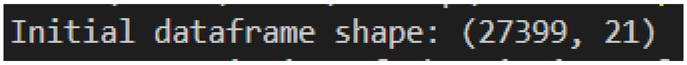
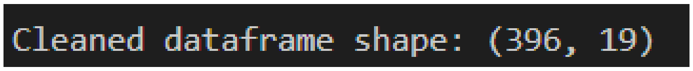
Report on dataset and findings

Initial data exploration and cleansing

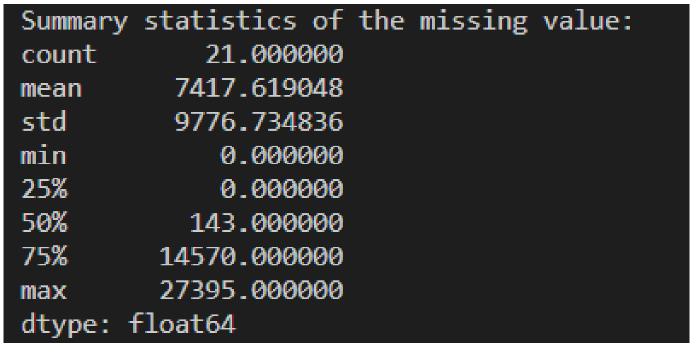


```
Initial dataframe shape: (27399, 21)
```

Taking a look at the initial data set without performing any cleansing or analysis allows us to recognise a set of important patterns and facts about said dataset. Firstly,it is mainly empty, just by a simple visual scan the percentage of empty to non-empty fields is staggering, which in turn means that those null values must be handled carefully so as to give us a proper representation of the data. Thus, after performing the cleaning presented in the below snippets.



```
Cleaned dataframe shape: (396, 19)
```

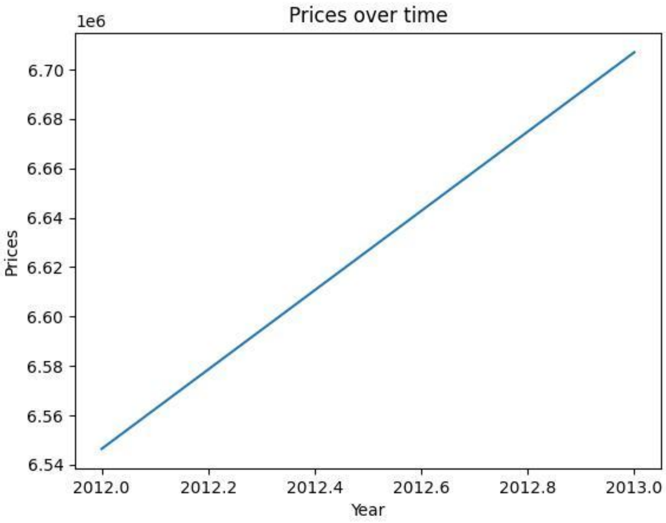


```
Summary statistics of the missing value:
count       21.000000
mean      7417.619048
std       9776.734836
min          0.000000
25%          0.000000
50%        143.000000
75%      14570.000000
max      27395.000000
dtype: float64
```

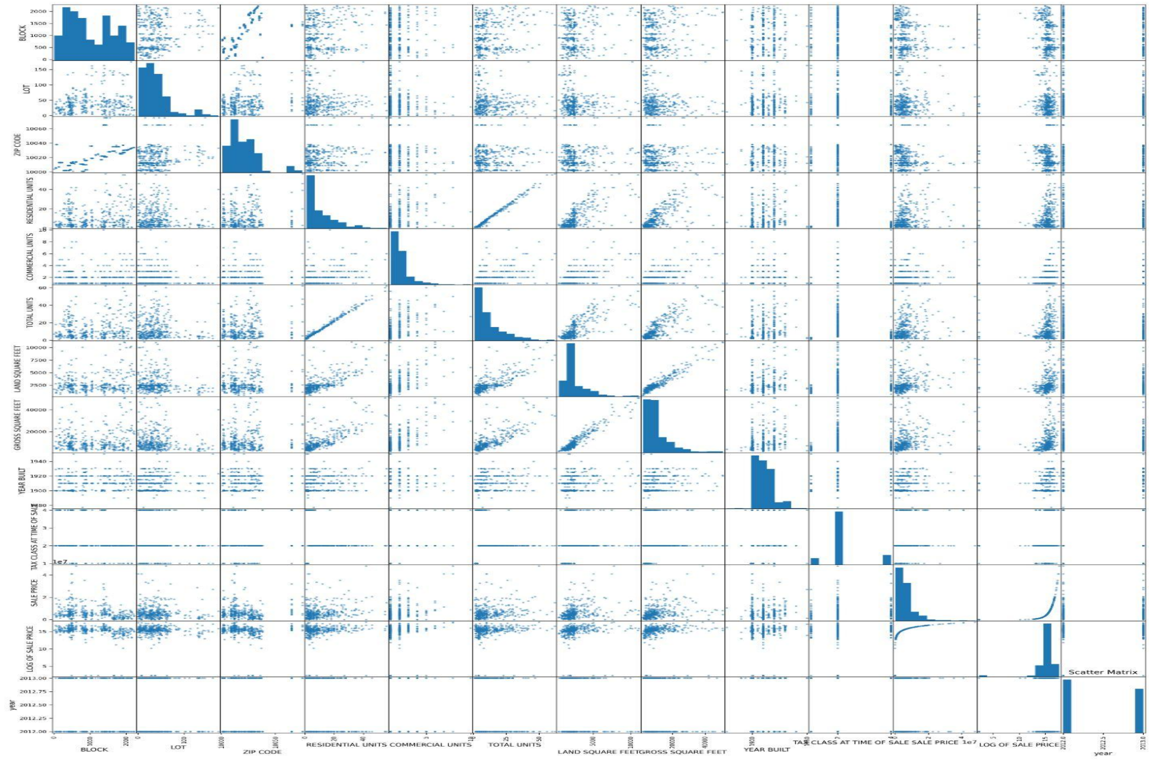
We can see that the actual dataset shape is much much smaller, it being roughly 1.5%. This allows for smaller, more concise and precise data to feed into prediction models.

The summary on the left describes the magnitude of removal of null variables.

Data visualization

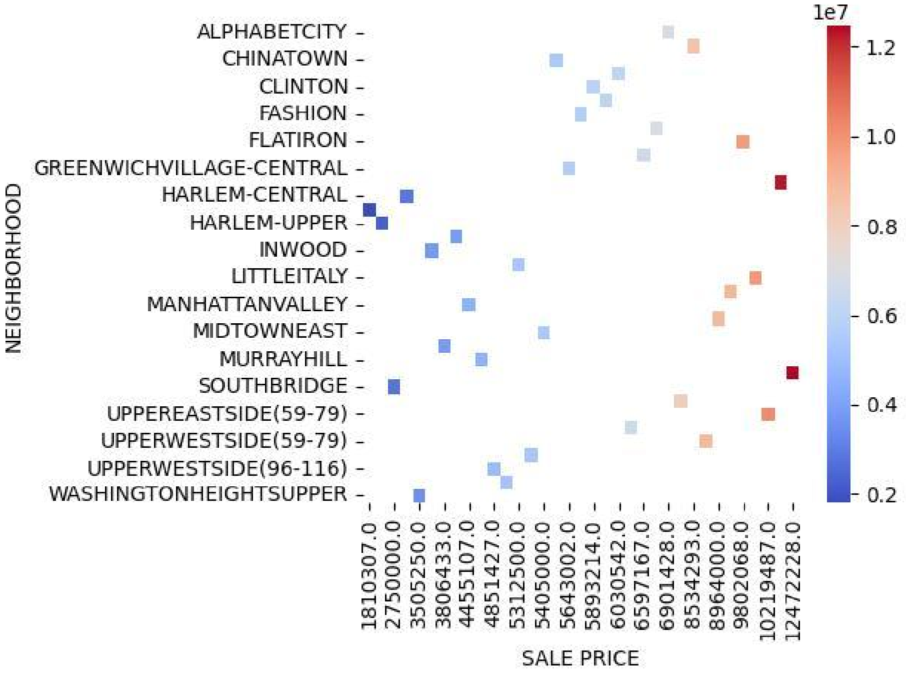


When it comes to data visualization I believe it’s best to first take a look at a big picture graph, such as the trend of house prices over time. Which in our case is a perfect linear acceleration, as more time passes houses get more expensive. This is without any impact from all the other features such as neighborhood, square footage etc. Which means that although those features might be important, they seem to not affect the overall global trend of increasing housing prices.

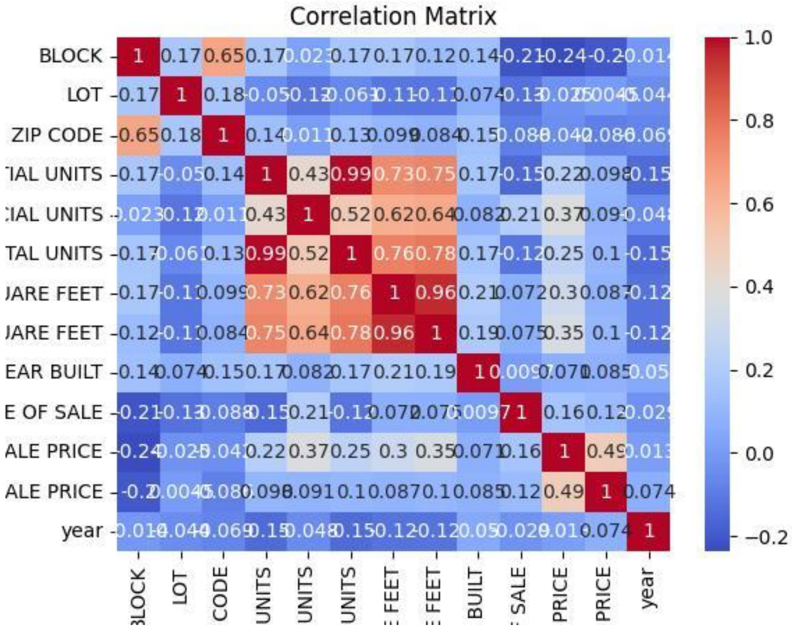


Next I have created a scatter matrix diagram, used to determine the representation of the relationship between multiple pairs of variables, in this case I have made use of only the numerical variables as those are really the only ones that make sense to be plotted on a scatter matrix. Following an inspection of the diagram, it can be seen that there are several variables that have strong correlation such as residential units / total units, gross square footage / land square footage and vice versa. However upon a generalized view of our variable of interested, respectively the sale price of the unit, it can be noticed that there aren’t any pairs that really perform a strong correlation with the price, which in turns results in weak correlation between the plot points of the scatter plots as shown in the matrix above. This is bizarre, perhaps a data error or an error of my own in terms of developing the visualization algorithm, but on the other hand it does make sense as house prices are very market driven and therefore the sentiment of the market usually outweighs any objective attributes, to a certain extent.

In order to take a closer look at some selected variables I have generated respective graphical plots in order to more easily visualize the relationships and patterns found in the data.

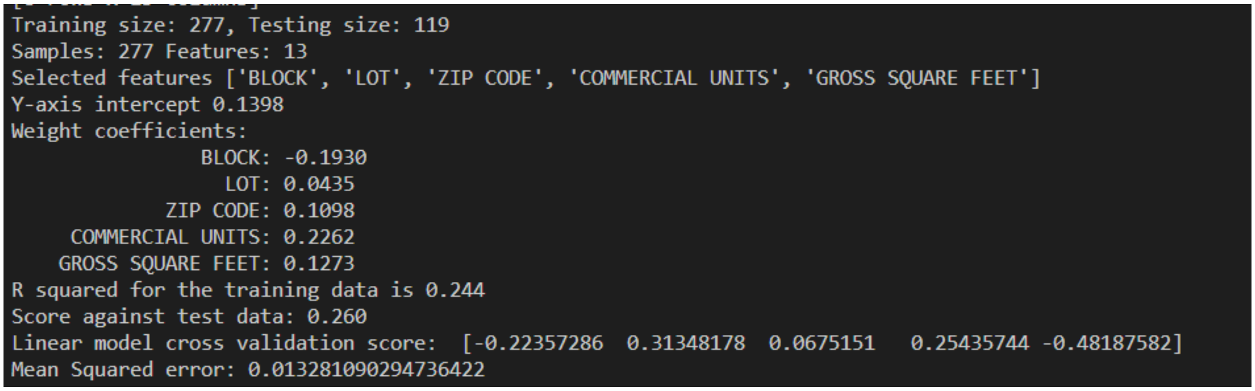


Firstly, a basic general heatmap of the relationship found between the Neighbourhood and sale prices of the listings. Where the darker color represented, indicates a stronger correlation between the neighborhood and the totality of sale prices in that neighborhood. Our heatmap indicates that there are a select few of highly popular and wanted neighborhoods where the sale prices can go up to the top percentiles almost always , whilst all the other neighborhoods are unable to access that top level of the market spectrum, which indicates a pattern of market or environmental forces that create this trend of highly priced and more luxurious neighborhoods compared to the lower tier ones, even though they are found in the same city.



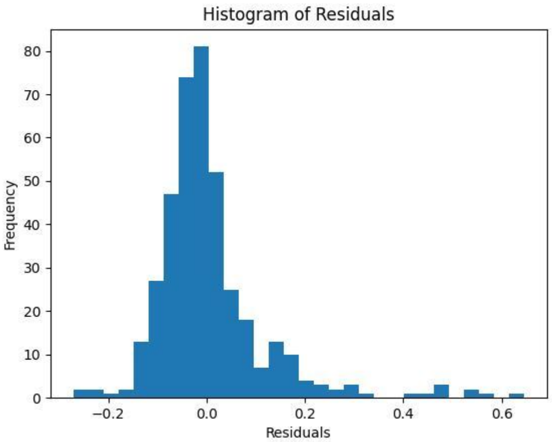
Afterwards, I’ve decided to implement a correlation matrix of all the numerical values from the dataframe. In order to analyze the relationship of the coefficients between the pairs of variables. There it can be seen the big group of high correlations in the middle of the diagram, it representing the pairs of building units to square footage and vice versa, meaning that the more units are built in the same apartment complex or area, the more it affects the square footage of each individual piece and vice versa. Which in turn allows me to decide whether to include or exclude those variables from the linear model.

Linear model



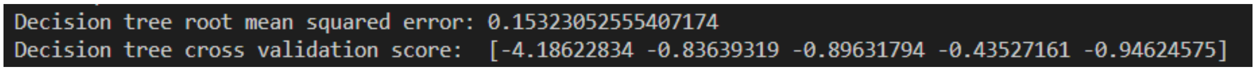
```
Training size: 277, Testing size: 119
Samples: 277 Features: 13
Selected features ['BLOCK', 'LOT', 'ZIP CODE', 'COMMERCIAL UNITS', 'GROSS SQUARE FEET']
Y-axis intercept 0.1398
Weight coefficients:
                BLOCK: -0.1930
                  LOT: 0.0435
             ZIP CODE: 0.1098
     COMMERCIAL UNITS: 0.2262
    GROSS SQUARE FEET: 0.1273
R squared for the training data is 0.244
Score against test data: 0.260
Linear model cross validation score:  [-0.22357286  0.31348178  0.0675151   0.25435744 -0.48187582]
Mean Squared error: 0.013281090294736422
```

The linear model presented above is a simple linear regression with a test/train split of 0.3. Using only numeric and normalized data, in order to reduce abnormalities and ensure model performance. The training and testing size is shown above together with all the weight coefficients and other useful information. In order to evaluate the performance of this linear regression model, I've used a multitude of measurements such as the R-squared value, linear model cross validation and mean squared error. The model seems to be performing relatively well as the mean squared error is relatively low, which means that the model is better at predicting the sale prices. However it seems that the split of train/test data is not optimal, which in turn might mean that the sample size is simply too small for the model, increasing the split also doesn't work so it's more than likely a problem with the sample size, as in the cleaning process I've had to remove a lot of null values, as the initial data set wasn't appropriately recorded.



New model

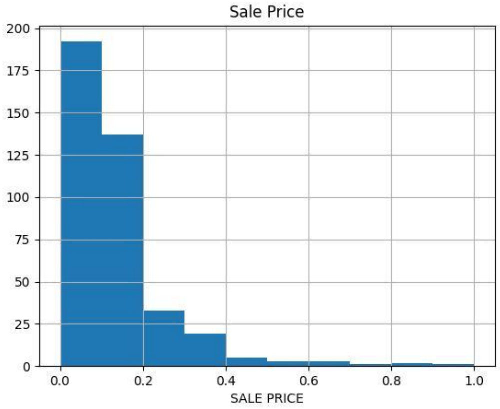
I've decided to use a decision tree model as it can be a good choice for predicting the sale price of a house based on several factors because it is a flexible and interpretable algorithm that can handle both categorical and continuous variables. It also has certain advantages such as handling of interactions between variables which is an important aspect of predicting the sale prices. Based on the assessors below there can be seen that the model is far from optimal, there's a significantly larger MSE compared to the linear model which in turn means that the model is not predicting the prices as accurately as I've wanted.

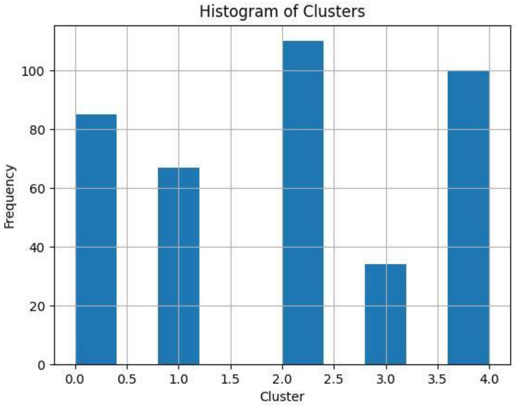


```
Decision tree root mean squared error: 0.15323052555407174
Decision tree cross validation score:  [-4.18622834 -0.83639319 -0.89631794 -0.43527161 -0.94624575]
```

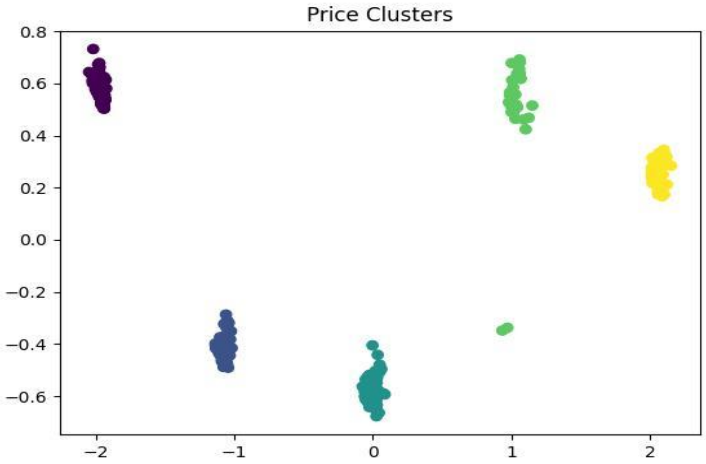
K-Means Algorithm

The use of a K-Means algorithm is appropriate in our dataset context. As it is an efficient algorithm that clusters data points into groups based on their similarity which can then be used to eventually predict the sale price of housing based on said data variables.

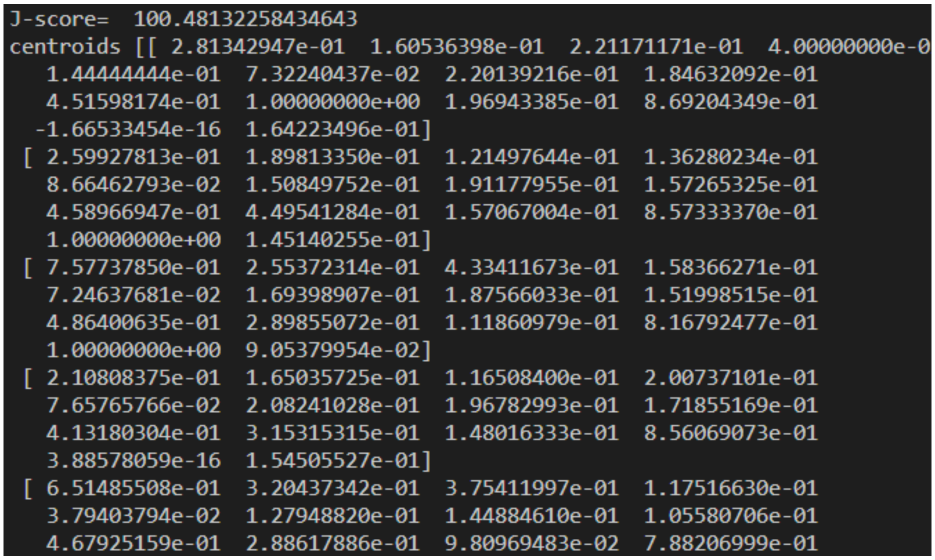




As we can see from the two histograms above, the K-means algorithm does work as it presents the data as being clustered in Fig2 compared to Fig1 which is just the raw normalized sale price data histogram. Meaning that the 5 clusters have been created. Which can then be visualized in the below graph, where they are highlighted even more in order to obtain an overall panoramic view of the way the data fits in these clusters.

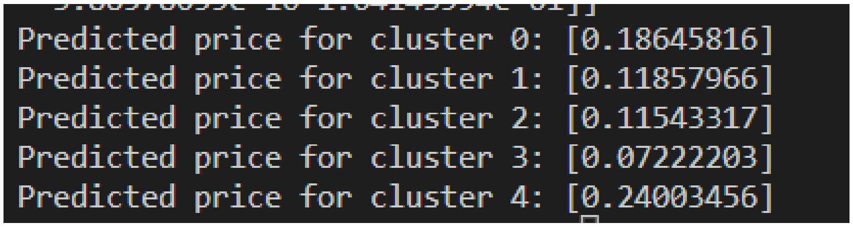


Overall the clustering seems to be working fine, as the clusters have been produced and visualized as seen above. Now I will try to obtain some more in-depth information about the model itself. As seen in the snippets below where the J-score and centroids are printed. Although being very hard to analyze I believe it’s important to have this piece of information in order to obtain a more comprehensive view.



```
J-score=  100.48132258434643
centroids [[ 2.81342947e-01  1.60536398e-01  2.21171171e-01  4.00000000e-0
   1.44444444e-01  7.32240437e-02  2.20139216e-01  1.84632092e-01
   4.51598174e-01  1.00000000e+00  1.96943385e-01  8.69204349e-01
  -1.66533454e-16  1.64223496e-01]
 [ 2.59927813e-01  1.89813350e-01  1.21497644e-01  1.36280234e-01
   8.66462793e-02  1.50849752e-01  1.91177955e-01  1.57265325e-01
   4.58966947e-01  4.49541284e-01  1.57067004e-01  8.57333370e-01
   1.00000000e+00  1.45140255e-01]
 [ 7.57737850e-01  2.55372314e-01  4.33411673e-01  1.58366271e-01
   7.24637681e-02  1.69398907e-01  1.87566033e-01  1.51998515e-01
   4.86400635e-01  2.89855072e-01  1.11860979e-01  8.16792477e-01
   1.00000000e+00  9.05379954e-02]
 [ 2.10808375e-01  1.65035725e-01  1.16508400e-01  2.00737101e-01
   7.65765766e-02  2.08241028e-01  1.96782993e-01  1.71855169e-01
   4.13180304e-01  3.15315315e-01  1.48016333e-01  8.56069073e-01
   3.88578059e-16  1.54505527e-01]
 [ 6.51485508e-01  3.20437342e-01  3.75411997e-01  1.17516630e-01
   3.79403794e-02  1.27948820e-01  1.44884610e-01  1.05580706e-01
   4.67925159e-01  2.88617886e-01  9.80969483e-02  7.88206999e-01
```

Finally, I will then use the K-means algorithm, together with a regressor in order to verify if the clusters are able to be used in predicting the price of a home. As seen below, the regressor works and the predicted prices are listed (in normalized form) based on a set of manually inputted values for the clusters.



```
Predicted price for cluster 0: [0.18645816]
Predicted price for cluster 1: [0.11857966]
Predicted price for cluster 2: [0.11543317]
Predicted price for cluster 3: [0.07222203]
Predicted price for cluster 4: [0.24003456]
```

To conclude, I have been able to explore, visualize and perform models on the dataset provided. Thus being able to form the conclusion that although the data was lacking a lot of samples, as they were either null or incorrect or simply not usable in the context of predicting a house price, I believe it is important to perform data cleansing in even more detail and perform different data formatting such as normalization, clustering etc to aid in the model’s performance. However, the models done by myself are fairly primitive and not able to predict in an overwhelmingly efficient manner.