 지정하는 옵션이고 두 데이터프레임이 왼쪽, 오른쪽에 있느냐에 따라 left\_on=, right\_on= 옵션도 쓸 수 있는데 이때 두 데이터프레임의 key값을 지정하면 그 키값을 기준으로 병합한다. how='left', left\_on='키' 와 같은 경우에는 왼쪽 데이터 프레임의 '키'를 기준으로 병합을 하는 것이다.  -객체.join() :이 메서드는 merge()메서드와 비슷한 기능을 하지만 join()메서드는 행 인덱스를 기준으로 병합을 한다.  그룹 연산  -객체.groupby(['기준이 되는 열']) :groupby()메서드는 데이터프레임의 특정 열을 기준으로 데이터프레임을 분할하여 그룹 객체를 반환해 주는 메서드이고 기준이 되는 열은 1개부터 여러개의 열을 리스트로 지정할 수도 있다.  \*객체.get\_group('열의 인덱스(ex-선실등급class-열 : First Secnod Third)') : 이 메서드는 인덱스를 이용하여 특정 그룹만을 추출하는 메서드이고 여러 개를 지정 할 수도 있다. 예제에서는 선실등급의 Third와 승객의 성별의 female을 지정하여 3등급 선실을 배정 받은 여성 손님의 데이터만을 추출해서 반환했다.  \*객체.agg(함수):std표준편차 min최소값 max최대값 mean평균값과 같은 집계 연산을 처리하는 사용자 정의 함수를 그룹 객체에 적용할 때 활용 할 수 있는 메서드이고 사용 방식은 그룹객체.agg(매핑함수)와 같이 사용하고 당연히 함수는 사전에 정의 되어있어야 한다. 괄호안에 함수는 여러개를 지정해도 되고 각 열마다 다른 함수를 지정할 때에는 딕셔너리형태로 객체.agg({'열1':함수1, '열2':함수2...})와 같이 활용해야 한다.  \*객체.transform(매핑함수): 이 메서드는 그룹별로 구분하여 각 원소에 함수를 적용하지만 그룹별 집계 대신 각 원소의 본래 행 인덱스와 열 이름을 기준으로 연산 결과를 반환 해준다.  -객체.filter(조건함수): 이 메서드는 조건이 참인 그룹만을 반환한다.  \*예제-데이터 개수가 200개 이상인 그룹만을 필터링하여 데이터프레임으로 반환  grouped.filter(lambda x:len(x) >= 200) #람다를 사용하여 간단히 grouped 그룹 객체의 조건을 설정한 경우이다.  멀티 인덱스  객체.xs('행'): xs인덱서는 특정 인덱스에서 특정 값을 갖는 행을 추출할때 활용 할 수 있다. ex)  gdf-그룹객체.xs('male', level='sex') 와 같은 경우는 'sex'인덱스에서 'male'값을 갖는 행을 추출한 것이다.  sql 문장  select 보여주는 리스트  from 테이블 또는 뷰(데이터프레임)  where (join, merge 조건)  group by (원하는 컬럼또는 컬럼의 조합으로 그룹)  having 그룹에 함수 적용을 해서 조건  order by 정렬  group = df.groupby(컬럼 또는 컬럼리스트)  grouped.함수() - min, max, mean, std etc...  grouped.get\_group(원하는 그룹의 컬럼 리스트)  grouped.agg(함수명/[함수 리스트])  grouped.agg({'컬럼명':[함수리스트],컬럼명:함수})  #데이터 개수가 200개 이상인 그룹만을 필터링  #하여 데이터프레임으로 반환  grouped.filter(lambda x:len(x) >= 200)  #age열의 평균이 30보다 작은 그룹만을  #필터링하여 데이터프레임으로 반환  grouped.filter(lambda x:x.age.mean() <30)  grouped.apply(lambda x:x.age.mean() <30)  데이터 분석이란? : 예측을 하기 위함  -자료를 가져옴  -누락데이터 처리 및 데이터 전처리  \*데이터 프레임 조인,분리,결합  \*데이터 가공  \*문자를 더미 one\_hot\_encoing etc...  \*분류화를 하거나...  \*NaN 데이터 처리  \*x와 y값 분리  -실험 데이터와 테스트 데이터로 분리  -모델 학습  -모델 학습 결과로 타겟값을 예측/계산  -테스트 데이터를 모델값에 적용 한 결과와 실제 테스트의 타겟값과 비교 후 점수를 확인  -예측된 값과 실제 값을 비교해서 그래프화 한다  -모델 검증.... |