**교육일지**

**교육 제목 : 데이터 사전 처리**

**교육 장소 : YGL C6 강의실**

**교육 일시 : 2021/10/01**

**오전 교육**

**isnull() : NaN이면 True, notnull: 값이 존재하면 데이터의 값,NaN 반환**

**# 누락된 데이터 처리**

**#제거, 치환**

**print(df.head().isnull())**

**print(df['deck'].head().isnull())**

**print("------")**

**print(df.head().notnull) # 값이 존재하면 값을, 없으면 NaN을 출력**

**print("-------")**

**print(df.head().isnull().sum(axis=0)) #누락된 데이터가 있는지 확인**

**# 누락된 자료가 500개 이상인 자료를 제거**

**df\_1 = df.dropna(axis=1, thresh=500)**

**"deck" in (df\_1.columns)**

**# 누락 데이터가 많은 컬럼을 제거하는 방법**

**df\_1 = df.dropna(axis=1,thresh=500)**

**# age 열에 NaN 값이 있으면 행을 제거하는 방법**

**df\_2 = df.dropna(subset=['age'], how='any', axis=0)**

**# age 열의 NaN값을 다른 나이 데이터의 평균으로 변경하기**

**mean\_age = df['age'].mean(axis=0)**

**df\_age = df['age'].fillna(mean\_age)**

**#앞의 값으로 수정**

**df\_age\_1 = df['age'].fillna(method='ffill')**

**df\_age\_1**

**#가장비번하게 나오는 값으로 수정**

**most\_cnt = df['age'].value\_count(dropna=True).idmax()**

**df\_age\_2 =df['age'].fillna(most\_cnt)**

**중복데이터 처리**

**#중복 데이터 처리**

**#중복 데이터를 갖는 데이터 프레임 만들기**

**df = pd.DataFrame({'c1':['a', 'a', 'b', 'a', 'b'],**

**'c2':[1, 1, 1, 2, 2 ],**

**'c3':[1, 1, 2, 2, 2]})**

**df\_dup = df.duplicated() # 행의 중복을 체크**

**print(df\_dup)**

**#특정 열의 중복 체크**

**df['c1'].duplicated()**

**#데이터 프레임에서 중복 행을 제거**

**df2 = df.drop\_duplicates()**

**print(df2)**

**#특정열을 기준으로 중복 행을 제거**

**df2 = df.drop\_duplicates(['c2','c3'])**

**데이터 자료형 변환**

**#자료형, 단위를 변환**

**#mpg(mile per gallon)를 kp(kilometer per liter)로 변환 (mpg\_to\_kpl = 0.425)**

**mpg\_to\_kpi = 1.60934 / 3.78541**

**#mpg 열에 0.425를 곱한 결과를 새로운 열(kpl)에 추가**

**df['kpl']= (df['mpg']\*mpg\_to\_kpl).round(2)**

**df.kpl.head(3)**

**# 1. 데이터 타입 확인 , unique(), '?'-> NaN으로 처리**

**df\_hp = df.copy()**

**df\_hp.horsepower.replace('?',np.nan, inplace=True) # '?'를 np.nan로 변경**

**# 2. NaN데이터 확인 후 -> 처리 -> 0.0값으로 치환**

**#df\_hp['horsepower'].fillna('0.0',inplace=True)**

**df\_hp['horsepower'].fillna('0.0',inplace=True) #누락데이터 행 삭제**

**# 3. 데이터 타입 변경 -> FLOAT으로 변경**

**df\_hp['horsepower'] = df\_hp['horsepower'],astype('float') #문자형을 실수형으로**

**#category, 데이터 타입을범주형 또는 문자형, 숫자형으로 원하는 형태로 변경**