02 데이터 전처리

학습 목표

- 데이터 오류에 대해 확인하고, 데이터 전처리를 수행한다.
- 모델을 제출해 본다.

- 대회 데이터 셋 오류로 인한 데이터 전처리
  - https://dacon.io/competitions/official/235745/talkboard/403708?page=1&dtype=recent (https://dacon.io/competitions/official/235745/talkboard/403708?page=1&dtype=recent)

In [1]:

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

In [2]:

```
import pandas as pd

train = pd.read_csv("../data/parking_demand/train.csv")
test = pd.read_csv("../data/parking_demand/test.csv")
sub = pd.read_csv("../data/parking_demand/sample_submission.csv")
age = pd.read_csv("../data/parking_demand/age_gender_info.csv")

train.shape, test.shape, sub.shape, age.shape
```

Out[2]:

```
((2952, 15), (1022, 14), (150, 2), (16, 23))
```

In [3]:

```
train.columns
```

Out[3]:

```
Index(['단지코드', '총세대수', '임대건물구분', '지역', '공급유형', '전용면적', '전용
면적별세대수', '공가수',
       '자격유형', '임대보증금', '임대료', '도보 10분거리 내 지하철역 수(환승노선 수
반영)',
       '도보 10분거리 내 버스정류장 수', '단지내주차면수', '등록차량수'],
      dtype='object')
```

In [4]:

```
train.columns = ['단지코드', '총세대수', '임대건물구분', '지역', '공급유형', '전용면적', '전용면적별
       '자격유형', '임대보증금', '임대료', '10분내지하철수',
       '10분내버스정류장수', '단지내주차면수', '등록차량수']

test.columns = ['단지코드', '총세대수', '임대건물구분', '지역', '공급유형', '전용면적', '전용면적별
       '자격유형', '임대보증금', '임대료', '10분내지하철수',
       '10분내버스정류장수', '단지내주차면수']
```

데이터 오류로 인한 데이터 제외

- - 테스트셋에서 평가 제외되는 데이터는 'C2675'(2번 사항에 해당), 'C2335', 'C1327'(3번 사항에 해당) 3개 단지입니다.

In [5]:

```
train.단지코드.unique()
```

Out[5]:

```
array(['C2483', 'C2515', 'C1407', 'C1945', 'C1470', 'C1898', 'C1244',
       'C1171', 'C2073', 'C2513', 'C1936', 'C2049', 'C2202', 'C1925',
       'C2576', 'C1312', 'C1874', 'C2650', 'C2416', 'C2013', 'C1424',
       'C2100', 'C2621', 'C2520', 'C2319', 'C1616', 'C1704', 'C2258',
       'C1032', 'C2038', 'C1859', 'C1722', 'C1850', 'C2190', 'C1476',
       'C1077', 'C1068', 'C1983', 'C2135', 'C2034', 'C1109', 'C1497',
       'C2289', 'C2597', 'C2310', 'C1672', 'C2132', 'C1439', 'C1613',
       'C2216', 'C1899', 'C1056', 'C2644', 'C1206', 'C2481', 'C1718',
       'C1655', 'C1430', 'C1775', 'C1519', 'C2221', 'C1790', 'C2109',
       'C1698', 'C1866', 'C1005', 'C1004', 'C1875', 'C2156', 'C2212',
       'C2401', 'C2571', 'C1175', 'C1833', 'C2445', 'C1885', 'C2368',
       'C2016', 'C2371', 'C2536', 'C2538', 'C1014', 'C1592', 'C1867',
       'C2326', 'C1015', 'C1620', 'C1049', 'C2000', 'C2097', 'C1668',
       'C1689', 'C1234', 'C2514', 'C1368', 'C1057', 'C2336', 'C1026',
       'C2256', 'C1900', 'C2666', 'C2361', 'C1642', 'C1013', 'C2232',
       'C1973', 'C2458', 'C2574', 'C2133', 'C2096', 'C2010', 'C1879',
       'C1131', 'C1468', 'C1213', 'C1173', 'C2492', 'C2032', 'C2094',
       'C1880', 'C2089', 'C1744', 'C2046', 'C2071', 'C2635', 'C2390',
       'C2561', 'C1663', 'C2490', 'C2066', 'C1585', 'C2276', 'C1155',
       'C1693', 'C1889', 'C2518', 'C1962', 'C1666', 'C1988', 'C1537',
       'C1329', 'C1762', 'C2008', 'C1319', 'C1141', 'C2340', 'C1929',
       'C1681', 'C1184', 'C2383', 'C1579', 'C2173', 'C1911', 'C1638',
       'C2412', 'C1871', 'C1309', 'C1527', 'C2208', 'C1940', 'C2596',
       'C2227', 'C2563', 'C2358', 'C1492', 'C1601', 'C1687', 'C1236',
       'C1487', 'C1379', 'C1386', 'C1656', 'C2526', 'C1022', 'C1896',
       'C1269', 'C1916', 'C2070', 'C1967', 'C2021', 'C1143', 'C2188',
       'C2651', 'C1036', 'C2657', 'C2527', 'C1502', 'C2262', 'C1084',
       'C2530', 'C1046', 'C1761', 'C1102', 'C2420', 'C1122', 'C2042',
       'C1375', 'C1410', 'C1641', 'C1706', 'C1307', 'C2601', 'C1085',
       'C2385', 'C1059', 'C2162', 'C1819', 'C2325', 'C2394', 'C1133',
       'C1281', 'C1194', 'C2308', 'C2036', 'C1394', 'C1180', 'C2503',
       'C1907', 'C2181', 'C1768', 'C1783', 'C2192', 'C2346', 'C2680',
       'C2631', 'C2141', 'C1569', 'C2099', 'C2287', 'C2055', 'C1428',
       'C2522', 'C2560', 'C2068', 'C2603', 'C1965', 'C1660', 'C2378',
       'C1268', 'C1994', 'C1837', 'C1000', 'C1465', 'C1448', 'C1516',
       'C2670', 'C1365', 'C1177', 'C1360', 'C2488', 'C1406', 'C1566',
       'C1227', 'C2460', 'C2486', 'C2106', 'C1572', 'C1773', 'C1677',
       'C1823', 'C1344', 'C2692', 'C2505', 'C2587', 'C2127', 'C1316',
       'C1674', 'C1713', 'C1845', 'C2082', 'C1328', 'C2357', 'C2565',
       'C1804', 'C1397', 'C2255', 'C1343', 'C1987', 'C2479', 'C2352',
       'C1310', 'C1738', 'C1039', 'C1863', 'C1426', 'C2659', 'C2489',
       'C2211', 'C2314', 'C1861', 'C2389', 'C1490', 'C1024', 'C1788',
       'C1740', 'C2620', 'C1286', 'C2085', 'C1089', 'C2237', 'C1341',
       'C1338', 'C2405', 'C1969', 'C2274', 'C1699', 'C2251', 'C1340',
       'C2373', 'C1455', 'C1095', 'C2137', 'C1985', 'C2583', 'C2663',
       'C2450', 'C2329', 'C1834', 'C1649', 'C1848', 'C1743', 'C1350',
       'C1402', 'C1103', 'C1129', 'C1027', 'C2377', 'C2431', 'C2661',
       'C1263', 'C1136', 'C2605', 'C2393', 'C1673', 'C1017', 'C2539',
       'C1933', 'C2316', 'C2051', 'C2414', 'C1301', 'C1700', 'C1636',
       'C2612', 'C1757', 'C2507', 'C1163', 'C2627', 'C2040', 'C2609',
       'C2001', 'C1065', 'C1363', 'C2579', 'C1048', 'C1210', 'C1320',
       'C1941', 'C1326', 'C1685', 'C2618', 'C1451', 'C2143', 'C1968',
       'C2470', 'C1258', 'C2453', 'C1659', 'C1724', 'C1802', 'C1939',
       'C1284', 'C2595', 'C2351', 'C2506', 'C1697', 'C2259', 'C1786',
       'C1357', 'C2570', 'C1652', 'C1565', 'C1910', 'C2359', 'C2139',
```

```
'C1979', 'C1803', 'C2508', 'C2531', 'C1695', 'C2556', 'C2086',
'C1544', 'C2154', 'C2496', 'C1756', 'C2362', 'C2568', 'C2245',
'C2059', 'C2549', 'C1584', 'C2298', 'C2225', 'C1218', 'C2328',
'C1045', 'C1207', 'C1970', 'C1732', 'C2433', 'C1894', 'C1156',
'C2142', 'C2153', 'C2186', 'C1176', 'C2446', 'C2586', 'C2035',
'C2020', 'C2437', 'C2532'], dtype=object)
```

우선 train 데이터 셋 확인

In [7]:

```
train.loc[ ((train['단지코드']=='C2675') |
          (train['단지코드']=='C2335') |
          (train['단지코드']=='C1327') ) , :]
```

Out[7]:

	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	10분내지하철수	10분내버스정류장수	단지내주차면수	등록차량수

테스트 데이터 셋 확인

In [8]:

```
test.loc[ ((test['단지코드']=='C2675') |
          (test['단지코드']=='C2335') |
          (test['단지코드']=='C1327') ) , :].head(3)
```

Out[8]:

	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	10분내지하철수	10분내버스정류장수	단지내주차면수
579	C2675	512	아파트	경기도	국민임대	36.65	130	9.0	A	18476000	154790	0.0	3.0	1016.0
580	C2675	512	아파트	경기도	국민임대	46.90	44	9.0	A	34082000	232200	0.0	3.0	1016.0
581	C2675	512	아파트	경기도	국민임대	46.90	80	9.0	A	34082000	232200	0.0	3.0	1016.0

테스트 데이터 셋에서 세개의 코드 데이터를 없애기

In [9]:

```
test = test.loc[ ~((test['단지코드']=='C2675') |
          (test['단지코드']=='C2335') |
          (test['단지코드']=='C1327') ) , :]
test.head()
```

Out[9]:

	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	10분내지하철수	10분내버스정류장수	단지내주차면수
0	C1072	754	아파트	경기도	국민임대	39.79	116	14.0	H	22830000	189840	0.0	2.0	683.0
1	C1072	754	아파트	경기도	국민임대	46.81	30	14.0	A	36048000	249930	0.0	2.0	683.0
2	C1072	754	아파트	경기도	국민임대	46.90	112	14.0	H	36048000	249930	0.0	2.0	683.0
3	C1072	754	아파트	경기도	국민임대	46.90	120	14.0	H	36048000	249930	0.0	2.0	683.0
4	C1072	754	아파트	경기도	국민임대	51.46	60	14.0	H	43497000	296780	0.0	2.0	683.0

확인

In [10]:

```
test.loc[ ((test['단지코드']=='C2675') |
          (test['단지코드']=='C2335') |
          (test['단지코드']=='C1327') ) , :]
```

Out[10]:

단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	10분내지하철수	10분내버스정류장수	단지내주차면수

오류 데이터 처리

- ※ 동일한 단지에 코드가 2개로 부여된 단지 코드 (3쌍) : ['C2085', 'C1397'], ['C2431', 'C1649'], ['C1036', 'C2675']

- - (참고 사항) 주차면수는 하나의 단지임을 전제로 산정된 것이고 총세대수는 두 개 단지의 합계입니다. 다만 등록차량대수는 ['C2085', 'C1397'] 단지의 경우 동일 수치

In [11]:

```
train.loc[ train['단지코드']=='C2085',  "총세대수" ] = 1339
train.loc[ train['단지코드']=='C1397',  "총세대수" ] = 1339
```

- 단지코드를 C2085,C1397 => N2085로 변경

In [12]:

```
print( train.loc[ train['단지코드']=='C2085', : ].shape  )
print( train.loc[ train['단지코드']=='C1397', : ].shape  )
```

```
(8, 15)
(6, 15)
```

변경 후, 처리 후, 단지코드를 N을 붙여 N2085로 변경

In [13]:

```
train.loc[ train['단지코드']=='C2085',  "단지코드" ] = 'N2085'
train.loc[ train['단지코드']=='C1397',  "단지코드" ] = 'N2085'
```

In [14]:

```
train.loc[ train['단지코드']=='N2085', : ].shape
```

Out[14]:

```
(14, 15)
```

오류 코드 변경

- C2431, C1649의 총세대수를 1047로 변경
- C2431, C1649의 등록차량대수를 1214로 변경
- C2431, C1649의 단지코드를 N2431로 변경

In [15]:

```
a = train.loc[ train['단지코드']=='C2431', : ]
b = train.loc[ train['단지코드']=='C1649', : ]

print(  a.shape, b.shape )
print( a['총세대수'], b['총세대수'])
print( a['등록차량수'], b['등록차량수'])
```

```
(2, 15) (4, 15)
2372    472
2373    472
Name: 총세대수, dtype: int64 2315    575
2316    575
2317    575
2318    575
Name: 총세대수, dtype: int64
2372    359.0
2373    359.0
Name: 등록차량수, dtype: float64 2315    855.0
2316    855.0
2317    855.0
2318    855.0
Name: 등록차량수, dtype: float64
```

In [16]:

```
train.loc[ train['단지코드']=='C2431',  "총세대수" ] = 1047
train.loc[ train['단지코드']=='C1649',  "총세대수" ] = 1047

train.loc[ train['단지코드']=='C2431',  "등록차량수" ] = 1214
train.loc[ train['단지코드']=='C1649',  "등록차량수" ] = 1214

train.loc[ train['단지코드']=='C2431',  "단지코드" ] = 'N2431'
train.loc[ train['단지코드']=='C1649',  "단지코드" ] = 'N2431'
```

In [17]:

```
train.loc[ train['단지코드']=='N2431', : ].shape
```

Out[17]:

```
(6, 15)
```

오류 코드 변경

- C1036의 총세대수를 1243로 변경
- C1036의 단지코드를 N1036로 변경

In [18]:

```
a = train.loc[ train['단지코드']=='C2675', : ]
b = train.loc[ train['단지코드']=='C1036', : ]
a.shape, b.shape
```

Out[18]:

```
((0, 15), (7, 15))
```

In [19]:

```
train.loc[ train['단지코드']=='C1036',  "총세대수" ] = 1243
train.loc[ train['단지코드']=='C1036',  "단지코드" ] = 'N1036'
```

In [20]:

```
train.loc[ train['단지코드']=='N1036', : ].shape
```

Out[20]:

```
(7, 15)
```

오류 3

3. 단지코드 등 기입 실수로 데이터 정제 과정에서 매칭 오류 발생
 - (오류 내용) 단지코드 등 기입 실수로 총세대수가 주차면수에 비해 과하게 많거나 적은 경우가 발생하였고, 점검 결과 일부 데이터의 단지코드, 총세대수, 주차면수 등에서 오류가 검출되었습니다.
 - (발생 원인) 원천데이터 수집 과정에서 단지 코드 등이 잘못 기입되었고 이를 인지하지 못한 채 데이터 정제를 하여 오류가 발생하였습니다.
 - (관련 데이터) 아래와 같이 총 9개 단지에서 같은 문제가 확인되었습니다.
※ 실수가 발생한 단지 코드 (9개 단지) : ['C2335', 'C1327', 'C1095', 'C2051', 'C1218', 'C1894', 'C2483', 'C1502', 'C1988']
 - C2335, C1327 단지는 테스트셋, 나머지는 트레인셋 입니다.

오류 처리

- train 데이터 셋에 오류 발생 코드를 ERR04로 변경 후, 데이터 셋을 두개로 분리

In [21]:

```
train.loc[ train['단지코드']=='C1095',  "단지코드" ] = 'ERR04_1095'
train.loc[ train['단지코드']=='C2051',  "단지코드" ] = 'ERR04_2051'
train.loc[ train['단지코드']=='C1218',  "단지코드" ] = 'ERR04_1218'
train.loc[ train['단지코드']=='C1894',  "단지코드" ] = 'ERR04_1894'
train.loc[ train['단지코드']=='C2483',  "단지코드" ] = 'ERR04_2483'
train.loc[ train['단지코드']=='C1502',  "단지코드" ] = 'ERR04_1502'
train.loc[ train['단지코드']=='C1988',  "단지코드" ] = 'ERR04_1988'
```

In [22]:

```
train.loc[ train['단지코드'].str.contains('ERR'), :].shape
```

Out[22]:

(56, 15)

In [23]:

```
train.loc[ train['단지코드'].str.contains('ERR'), :]
```

Out[23]:

	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	10분내지하철수	10분내버스정류장수	단지내주차면수	등록차량수
0	ERR04_2483	900	아파트	경상북도	국민임대	39.72	134	38.0	A	15667000	103680	0.0	3.0	1425.0	1015.0
1	ERR04_2483	900	아파	경상북	국민임	39.72	15	38.0	A	15667000	103680	0.0	3.0	1425.0	1015.0

데이터 오류 처리 후, csv파일을 만들기

In [24]:

```
train_df = train.copy()
train_df_errno = train.loc[ ~train['단지코드'].str.contains('ERR'), :]
test_df = test.copy()
```

In [25]:

```
train_df.to_csv("train_df.csv", index=False)
train_df_errno.to_csv("train_df_errno.csv", index=False)

test_df.to_csv("test_df.csv", index=False)
```

데이터 전처리 후, 오류 2,3번 해결된 csv파일 생성됨.

- train_df.csv : 오류 2번 해결, 3번은 ERR 코드를 붙임.
- train_df_errno.csv : 오류 2,3번 해결
- test_df.csv : 오류 코드 제외 csv파일