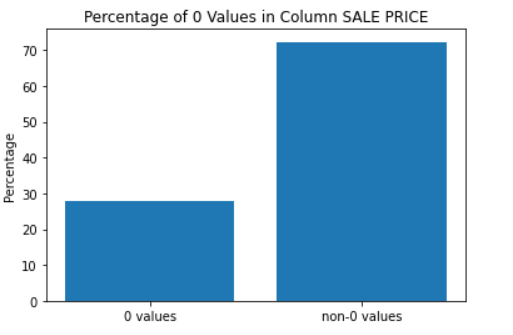
# Big Data Coursework 2 Report

### Data Exploration

**Before Data cleaning**



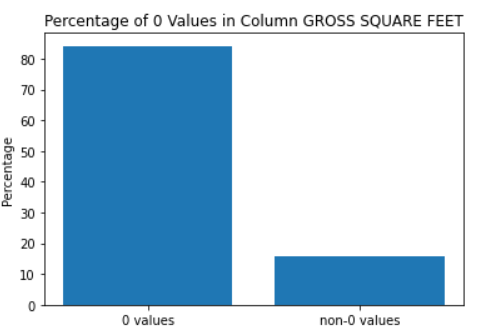
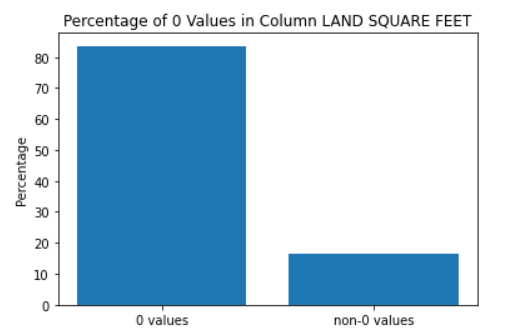
**Description**: as the bar chart shows, the Sale Price column with 0 takes a large proportion of all data.

### 

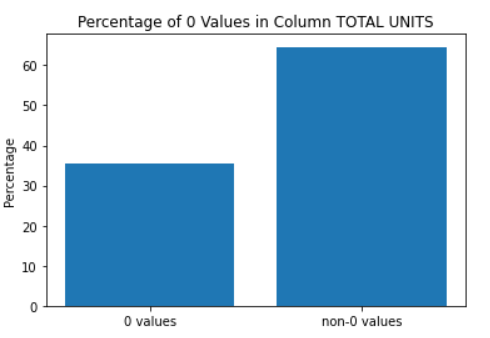
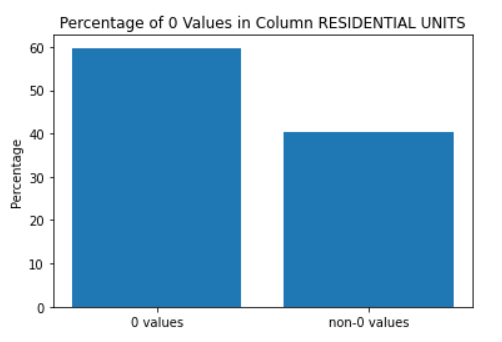
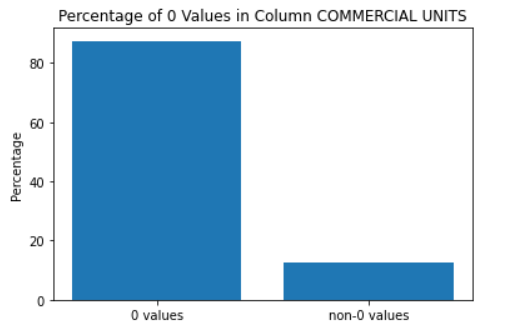
**Description**: as the bar chart shows, the Apartment Number column with a null value takes a large proportion of all data.

### 

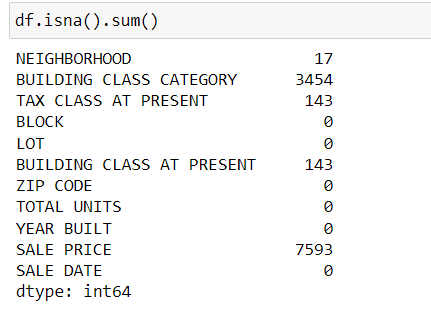
**Description:** as the graph shows, column BOROUGH and EASEMENT can be dropped due to only containing a single value. Column ADDRESS contains too many different values and it is not suitable as a predictor, BLOCK and LOT have the same information as ADDRESS, so ADDRESS can be dropped either. “BUILDING CLASS AT TIME OF SALE” has the same value as “BUILDING CALSS AT PRESENT”, so one of them can be dropped, “TAX CLASS AT PRESENT” provide more information than “TAX CLASS AT TIME OF SALE”, so TAX CLASS AT TIME OF SALE column can be dropped. Because the APARTMENT NUMBER contains a lot of null-value, and the rest is a very small set, and at the same time, it contains same information as BLOCK and LOT, so this column can be dropped to-o when training a model.



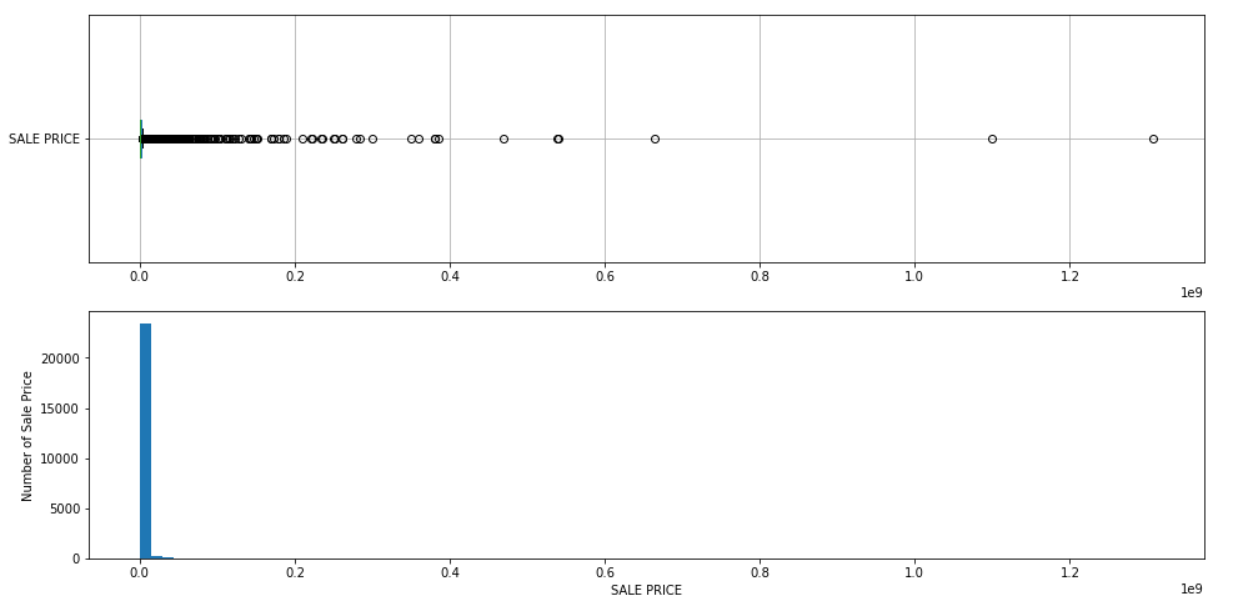
**Description**: as the bar chart shows, both the LAND SQUARE FEET and GROSS SQUARE FEET columns contain tons of 0-values, so these two columns can be dropped too. Or later we can use other methods to fill the missing value, such as mode, mean or KNNImputer.

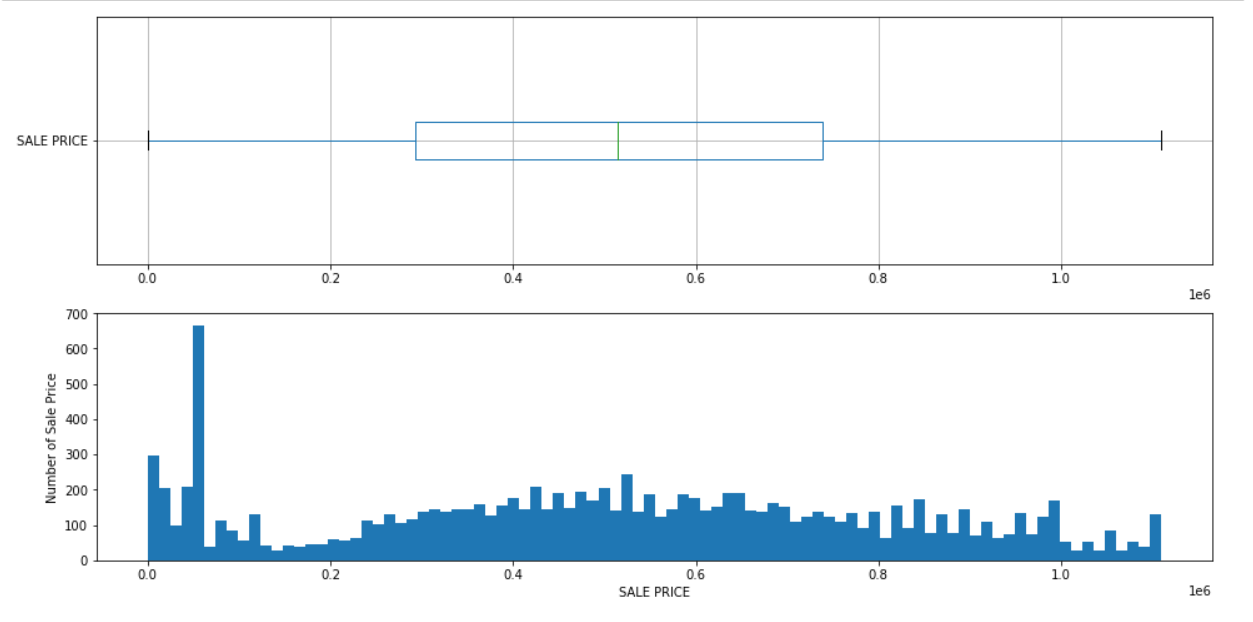


**Description**: as the bar chart shows, COMMERCIAL UNITS and RESIDENTIAL UNITS contain too many 0-values, but TOTAL UNITS contain less compare to them, for a small dataset, we can drop COMMERCIAL UNITS and RESIDENTIAL UNITS, but for the big dataset, we should find a way to deal with them properly.

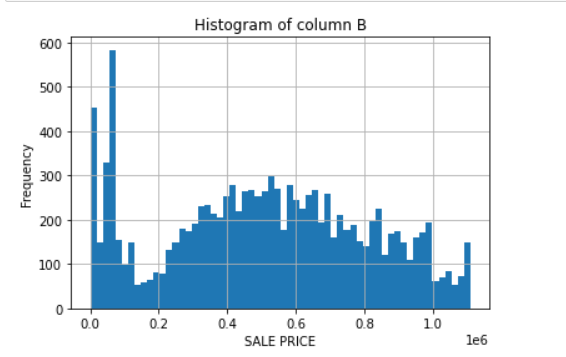


**Description**: as the figure shows, there are still a lot of rows containing NaN values, so we can delete these rows before training a model



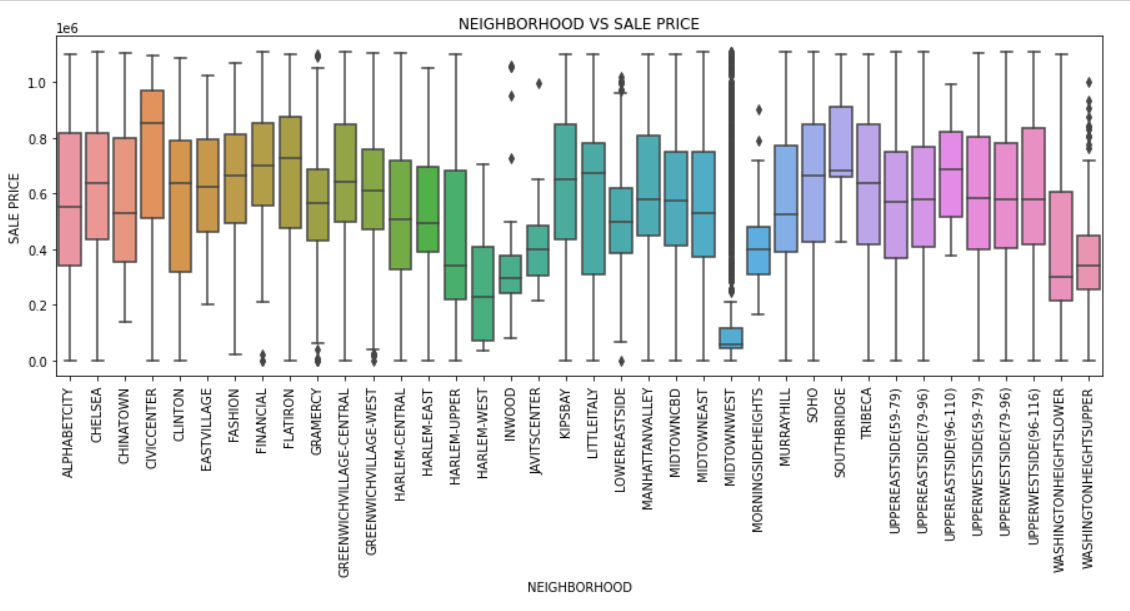


**Description**: The picture above shows the outlier in the original dataset, and the picture below shows the distribution of prices after removing enough outliers. Now we can based on the cleaned dataset explore some business logic relations between different features and the sale prices.

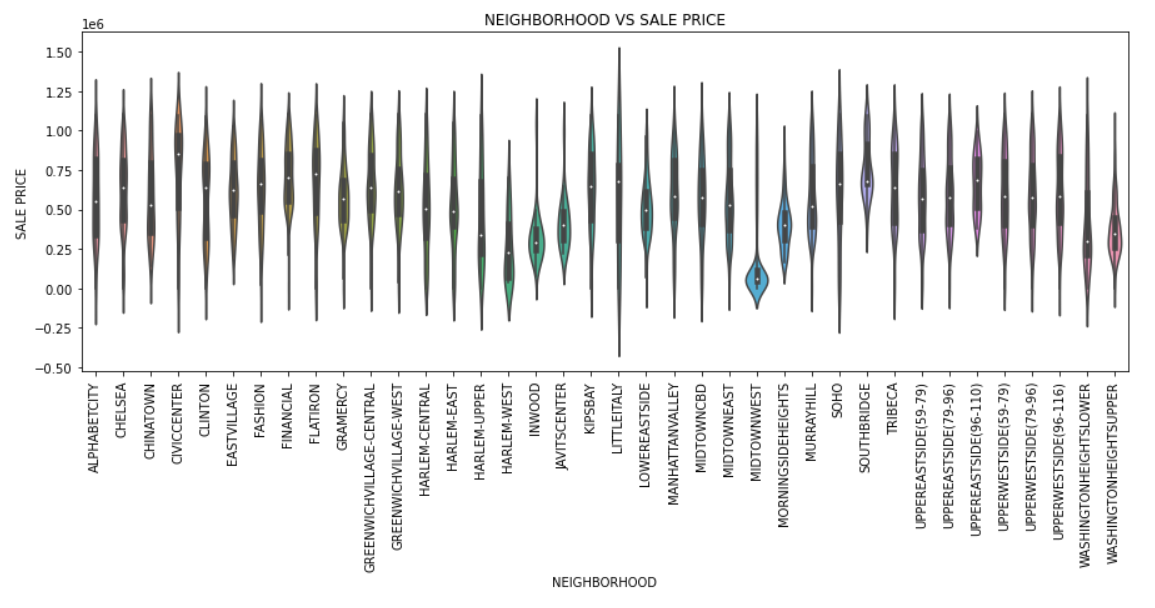


**Description**: This graph shows the distribution of the SALE PRICE value after sifting out the outliers, obviously, most prices are concentrated in 0~100000.

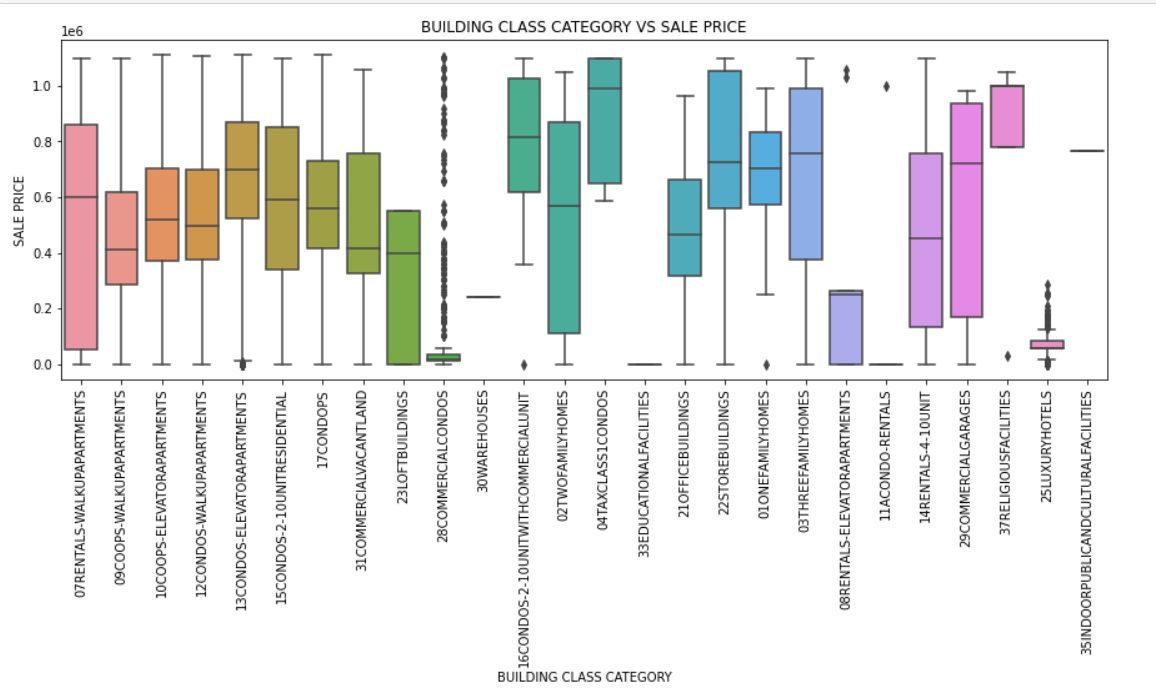
**After Data cleaning**



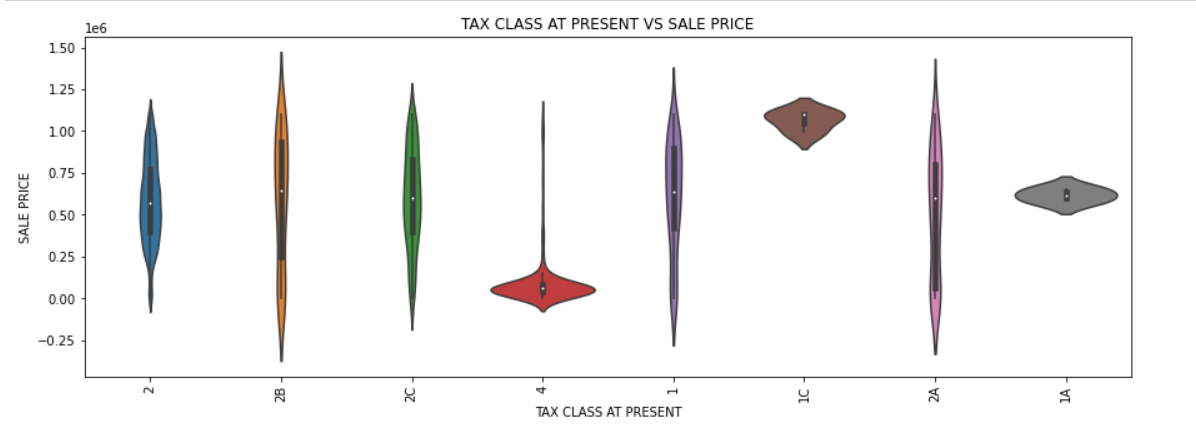
**Description**: most Neighborhood prices are at the same level, but still significantly low than other neighbourhoods, such as HARLEM-WEST and MIDTOWNWEST etc,



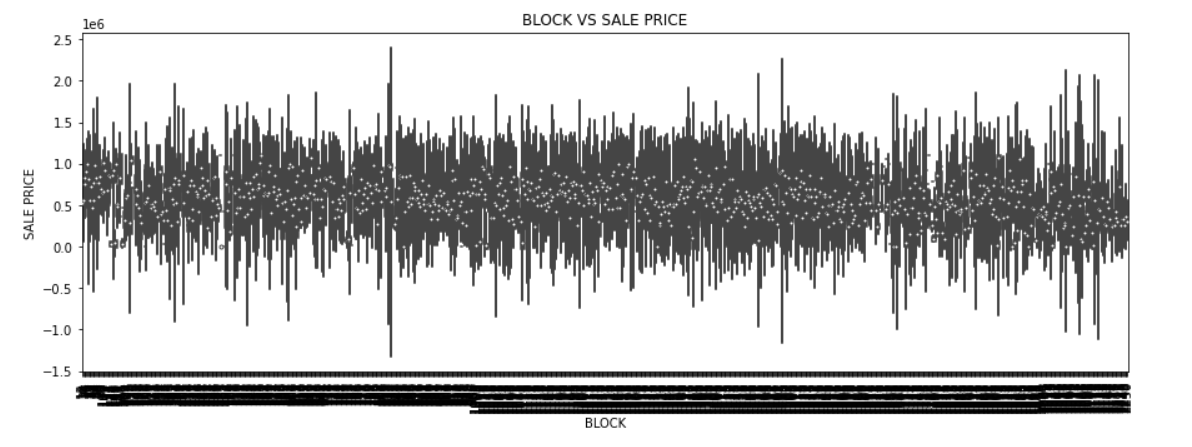
**Description**: most Neighborhood prices are very evenly distributed. But MIDTOWN WEST mainly locate at an interval.

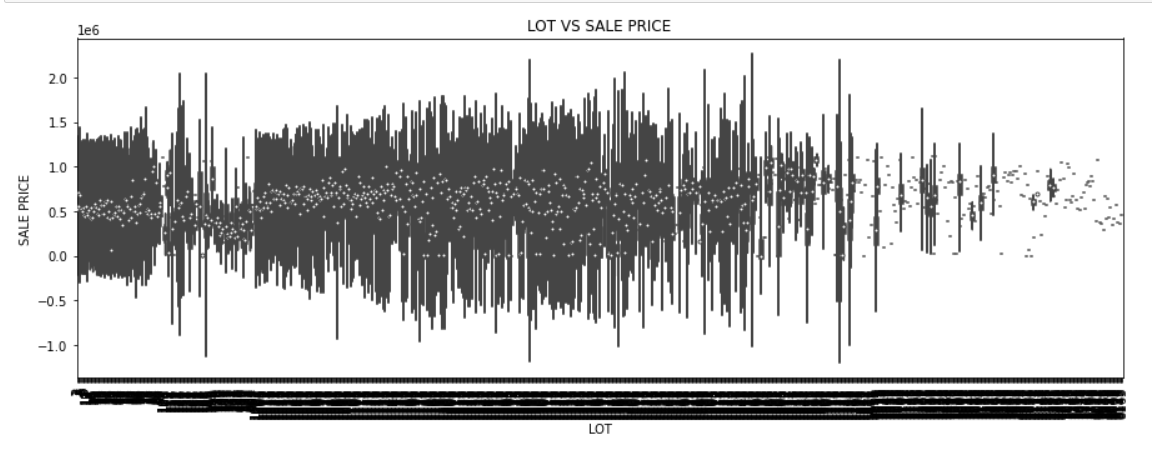


**Description**: for BUILDING CLASS CATEGORY columns, 28COMMERCIALCONDOS contains a lot of outliers, same with the 25LUXURYHOTELS. For 33EDUCATIONALFACILITIES and 11ACONDO\_RENTALS, no accuracy prices exist.



**Description**: For the TAX CLASS, if a property at the 4 TAX CLASS, it most probably has a lower price. But for 1C, while has a higher price. For 1A, located between 4 and 1C, for the other TAX CLASS, the price is evenly distributed.





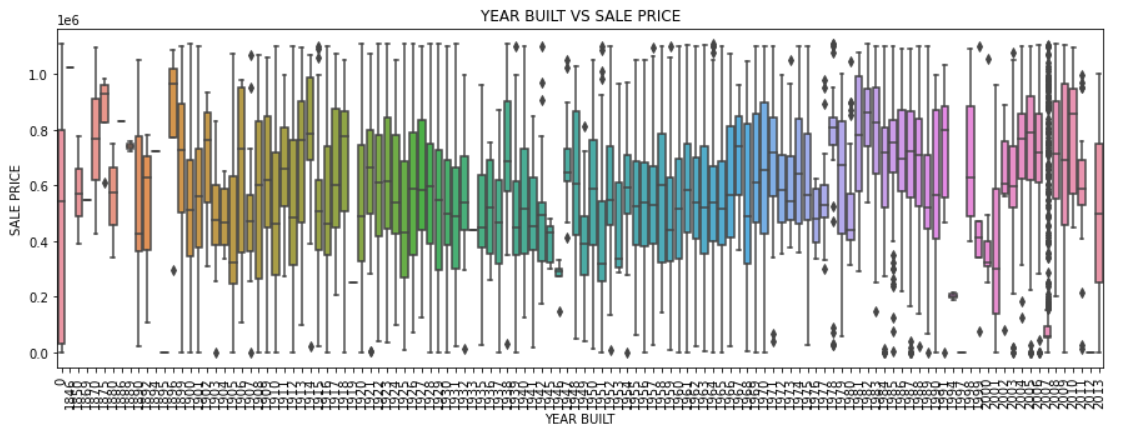
**Description**: LOT with a low number has high density, and for high value has a low density. This means the dataset is not sufficient for a high LOT value, this is different with BLOCK. So if we want to build a better model, we should handle this issue carefully.



**Description**: First, the data is not evenly distributed over this category, and RH and H2 contain many outliers. R6, C8, C2 and C9 have a higher sale price. And L1, RG, H2 and D6 have a lower price.



**Description**: When the TOTAL UNITS value exceeds 21, Sample data is scarce, for the TOTAL UNITS value is 2, sale prices concentrate on a low interval. Also, after cleaning data, still, some noise data exist. For example, 0 shouldn’t exist.



**Description**: From the graph, SALE PRICE goes up slightly with the growth of the year. But for some years, the price decreased sharply. E.g., 1999 – 2002, for every year, sale prices are normally distributed.

### Model

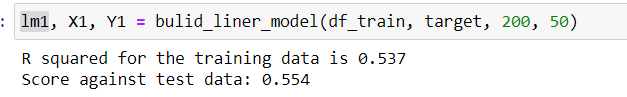
#### Small Dataset(i.e., remove all NaN values)

**Data Cleaning**

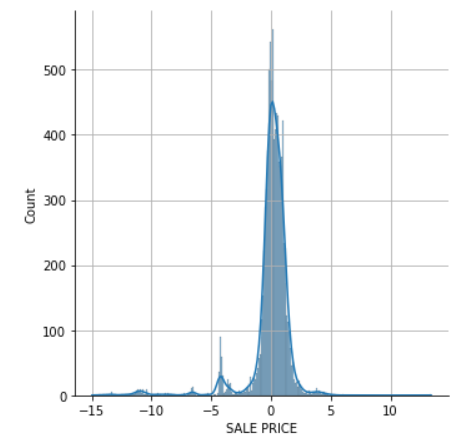
* Code can be found under the small\_dataset folder and the file small\_set\_cleaning

**Liner Moel with Small Dataset**

* Code can be found under the small\_dataset folder and the file Liner\_model\_with\_samll\_dataset.
* Performance:



* Residential Graph:

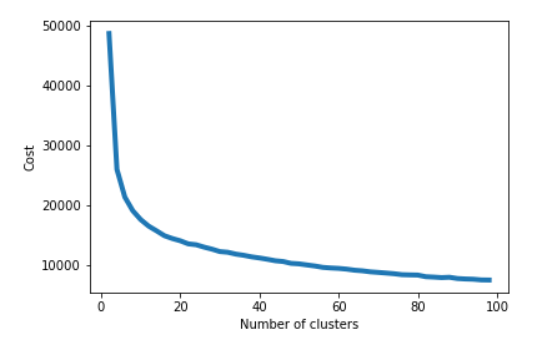


**Random Forest Moel with Small Dataset**

* Code can be found under the small\_dataset folder and the file Liner\_model\_with\_samll\_dataset.
* Performance: for the training set is 0.8, for test set is 0.6

**KMeans Moel with Small Dataset**

* Code can be found under the small\_dataset folder and the file Liner\_model\_with\_samll\_dataset.
* Elbow graph



* Performance: Silhouette score: -0.55
* Local Regressor based on clustering:
  + For the Liner model, the r2 score is 0.62. improving by 17%.

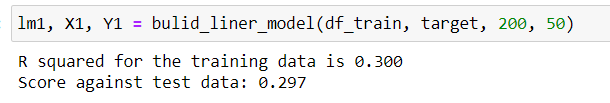
#### Big Dataset(i.e., Filling missing value with KNNImputer or other methods)

**Data Cleaning**

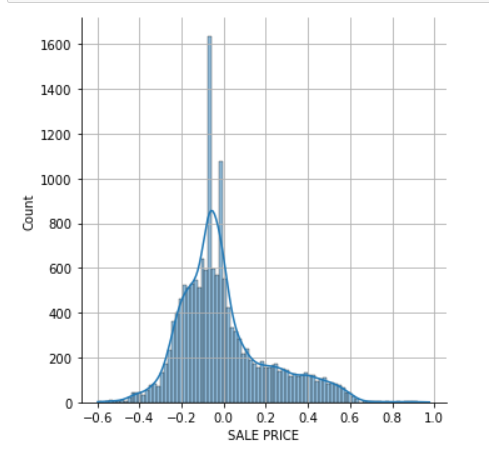
* Code can be found under the big\_dataset folder and the file Big\_dataset\_cleaning.

**Liner Moel with Big Dataset**

* Code can be found under the small\_dataset folder and the file Liner\_model\_with\_samll\_dataset.
* Performance:



* Residential Graph:

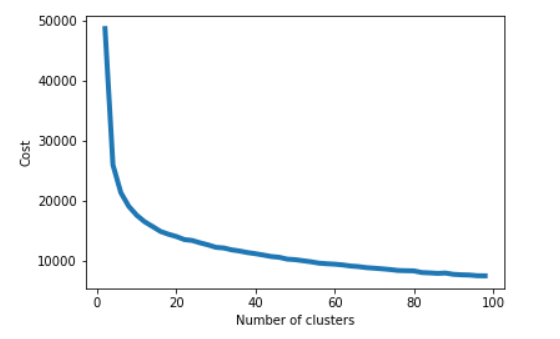


**Random Forest Moel with Small Dataset**

* Code can be found under the small\_dataset folder and the file Liner\_model\_with\_samll\_dataset.
* Performance: for the training set is 0.8, for test set is 0.6

**KMeans Moel with Small Dataset**

* Code can be found under the small\_dataset folder and the file Liner\_model\_with\_samll\_dataset.
* Elbow graph



* Performance: Silhouette score: -0.68
* Local Regressor based on clustering:
  + For the Liner model, the r2 score is 0.47. improving by 56%.