대회 첫 모델 만들기

- 데이터 살펴보기

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.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2952 entries, 0 to 2951
Data columns (total 15 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   단지코드                            2952 non-null   object 
 1   총세대수                            2952 non-null   int64  
 2   임대건물구분                          2952 non-null   object 
 3   지역                            2952 non-null   object 
 4   공급유형                            2952 non-null   object 
 5   전용면적                            2952 non-null   float64
 6   전용면적별세대수                         2952 non-null   int64  
 7   공가수                             2952 non-null   float64
 8   자격유형                            2952 non-null   object 
 9   임대보증금                           2383 non-null   object 
 10  임대료                             2383 non-null   object 
 11  도보 10분거리 내 지하철역 수(환승노선 수 반영)  2741 non-null   float64
 12  도보 10분거리 내 버스정류장 수              2948 non-null   float64
 13  단지내주차면수                         2952 non-null   float64
 14  등록차량수                           2952 non-null   float64
dtypes: float64(6), int64(2), object(7)
memory usage: 346.1+ KB
```

In [4]:

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1022 entries, 0 to 1021
Data columns (total 14 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   단지코드                            1022 non-null   object
 1   총세대수                            1022 non-null   int64
 2   임대건물구분                          1022 non-null   object
 3   지역                            1022 non-null   object
 4   공급유형                            1022 non-null   object
 5   전용면적                            1022 non-null   float64
 6   전용면적별세대수                         1022 non-null   int64
 7   공가수                             1022 non-null   float64
 8   자격유형                            1020 non-null   object
 9   임대보증금                            842 non-null    object
 10  임대료                             842 non-null    object
 11  도보 10분거리 내 지하철역 수(환승노선 수 반영)  980 non-null    float64
 12  도보 10분거리 내 버스정류장 수              1022 non-null   float64
 13  단지내주차면수                         1022 non-null   float64
dtypes: float64(5), int64(2), object(7)
memory usage: 111.9+ KB
```

결측치가 얼마나 될까?

In [5]:

```
train.isna().sum()
```

Out[5]:

```
단지코드                              0
총세대수                              0
임대건물구분                            0
지역                              0
공급유형                              0
전용면적                              0
전용면적별세대수                          0
공가수                               0
자격유형                              0
임대보증금                             569
임대료                               569
도보 10분거리 내 지하철역 수(환승노선 수 반영)    211
도보 10분거리 내 버스정류장 수                 4
단지내주차면수                           0
등록차량수                             0
dtype: int64
```

In [6]:

```
test.isna().sum()
```

Out[6]:

```
단지코드                                0
총세대수                                0
임대건물구분                              0
지역                                  0
공급유형                                0
전용면적                                0
전용면적별세대수                            0
공가수                                 0
자격유형                                2
임대보증금                               180
임대료                                 180
도보 10분거리 내 지하철역 수(환승노선 수 반영)     42
도보 10분거리 내 버스정류장 수                   0
단지내주차면수                             0
dtype: int64
```

- 임대보증금, 임대료, 지하철역수, 버스정류장수(train only), 자격유형(test only)

In [7]:

```
train.head()
```

Out[7]:

	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	도보10분거리내지하철역수(환승노선수반영)	도보10분거리내버스정류장수	단지내주차면수	등록차량수
0	C2483	900	아파트	경상북도	국민임대	39.72	134	38.0	A	15667000	103680	0.0	3.0	1425.0	1015.0
1	C2483	900	아파트	경상북도	국민임대	39.72	15	38.0	A	15667000	103680	0.0	3.0	1425.0	1015.0
2	C2483	900	아파트	경상북도	국민임대	51.93	385	38.0	A	27304000	184330	0.0	3.0	1425.0	1015.0
3	C2483	900	아파트	경상북도	국민임대	51.93	15	38.0	A	27304000	184330	0.0	3.0	1425.0	1015.0
4	C2483	900	아파트	경상북도	국민임대	51.93	41	38.0	A	27304000	184330	0.0	3.0	1425.0	1015.0

In [8]:

```
test.head()
```

Out[8]:

	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	도보 10분거리 내 지하철역 수(환승노선 수 반영)	도보 10분거리 내 버스정류장 수	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

In [9]:

```
train.columns
```

Out[9]:

```
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 [10]:

```
sel = ['총세대수']
X_train = train[sel]
# X_train = train['총세대수']
print(X_train.shape)
X_test = test[sel]
y_train = train['등록차량수']
```

```
(2952, 1)
```

In [11]:

```
from sklearn.linear_model import LinearRegression
```

In [12]:

```
model = LinearRegression()
model.fit(X_train, y_train)
pred = model.predict(X_test)
pred
```

Out[12]:

```
array([524.31256846, 524.31256846, 524.31256846, ..., 424.88994317,
       424.88994317, 424.88994317])
```

In [13]:

```
test.shape, sub.shape
```

Out[13]:

```
((1022, 14), (150, 2))
```

In [14]:

```
test.columns
```

Out[14]:

```
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'],
      dtype='object')
```

In [15]:

```
len( test['단지코드'].unique() )
```

Out[15]:

```
150
```

In [16]:

```
test['등록차량수'] = pred
```

In [17]:

```
test.groupby('단지코드')['등록차량수'].mean()
```

Out[17]:

```
단지코드
C1003     451.081925
C1006     725.028675
C1016     494.646140
C1019     408.586771
C1030     342.839551
             ...
C2653     557.720709
C2675     459.634409
C2676    1010.200560
C2688     362.884435
C2691     527.252485
Name: 등록차량수, Length: 150, dtype: float64
```

In [18]:

```
import numpy as np
```

In [19]:

```
test['코드별차량수평균'] = test.groupby('단지코드')['등록차량수'].transform(np.mean)
test.head(10)
```

Out[19]:

	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	도보10분거리내지하철역수(환승노선수반영)	도보10분거리내버스정류장수	단지내주차면수	등록차량
0	C1072	754	아파트	경기도	국민임대	39.79	116	14.0	H	22830000	189840	0.0	2.0	683.0	524.3125
1	C1072	754	아파트	경기도	국민임대	46.81	30	14.0	A	36048000	249930	0.0	2.0	683.0	524.3125
2	C1072	754	아파트	경기도	국민임대	46.90	112	14.0	H	36048000	249930	0.0	2.0	683.0	524.3125
3	C1072	754	아파트	경기도	국민임대	46.90	120	14.0	H	36048000	249930	0.0	2.0	683.0	524.3125
4	C1072	754	아파트	경기도	국민임대	51.46	60	14.0	H	43497000	296780	0.0	2.0	683.0	524.3125
5	C1072	754	아파트	경기도	국민임대	51.71	51	14.0	H	43497000	296780	0.0	2.0	683.0	524.3125
6	C1072	754	아파트	경기도	국민임대	51.96	198	14.0	H	43497000	296780	0.0	2.0	683.0	524.3125

7	C1072	754	아파트	경기도	국민임대	51.96	67	14.0	H	43497000	296780	0.0	2.0	683.0	524.3125
8	C1128	1354	아파트	경기도	국민임대	39.79	368	9.0	H	22830000	189840	0.0	3.0	1216.0	684.6716
9	C1128	1354	아파트	경기도	국민임대	39.79	30	9.0	H	22830000	189840	0.0	3.0	1216.0	684.6716

In [20]:

```
test.drop_duplicates(['단지코드'], keep='first')
```

Out[20]:

	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	도보 10분거리내지하철역수(환승노선수반영)	도보 10분거리내버스정류장수	단지내주차면수	등록
0	C1072	754	아파트	경기도	국민임대	39.79	116	14.0	H	22830000	189840	0.0	2.0	683.0	524.
8	C1128	1354	아파트	경기도	국민임대	39.79	368	9.0	H	22830000	189840	0.0	3.0	1216.0	684.
17	C1456	619	아파트	부산광역시	국민임대	33.40	82	18.0	A	19706000	156200	0.0	16.0	547.0	488.
26	C1840	593	아파트	전라북도	국민임대	39.57	253	7.0	A	14418000	108130	0.0	3.0	543.0	481.
30	C1332	1297	아파트	경기도	국민임대	39.99	282	11.0	H	28598000	203050	0.0	2.0	1112.0	669.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
996	C2456	349	아파트	제주특별자치도	국민임대	26.44	24	17.0	H	6992000	117000	0.0	4.0	270.0	416.

1000	C1266	596	아파트	충청북도	국민임대	26.94	164	35.0	H	8084000	149910	0.0	1.0	593.0	482.
1005	C2152	120	아파트	강원도	영구임대	24.83	66	9.0	C	-	-	0.0	1.0	40.0	354.
1007	C1267	675	아파트	경상남도	국민임대	24.87	28	38.0	H	6882000	104370	0.0	1.0	467.0	503.
1018	C2189	382	아파트	전라북도	국민임대	29.19	96	45.0	H	6872000	106400	0.0	2.0	300.0	424.

150 rows × 16 columns

In [21]:

```
test_new = test.drop_duplicates(['단지코드'], keep='first').reset_index()
test_new
```

Out[21]:

	index	단지코드	총세대수	임대건물구분	지역	공급유형	전용면적	전용면적별세대수	공가수	자격유형	임대보증금	임대료	도보 10분거리내지하철역수(환승노선수반영)	도보 10분거리내버스정류장수	단지내주차면수
0	0	C1072	754	아파트	경기도	국민임대	39.79	116	14.0	H	22830000	189840	0.0	2.0	683.0
1	8	C1128	1354	아파트	경기도	국민임대	39.79	368	9.0	H	22830000	189840	0.0	3.0	1216.0
2	17	C1456	619	아파트	부산광역시	국민임대	33.40	82	18.0	A	19706000	156200	0.0	16.0	547.0
3	26	C1840	593	아파트	전라북도	국민임대	39.57	253	7.0	A	14418000	108130	0.0	3.0	543.0
4	30	C1332	1297	아파트	경기도	국민임대	39.99	282	11.0	H	28598000	203050	0.0	2.0	1112.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
145	996	C2456	349	아파트	제주특별자치도	국민임대	26.44	24	17.0	H	6992000	117000	0.0	4.0	270.0

146	1000	C1266	596	아파트	충청북도	국민임대	26.94	164	35.0	H	8084000	149910	0.0	1.0	593.0
147	1005	C2152	120	아파트	강원도	영구임대	24.83	66	9.0	C	-	-	0.0	1.0	40.0
148	1007	C1267	675	아파트	경상남도	국민임대	24.87	28	38.0	H	6882000	104370	0.0	1.0	467.0
149	1018	C2189	382	아파트	전라북도	국민임대	29.19	96	45.0	H	6872000	106400	0.0	2.0	300.0

150 rows × 17 columns

In [22]:

```
sub_df = test_new[ ['단지코드', '코드별차량수평균']]
sub_df.columns = ['code', 'num']
sub_df
```

Out[22]:

	code	num
0	C1072	524.312568
1	C1128	684.671641
2	C1456	488.231777
3	C1840	481.282884
4	C1332	669.437530
...	...	...
145	C2456	416.070194
146	C1266	482.084679
147	C2152	354.866481
148	C1267	503.198624
149	C2189	424.889943

150 rows × 2 columns

In [23]:

```
sub_df.to_csv('baseline_0712.csv', index=False)
```

In [24]:

```
import os
os.listdir(os.getcwd())
```

Out[24]:

```
['.git',
 '.ipynb_checkpoints',
 '01_competition_firstmodel.ipynb',
 '01_대회_첫모델만들기-Copy1.ipynb',
 '01_대회_첫모델만들기.md',
 '02_second_data.md',
 '02_second_datapreprocessing.ipynb',
 '03_second_linear_model-Copy2.ipynb',
 '03_second_linear_model.ipynb',
 '03_second_linear_ridge_lasso.ipynb',
 '04_second_rf_model.ipynb',
 '05_second_etc_model.ipynb',
 'baseline_0712.csv',
 'fourth_rf_0714.csv',
 'README.md',
 'second_rf_0712.csv',
 'test_df.csv',
 'third_rf_0714.csv',
 'train_df.csv',
 'train_df_errno.csv',
 'Untitled.ipynb']
```