|  |  |
| --- | --- |
| 교육 제목 | 판다스 데이터 전처리 |
| 교육 일시 | 10월 1일(금) |
| 교육 장소 | 강의실 |
| **교육 내용** | |
|  | * 주어진 데이터 내에서 NaN 값이 있는지 탐색 한 후 있다면 제거하거나 다른 값으로 대치 해주어야 한다. 너무 많은 NaN 값이나 NaN이 있는 행은 데이터 분석에서 결과에 영향을 주거나 왜곡 시킬수 있기 때문에 데이터 분석 실행전 NaN 값들에 대하여 적절한 처리를 해야 한다   .isnull() => null 값이 있다면 True 아니면 False  .notnull() = > null값이 있다면 False 아니면 값 리턴  .isnull().sum() => 결측치의 개수를 파악할수 있다.   * 누락데이터의 처리는 크게  1. 행을 제거하는 방법, 2. 평균으로 넣는법, 3. 앞의값으로 넣는법, 4. 최대빈도 값으로 넣는법 등의 처리 방법이 있다.   평균값 대치  Mean\_age = df[‘age’].mean(axis=0)  df\_age = df[‘age’].fillna(mean\_age)  앞의 값으로 수정  df[‘age’].fillna(method=’ffill’)  최대 빈도값으로 대치  Most\_cnt = df[‘age’].value\_counts(dropna=True).idxmax()  df\_age\_2 = df[‘age’].fillna(most\_cnt)   * 행의 중복을 체크   df\_dup = df.duplicated() => 행의 중복을 체크   * 열의 중복을 체크   Df[‘c1’].duplicated()   * 범주형 데이터 처리   범주형 데이터들은 컴퓨터가 바로 계산할 수 없으므로 one\_hotencoding등으로 대체 해야 한다.   * 범위로 나누기   Bin\_names = [‘유아기’, ‘청소년’, ‘청년’, ‘장년’]  Bin\_values = [0, 20, 40, 60, 100]  Titanic[‘age\_bin’] = pd.cut( x = titanic[‘age’],  Bins = bin\_values,  Labels = bin\_names,  Include\_lowest = True)   * Sklearn을 활용한 원핫 인코딩   Label\_encoder = preprocessing.LabelEncoder()  Onehot\_encoder = preprocessing.OneHotEncoder() |
| 전체 내용 |