﻿**Descriptive Analysis**

Since the variable I’m trying to predict is the sale price. I decided to explore the summary statics of the price column and take a closer look at the likes of the mean and minimum and see if there was anything I could learn.



A picture containing table

Description automatically generated

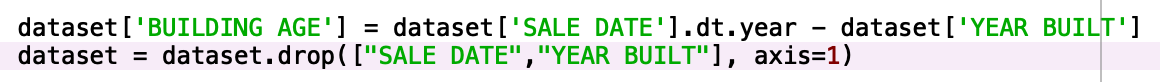
Looking at the summary statistics for sale price, I can see that the 25th percentile sales were 0. Looking at the glossary, these would be cases such as the house being incorrectly transferred or incorrect data, so this could be filtered out. Furthermore, I also discovered that the highest recorded sale was 1.3 billion, which is obviously an anomaly considering the mean house sale is 1.8 million, so this can also be filtered.

**Cleaning**

I initially take actions upon the first few things I notice while looking at the data, which is the column ‘easement’ being empty and borough containing the number 1 throughout the dataset. I also noticed ‘apartment name’ and ‘sale price’ are named incorrectly, so I rectify this both of these.



I also thought that it would be easier to look at the age of the building sold rather than the year built, so I created and new column and dropped the ‘sale date’ and ‘year built’ column.

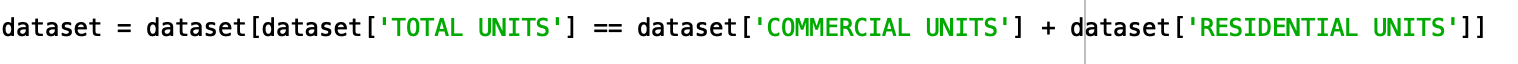


After taking a closer look at the glossary, I came to the conclusion that of the numerical values, ‘sales price’ and ‘total units’ couldn’t be null, so I replace the zeros with NaN and drop these.

A screenshot of a cell phone

Description automatically generated

After dropping the duplicates, I removed ﻿the address, zip code, apartment number,block and lot columns since these were only helpful in spotting duplicates and wouldn’t be useful for predicting. The totsl units is also supposed be the sum of commercial and residential units, so I filtered the dataset where the total of commercial and residential units is equal to total.



Upon looking at land square feet and gross square feet, I notice that over 90% of these columns are null and it would be useless to try predict the price with these columns, so I dropped these.



**Visualisation**

Initially the sales prices have an abnormal distribution, which is bad for the model as it can lead to skewed predictions. To solve this, I log-transform this value into a new column called ‘lnprice’. I then remove the outliers from lnprice, which helps to make the distribution more normal.

**Before After**

**A close up of a map

Description automatically generatedA close up of a map

Description automatically generated**

After this, I attempted to find the features that correlated with each other the most by creating a heatmap. From the heatmap obsessive, I can see that there isn’t much correlation between the independent values bar total units and residential units, which means there could be multicollinearity among these two features. Furthermore, no features have greater than 0.5 correlation.

A picture containing bus

Description automatically generated

Since I’m attempting to predict the sales price, it would useful to see what features correlate with the price, so I made listed these features in order of correlation. As seen from the output, the main feature correlating with sales price is the building age, with commercial units the only column having a negative correlation.

**A screenshot of a cell phone

Description automatically generated**

Plotting tax class against the sale price, you can see that prices are usually higher in 2A and 2B, indicating that primarily residentials, such as cooperatives and condominiums, tend to be priced higher.

**A screenshot of a cell phone

Description automatically generated**

Since Manhattan has various neighbourhoods, it would be useful to look at how the sales price varies across different neighbourhoods, whether that could be due to the general population being upper class. After visualising, I can see that prices in Tribeca and Soho are kthe most expensive, with sales generally being over the 600k mark. But these seem to be exceptions, as the mean prices of all other neighbourhoods seems to be underneath 600k. **A screenshot of a cell phone

Description automatically generated**

It also seems that most of the buildings in the dataset are around 100 years old.**A screenshot of a cell phone

Description automatically generated**

**Removing Outliers**

I decided to remove outliers for sales because this is the dependent variable I’m looking to predict, so it would be hard to build a model that was capable of predicting both normal and outlier sales. A screenshot of a cell phone

Description automatically generated

The method I created uses the interquartile rate to detect lower and upper bound prices, mark the temp column with and then filter to only include rows where temp is equal to zero.

I also noticed that the majority of the values for total units was 1, so I removed this value.



**Encoding**

Since I’m doing linear regression, all columns will need to be numbers, so I performed on hot encoding on all of the categorical values.

A screenshot of a cell phone

Description automatically generated

**Model Building & Evaluation**

For my model, after encoding, I now have around 80 columns, which would make it tedious and difficult to search through my dataset and calculate which columns would be useful to include in my dataset, so I picked the top 30 columns which correlated the most with sale price.

﻿The mathematical equation used to calculate the log price for my model is ﻿:

﻿14.7148 \* COMMERCIAL UNITS \* 1.0135 + TOTAL UNITS \* 0.8519 + NEIGHBORHOOD\_CHELSEA \* 0.3159 + NEIGHBORHOOD\_FASHION \* 0.3132 + NEIGHBORHOOD\_FLATIRON \* 0.0132 + NEIGHBORHOOD\_GREENWICH VILLAGE \* 0.2795 + NEIGHBORHOOD\_HARLEM \* -1.3206 + NEIGHBORHOOD\_HARLEM-EAST \* -1.3537 + NEIGHBORHOOD\_INWOOD \* -1.4430 + NEIGHBORHOOD\_JAVITS CENTER \* -2.6283 +

NEIGHBORHOOD\_KIPS BAY \* 0.5731 + NEIGHBORHOOD\_MANHATTAN VALLEY \* -0.7450 +

NEIGHBORHOOD\_SOHO \* 0.4216 + NEIGHBORHOOD\_SOUTHBRIDGE \* -0.2654 +

NEIGHBORHOOD\_TRIBECA \* 0.6702 + NEIGHBORHOOD\_UPPER EAST SIDE \* 0.3246 +

NEIGHBORHOOD\_UPPER WEST SIDE \* 0.1477 + NEIGHBORHOOD\_WASHINGTON HEIGHTS LOWER \* -0.5848 + NEIGHBORHOOD\_WASHINGTON HEIGHTS UPPER \* -1.1091 + TAX CLASS AT PRESENT\_1 \* 0.3194 + BUILDING CLASS CATEGORY\_08 RENTALS - ELEVATOR APARTMENTS \* 0.4874 + BUILDING CLASS CATEGORY\_10 COOPS - ELEVATOR APARTMENTS \* -0.8776 + BUILDING CLASS CATEGORY\_11A CONDO-RENTALS \* 0.9642 + BUILDING CLASS CATEGORY\_21 OFFICE BUILDINGS \* 0.1486 + BUILDING CLASS CATEGORY\_22 STORE BUILDINGS \* 0.4173 + BUILDING CLASS CATEGORY\_25 LUXURY HOTELS \* -3.7073

BUILDING CLASS CATEGORY\_26 OTHER HOTELS \* 0.9692 + BUILDING CLASS CATEGORY\_29 COMMERCIAL GARAGES \* 1.6511 + BUILDING CLASS CATEGORY\_30 WAREHOUSES \* 1.5080 + BUILDING CLASS CATEGORY\_35 INDOOR PUBLIC AND CULTURAL FACILITIES \* 0.0000

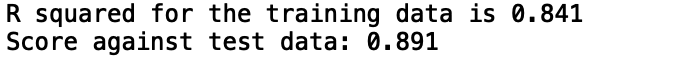
**A screenshot of a cell phone

Description automatically generated**

After I plot the graph, you can see that the model is linear. You can also see that the values are missing for the first portion of the graph, then the point start to appear and become linear as the price increases.

**A screenshot of a social media post

Description automatically generated**My r squared for test data ended up being 0.84, with the r score for training data being 0.89. This greatly improved upon when I initially built my model and attempted to replace the NaN with the mean and had over 10k rows.

****