1. 전용면적별 세대수 합계와 총세대수가 일치하지 않는 경우

- (오류 내용) 제공한 데이터(트레인 및 테스트) 중 총 58개 단지에서 전용면적별 세대수 합계가 총세대수보다 적은 경우가 발생하였습니다.

- (분포 현황) 차이는 최소 1세대부터 최대 452세대까지이며, 그 가운데 48개 단지(83%) 는 차이가 1~14세대, 나머지 10개 단지(17%) 는 차이가 94~452세대 사이에 분포합니다.

- (발생 원인) 차이는 분양상가 또는 분양아파트가 임대단지 안에 혼합되어 발생한 것으로, 분양 완료된 상가 ․ 아파트는 관리대상에서 제외되어 전용면적별 세대수에서는 포함되지 않았으나 총세대수에는 포함되었습니다.

- (관련 데이터) 차이가 14세대 이하인 48개 단지는 분양상가, 차이가 94~452세대에 분포하는 10개 단지는 분양아파트 혼합단지인 것으로 확인되며, 아래 10개 단지의 코드를 확인하셔서 데이터 분석에 참고해주시기 바랍니다.

※ 차이가 14세대 이하인 48개 단지 : 하단 별첨 참조

※ 차이가 94~452세대인 10개 단지(크기순) : ['C1490', 'C2497', 'C2620', 'C1344', 'C1024', 'C2470', 'C1206', 'C1740', 'C2405', 'C1804’]

2. 동일한 단지에 단지코드가 2개로 부여된 경우

- (오류 내용) 동일한 단지임에도 불구하고 코드가 2개로 부여되어 별개의 단지로 인식되도록 데이터가 제공되었고, 그로 인해 주차면수는 2개 단지에 합산값으로 기재되었고 총세대수는 2개 단지에 각각 계상된 형태로 나타났습니다.

- (발생 원인) 1번과 마찬가지로 분양과 임대가 혼합된 유형의 단지임으로 인해 발생한 오류이나 1번에서는 분양세대에 단지코드가 부여되지 않고 전용면적별 세대수에서 빠져있는 반면, 2번에서는 분양세대에도 단지코드가 부여되어 별개의 단지인 것처럼 인식되었습니다.

- (참고 사항) **주차면수는 하나의 단지임을 전제로 산정된 것이고 총세대수는 두 개 단지의 합계입니다. 다만 등록차량대수는 ['C2085', 'C1397'] 단지의 경우 동일 수치, ['C2431', 'C1649']] 단지의 경우 두 개 단지 합계입니다.**



- (관련 데이터) 아래와 같이 총 3개 단지 6개 코드에서 같은 유형의 오류가 확인되었습니다.

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

- C2675 단지는 테스트셋, 나머지는 트레인셋 입니다.

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.

# bad string cleaning (공란)

df = df.replace('-', np.nan)

대부분의 임대상가는 임대료나 임대보증금이 없다

아파트 임대료 임대보증금은 등록차량수와 관계가 없는것으로 보인다. 따라서 두 변수는 제외해도 좋을듯? 이라고 생각했지만 Heatmap으로 봤을 때 중간정도의 상관관계를 보임

df\_shops = df[df['rental\_building classification'] == '상가']

df\_apt = df[df['rental\_building classification'] == '아파트']

print("상가 임대료 missing value count: {}".format(df\_shops.rent.isna().sum()), "out of {}".format(df\_shops.shape[0]))

print("상가 임대보증금 missing value count:{}".format(df\_shops.rental\_deposit.isna().sum()) , "out of {}".format(df\_shops.shape[0]))

print("아파트 임대료 missing value count: {}".format(df\_apt.rent.isna().sum()), "out of {}".format(df\_apt.shape[0]))

print("아파트 임대보증금 missing value count:{}".format(df\_apt.rental\_deposit.isna().sum()), "out of {}".format(df\_apt.shape[0]))

missing number of subway와 bus는 0으로 -> city에 위치해 있지 않기때문에 없다고 채워도 될 듯

# impute missing number of subways and bus with 0. All units with missing values in these columns are not located in cities, so we can safely assume there are no subways

df\_apt.nearby\_bus.fillna((df\_apt.rental\_deposit.median()), inplace=True)

df\_apt.nearby\_subways.fillna((df\_apt.rental\_deposit.median()), inplace=True)

# impute missing rental\_deposit values with median

df\_apt.rental\_deposit.fillna((df\_apt.rental\_deposit.median()), inplace=True)

# impute missing rent values with median

df\_apt.rent.fillna((df\_apt.rent.median()), inplace=True)

주차등록수는 0.8이상의 강한 coreelation을 보임

Train 데이터 기준 차집합 임대건물구분 : []

Train 데이터 기준 차집합 지역 : ['서울특별시']

Train 데이터 기준 차집합 공급유형 : ['공공분양', '공공임대(5년)', '장기전세']

Train 데이터 기준 차집합 자격유형 : ['O', 'F', 'B']

IN test data

임대료 있고, 임대보증금 있는데 공란이네, c2411은 A로, c2253은 C로 넣으면 되겠네

지역에 맞춰서 데이터를 병합 하셨네

* 그러나 충남 북부나, 경남 남부 같은 경우는 지하철이 있는 도시가 있습니다. 마찬가지로 대전은 대전지하철이 있습니다. 결측치를 어떻게 채울지 고심해봐야 합니다.
* test set에서는 5개 단지에서 지하철 정보가 없습니다. 전부 충남과 대전입니다.

1. **데이터 오류 제거하기**

2번오류 :  ['C2085', 'C1397'], ['C2431', 'C1649'], ['C1036', 'C2675']

3번 오류 :

['C2335', 'C1327', 'C1095', 'C2051', 'C1218', 'C1894', 'C2483', 'C1502', 'C1988']

* Train dataset에서 제거

1. **지역에 맞춰서 데이터 병합하기 -> 제공된 데이터가 지역/ 나이로 되어 있기에 train set과 공통된 지역으로 합치는게 맞지 않을까?**
2. **결측치 제거하기**

**Train**

임대보증금 569

임대료 569

도보 10분거리 내 지하철역 수(환승노선 수 반영) 211

도보 10분거리 내 버스정류장 수 4

**Test**

자격유형 2 -> c2411은 A로, c2253은 C

임대보증금 180

임대료 180

도보 10분거리 내 지하철역 수(환승노선 수 반영) 42

4. Train Dataset에만 있는 값들

Train 데이터 기준 차집합 임대건물구분 : []

Train 데이터 기준 차집합 지역 : ['서울특별시']

Train 데이터 기준 차집합 공급유형 : ['공공분양', '공공임대(5년)', '장기전세']

Train 데이터 기준 차집합 자격유형 : ['O', 'F', 'B']

->그냥 냅두자…

1. 숫자형 데이터를 Column별로 정규화? 아니면 전체로?
   * 열별로 하면 임대료안에서의 ZSCORE 값이 생성되는거고
   * 전체 데이터를 대상으로 하면 전체 데이터의 ZSCORE값으로 변환
2. 면적은 구간화 후 범주형으로? / 아니면 숫자형 데이터 그대로?
3. 새로운 Feature 생성할 필요가 있어보인다

[Feature selection — scikit-learn 0.24.2 documentation](http://scikit-learn.org/stable/modules/feature_selection.html)

KFOLD 로 검증해서 제출

**데이터 전처리 방법**

* 결측치 제거
  + 임대보증금과 임대료를 각 지역마다 평균을 내서 그 값으로 NaN값을 채움
  + 버스 NaN값은 오류데이터 삭제하니까 없어짐
  + 지하철 NaN값은 0으로 채움
* 오류데이터 제거
  + 공지에 나온 오류데이터 제거 했습니다.
* Age Data 지역으로 맞춰서 병합
* 같은 단지코드끼리 병합(병합한게 더 좋게 나온다)
* 정규화는 못했음 ㅜ
* One-Hot Encoding (공급유형, 지역, 신분)
  + Train Dataset : (411,68)
  + Test Dataaset : (150,68)
* 단지코드랑 임대건물구분 제거 (임대건물 구분은 아파트밖에 없다)

**Training Model : LightGBM**

* 데이터의 수가 적긴 하지만 일단 train\_test\_split으로 나눔 20%
* RandomizedSearchCV를 이용하여 최적의 파라미터 자동검색
* KFold로 검증도 해봄