2023

CO3093 – Big Data and Predictive Analysis

Table of Contents

Introduction

Different data cleaning procedures have been applied to 'Manhattan12.csv' file to build a model. These include two linear regressor models and one cluster-based local regressor. This report aims to justify the decisions made in the building the model, as well as analysing the results and evaluating these models.

Dataset

The dataset 'Manhattan12.csv' had to undergo a lot of cleaning before allowing us to explore it and create a model. Starting with the first four rows of the file which described the data set, see Figure 1 in the appendix. Furthermore, certain columns used later in the model initially contained up too 87.5% missing values, these were removed in the initial linear model but kept and delt with differently later. Additionally, 1 593 duplicate values had to be removed. Lastly, columns had empty spaces that also had to be removed and converted to NaN as well as '0' values in numerical columns and in the 'SALE DATE' column.

Objective

We want to create and train a model that would give us the best performance. We will start with an initial linear regression model and then hope to improve it further. The main issue encountered was the very large presence of NaN values and choosing which features to use in the model to predict the price.

Initial Linear Regression

Data Cleaning

The dataframe used in the this first linear regression model had a shape of 463 rows and 19 columns, which is a great difference from the starting shape of 27 395 rows and 21 columns. This is because of the harsh data cleaning decisions made in the process. After having correctly formatted the initial CSV into a dataframe, the first step was to change column types to their correct and respective type. This is done to allow the linear regressor later to identify numerical variables and treat them as numerical. Secondly, empty cells had to be identified and correctly labelled as NaN, to allow us to identify these NaN values at a later stage and remove them as a Linear Regressor cannot be trained using NaN values. Moreover, a series of boxplot graphs for each numerical variable was used to visually identify the outliers and set the threshold in the 'removing_outliers' function where they are later removed. The entities were grouped by 'BUILDING CLASS CATEGORY' as its one of the best ways to look at similar apartments together. Removing outliers is done to avoid a model being trained on potential entities which are a result of poor sampling or an error when entering the data. Lastly, duplicates were removed to avoid biased results and the 'LOG_PRICE' column got calculated based on the 'SALE PRICE'.

Data Exploration

Identifying any trends in the dataset was the next step. In Figure 2 in the appendix, it is possible to see how Neighbourhoods have can have similar prices[1], Chinatown, Civic Centre, Clinton, and the East Village, for example. This boxplot graph also shows the great spread in prices in some neighbourhoods such as the 'Upper West Side (79-96)'. Figure 3 in the appendix, shows that there is no clear relationship between price and the sale date, but troughs can be noticed and may relate to certain events happening during that date. Figure 4 in the appendix shows no relationship between residential (blue) or commercial (orange) units with log_price. Figure 5 in the appendix, shows a very strong relationship between residential units (blue) and total units but no relationship between commercial units (orange) and total units. Figure 6 in the appendix shows a positive relationship between gross square feet with total units, which should be expected as for each unit added you add space. Figure 7 also shows a strong relationship between gross square feet with residential units (blue), but not with commercial units (orange), this tells us that most of the space is taken by residential units after about 0.1 residential units. The scatter plot matrix in Figure 8 in the appendix, shows all the relationships as well as those highlighted above. The correlation matrix on the other hand in Figure 9 in the appendix, helps us understand the relationship seen in Figures 2-8 using the r^2 value. This helps us confirm relationship stated above such as the very strong one ($r^2=1$) between residential units and total units or between gross square feet and total units ($r^2=0.9$), it also helps us notice that the sale price or log_price does not have a strong relationship with any of the features.

Model Build

To build a linear regressor model only numerical features are used. The first step aside from selecting numerical columns is too select the best features to use in the model. I chose to select four features using the recursive feature elimination (RFE) method as there are seven initial features, two of which are 'SALE PRICE' and 'LOG_PRICE' leaving only five left, therefore letting the RFE method to remove the weakest one. The features selected by the RFE method are 'COMMERCIAL UNITS', 'TOTAL UNITS', 'LAND SQUARE FEET' and 'GROSS SQUARE FEET'. The data was then split into a train set and a test set, with the train set used to train the model.

Model Results & Evaluation

---

[1] All values in all the graphs have been normalised.

The results provided by this initial regression model when predicting the 'LOG_PRICE' are highlighted in Figure 10, see appendix. The coefficients and the y-intercept allow us to create the following equation:

$$y = 0.148*CU + 0.39*RU - 0.174*LSF - 0.633*GSF + 0.822$$

Where CU is 'COMMERCIAL UNITS', TU is 'TOTAL UNITS', LSF is 'LAND SQUARE FEET' and GSF is 'GROSS SQUARE FEET'.

The r^2 value was 0.065, meaning 0.935 (94%) of the variability cannot be explained by the model, making the model completely unreliable. Additionally, cross validation was performed with 8 folds and resulted in an average r^2 value of -0.047, meaning that this model performs worse than a constant (horizontal line) which in this case is the mean of 'LOG_PRICE', 0.82. The MSE for this model is 0.014, being very high. Moreover, the graph showing the residuals is not normally distributed, is skewed right and shows a small peak at $x \approx 0.7$, further proving the great inaccuracy of this model.

Lastly, it must be noted that the data used to build this model is only a small fraction of the initial dataset, therefore the data cleaning method must be improved to allow us to use more entities of the dataset and create a better performing model.

Improved Linear Regression Models

Approach

For this improved model data cleaning was carried out similarly to the initial model but with some key changes. In certain cases, categorical variables were turned into numeric. Moreover, to solve the problem of the great number of NaN values, entities containing *only* NaN values were dropped and the remaining NaN values were imputed using IterativeImputer from the sklearn.impute package. This allowed us to use 25 697 entities instead of 463.

The next step was choosing the appropriate features to use. This was done in 3 main ways:

1. Using RFE method (like in the initial linear model) after imputation.
2. Converting all the columns to numeric before imputing and selecting the features personally using data exploration as a guidance.
3. Converting all the columns to numeric before imputing and using all the columns in the model.

Results & Evaluation

By using the RFE method to select four features after imputing the NaN values the linear model gave a r^2 value of 0.698 using cross validation, see Figure 11 in the appendix. We also get an MSE of

0.00410. The graph of the residuals shows that they are not yet normally distributed, but the graph is not as skewed to the right as before and more residuals are present when x=0, with some smaller peaks to the left.

When we converted the categorical variables to numeric, the correlation matrix heatmap (see Figure 12, in the Appendix) along with the line graphs explained were used to determine which features to choose and train the model with those new features. This had to be done manually due to the great computational cost to use RFE in this scenario. These are the features selected: 'TAX CLASS AT PRESENT', 'LAND SQUARE FEET', 'GROSS SQUARE FEET', 'TAX CLASS AT TIME OF SALE', 'BLOCK'. This model returned a r^2 of 0.413 using cross validation, see Figure 13 in the appendix. Here we get an MSE of 0.0075, being larger than using the RFE method, but still significantly better than the initial linear regression model. The graph of the residuals arguably looks normally distributed than when using the RFE method.

Converting the categorical variables to numeric, and using all the features in this model, gives an r^2 of 0.651 using cross validation, see Figure 14 in the appendix. Here we get an MSE of 0.00423, being larger than using the RFE method but also very similar. This model is again an improvement from the initial linear regression model and performs very similar when using RFE for feature selection. Although the graph of the residuals appears to be normally distributed in this case, which is what we want.

Lastly all five numerical columns were used in this model and returned an r^2 of 0.754 using cross validation, see Figure 15 in the appendix. Here we get an MSE of 0.00251, being the best result, we got so far. This is the best performing linear regression models so far. The graph of the residuals seems to be normally distributed even though there are some peaks on the left side once again. As this is the best performing model, we can use the equation below to help predict the log_price:

$$y = -3.49*CU - 64.6*RU + 63.8*TU + 0.782*LSF + 0.445*GSF + 0.641$$

Where CU is 'COMMERCIAL UNITS', RU is 'RESIDENTIAL UNITS', TU is 'TOTAL UNITS', LSF is 'LAND SQUARE FEET' and GSF is 'GROSS SQUARE FEET'.

It must be mentioned that an additional model was built using a set of data where the 'log_price' or 'SALE PRICE' were dropped if they were NaN values before imputation, therefore not predicting missing values for these columns. This method had a worse performance, with a r^2 using cross validation of 0.525 and an MSE of 0.0065, see Figure 16 in the Appendix.

Cluster-based Local Linear Regression Model

The last model explored was a local linear regression model based on the clusters returned by the K-Means algorithm. To decide the number of clusters to use an 'elbow' graph was drawn and the 'kink' in the graph had to identify. This graph that can be seen in Figure 17 in the appendix, appears to show a ‘kink’ at 4 clusters, therefore that was the number of clusters used. The K-Means algorithm was then used to produce clusters and the output can be seen in Figure 18 in the appendix. The clusters seem to be well separated with nearly no overlapping.

A linear regression model was built and used for each of those clusters, of which the output can be seen in Figure 19 in the Appendix. There is a big variability in the r^2 returned from as low as 0.0274 to as high as 0.752, and with a mean of 0.351. This shows that this method can perform as well as the best performing linear regressor described previously but not consistently as shown by the mean. The great variability also makes this model unreliable. This can be explained by the graph in Figure 18, as it shows the spread between the points in some clusters that can produce poor results.

Conclusion

Different methods have been used to clean the data and then build a Linear Regression model. The best performing one uses imputation of the missing values by predicting them using a Linear Regression model and then uses all five numerical columns as features used to predict the log_price. This had a 1704.26% increase in performance from the initial linear regression model. It must be noted that RFE for feature selection couldn’t be used when converting categorical variables to numeric due to its great computational cost, and it could have provided even better results. To conclude, even with a great number of missing values it was possible to build a decent performing model using linear regression.

Appendix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	Manhattan Rolling Sales File. All Sales From August 2012 - August 2013.																	
2	Sales File as of 08/30/2013 Coop Sales Files as of 09/18/2013																	
3	Neighborhood Name 09/06/13, Descriptive Data is as of 06/01/13																	
4	Building Class Category is based on Building Class at Time of Sale.																	
5	BOROUGH	NEIGHBORH	BUILDING CL	TAX CLASS A	BLOCK	LOT	EASE-MENT	BUILDING CL	ADDRESS	APART MENT NUMBER	ZIP CODE	RESIDENTIAL	COMMERCIA	TOTAL UNITS	LAND SQUAI	GROSS SQU/	YEAR BUILT	TAX C
6	1		13 CONDOS		738	1306			345 WEST 1		10014	0	0	0	0	0	0	
7	1		13 CONDOS		738	1307			345 WEST 1		10014	0	0	0	0	0	0	
8	1		13 CONDOS		738	1308			345 WEST 1		10014	0	0	0	0	0	0	
9	1		13 CONDOS		738	1309			345 WEST 1		10014	0	0	0	0	0	0	
10	1		13 CONDOS		738	1310			345 WEST 1		10014	0	0	0	0	0	0	
11	1		13 CONDOS		738	1311			345 WEST 1		10014	0	0	0	0	0	0	
12	1		13 CONDOS		738	1312			345 WEST 1		10014	0	0	0	0	0	0	
13	1		13 CONDOS		738	1314			345 WEST 1		10014	0	0	0	0	0	0	
14	1		13 CONDOS		738	1317			345 WEST 1		10014	0	0	0	0	0	0	
15	1		13 CONDOS		738	1318			345 WEST 1		10014	0	0	0	0	0	0	
16	1		13 CONDOS		738	1319			345 WEST 1		10014	0	0	0	0	0	0	
17	1		13 CONDOS		738	1320			345 WEST 1		10014	0	0	0	0	0	0	
18	1		13 CONDOS		738	1323			345 WEST 1		10014	0	0	0	0	0	0	
19	1		13 CONDOS		738	1324			345 WEST 1		10014	0	0	0	0	0	0	
20	1		13 CONDOS		738	1325			345 WEST 1		10014	0	0	0	0	0	0	
21	1		13 CONDOS		738	1328			345 WEST 1		10014	0	0	0	0	0	0	
22	1		13 CONDOS		738	1333			345 WEST 1		10014	0	0	0	0	0	0	
23	1	ALPHABET CI		4	384	1401		RK	229 EAST 2N	1A	10009	0	0	1	0	0	2008	
24	1	ALPHABET CI	03 THREE F/	1	377	66		C0	243 EAST 7T		10009	3	0	3	2,381	3,084	1899	
25	1	ALPHABET CI	04 TAX CLAS	1C	399	1102		R6	238 EAST 4T	-	10009	1	0	1	0	0	1955	
26	1	ALPHABET CI	07 RENTALS	2B	374	1		C7	303 EAST 4T		10009	8	2	10	1,501	6,929	1900	
27	1	ALPHABET CI	07 RENTALS	2	375	62		C4	715 EAST 5T		10009	20	0	20	2,426	9,345	1900	
28	1	ALPHABET CI	07 RENTALS	2	376	30		C4	274 EAST SE		10009	13	0	13	2,726	13,002	1910	

Figure 1: Snapshot of the initial Manhattan12.csv file

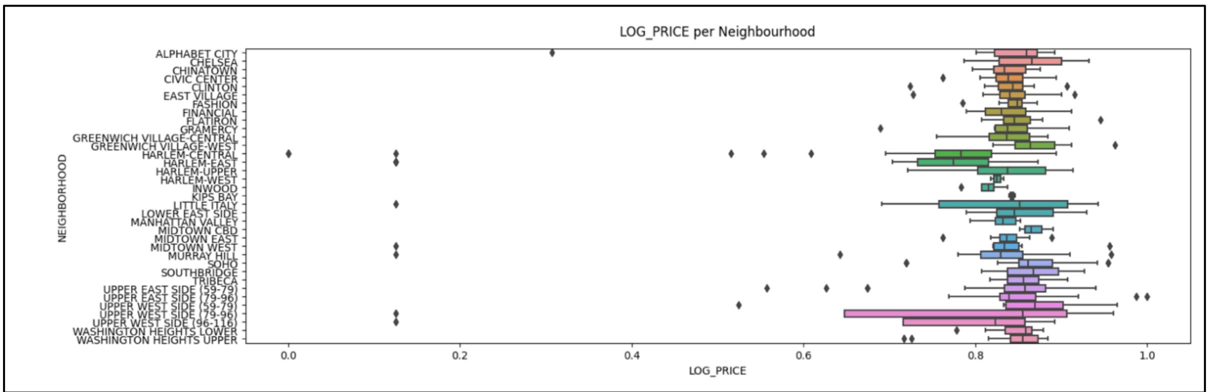

Figure 2: Shows boxplots of LOG_PRICE per NEIGHBORHOOD

Figure 3: Shows the relationship between LOG_PRICE and SALE DATE

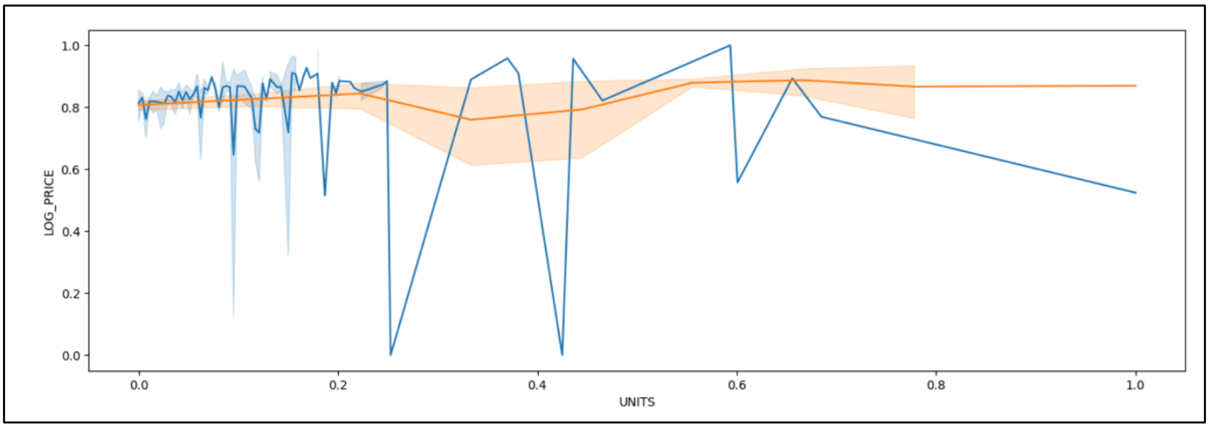

Figure 4: Shows the relationship between RESIDENTIAL UNITS (blue) and COMMERCIAL UNITS (orange) with LOG_PRICE.

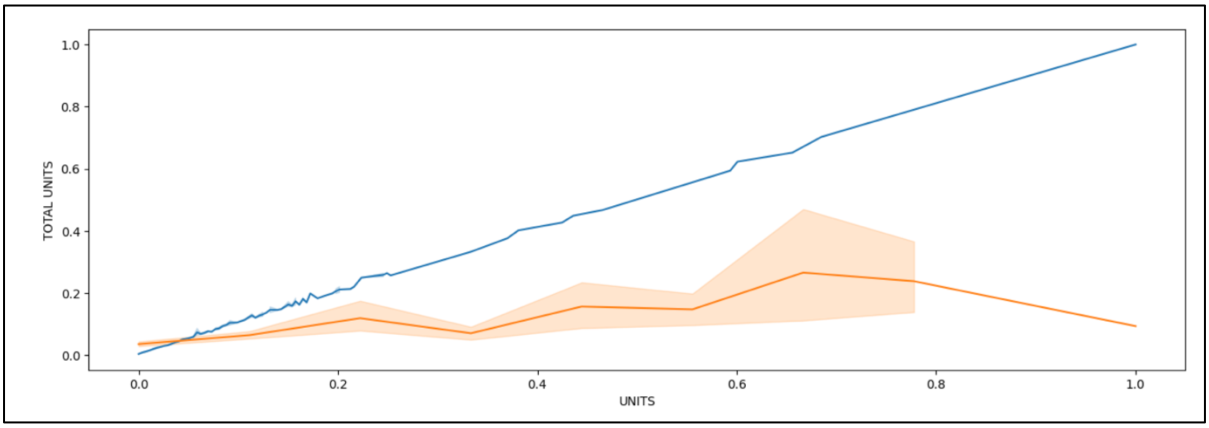

Figure 5: Shows relationship of RESIDENTIAL UNITS (blue) and COMMERCIAL UNITS (orange) with the TOTAL UNITS.

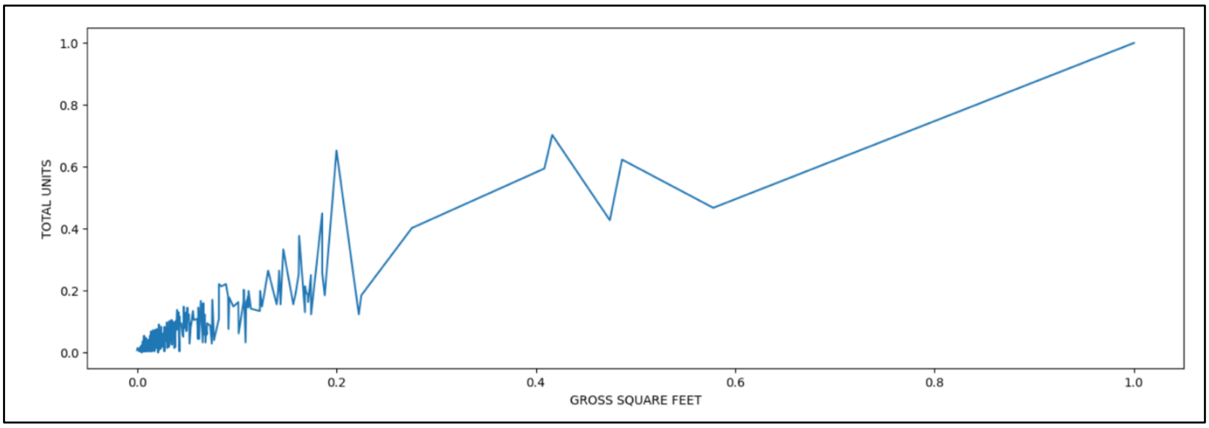

Figure 6: Shows relationship between GROSS SQUARE FEET and TOTAL UNITS.

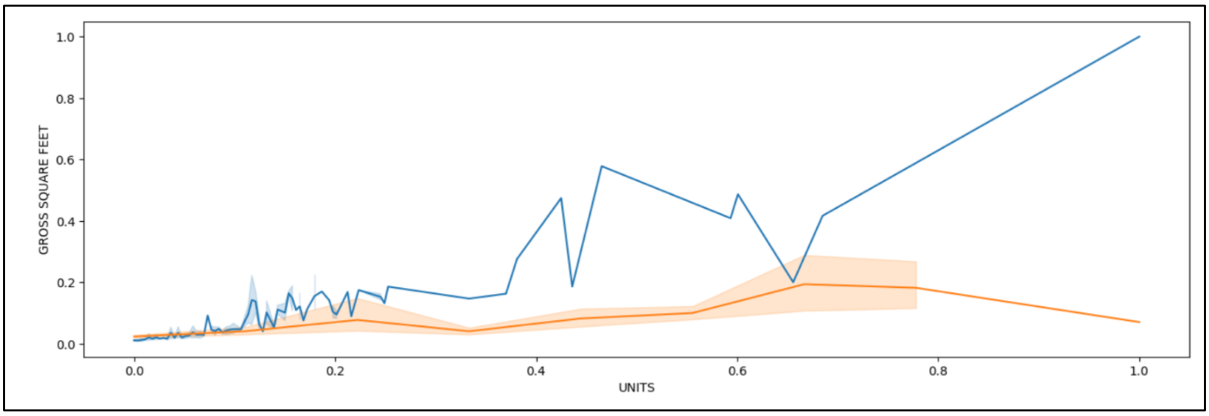

Figure 7: Shows the relationship between RESIDENTIAL UNITS and GROSS SQUARE FEET.

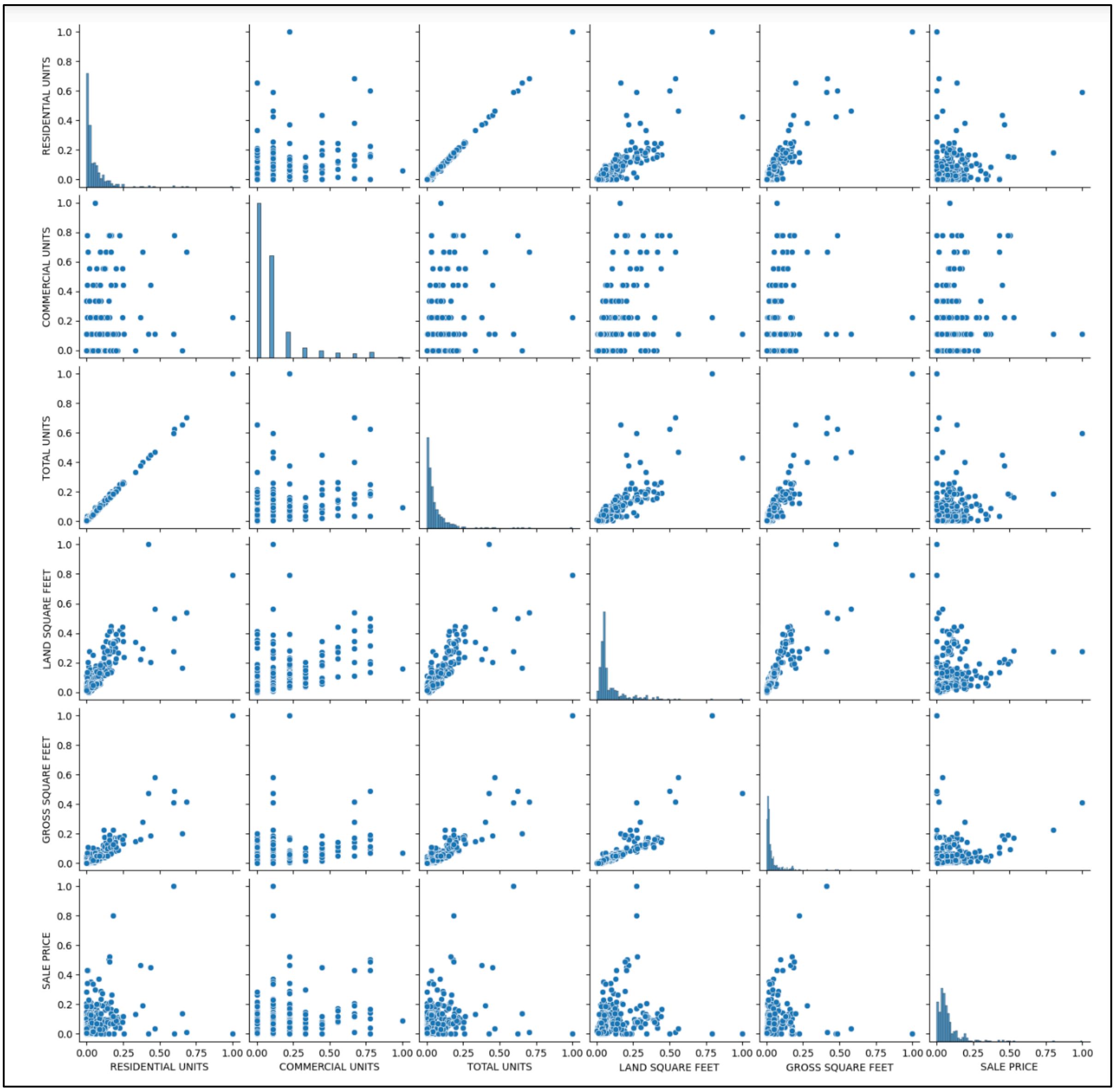

Figure 8: Shows a scatter plot matrix of the numerical variables.

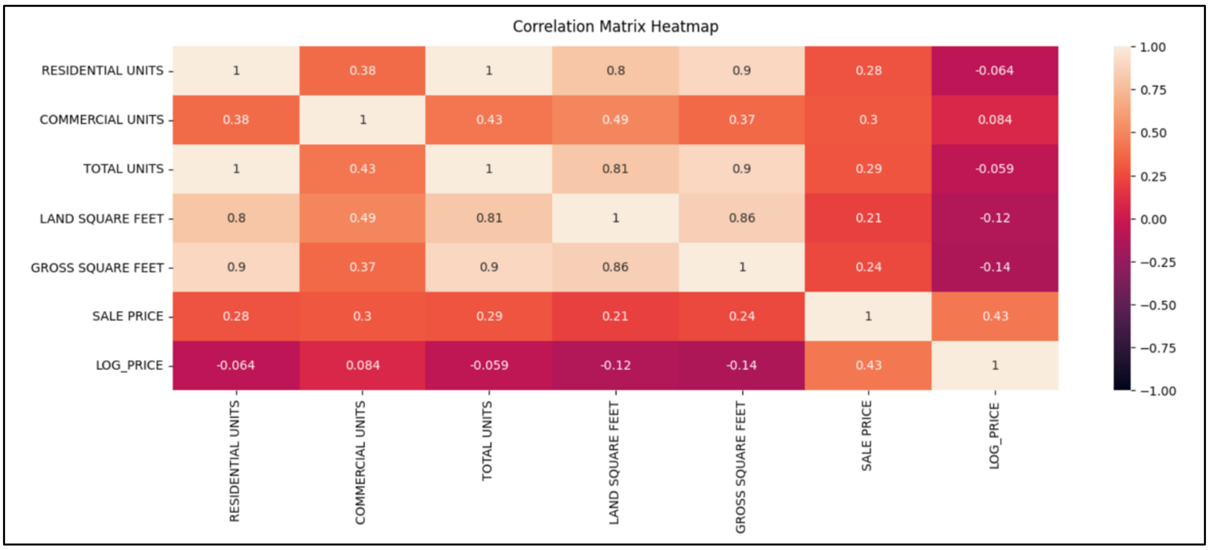

Figure 9: Shows the correlation matrix heatmap for the numerical variables.

```
Y-axis intercept 0.8221

Coefficients:
    RESIDENTIAL UNITS: 0.3910
     COMMERCIAL UNITS: 0.1478
     LAND SQUARE FEET: -0.1742
    GROSS SQUARE FEET: -0.6333

R squared for the training data: 0.065

R squared for the test data: 0.053
Mean square error (MSE): 0.013956476061151708

Cross Validation r2 score: -0.0469
DONE
```

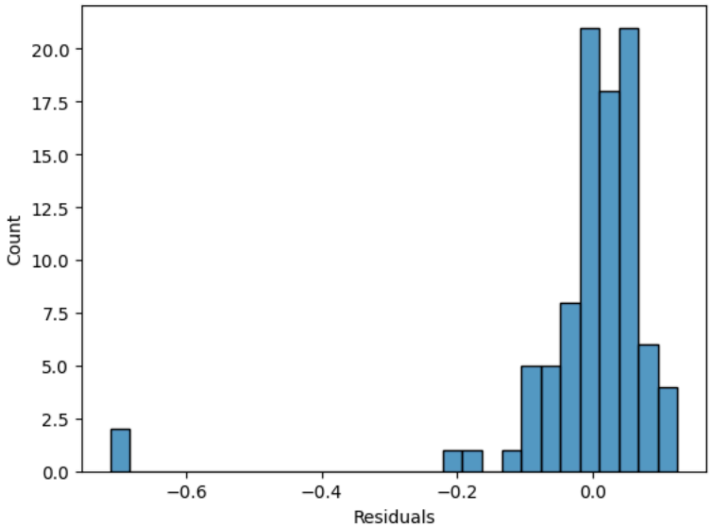

Figure 10: A screen capture of the output of the initial linear regression model.

```
Creating model...
These are the features used to predict:
['RESIDENTIAL UNITS', 'TOTAL UNITS', 'LAND SQUARE FEET', 'GROSS SQUARE FEET']
DONE

Y-axis intercept 0.8728

Coefficients:
    RESIDENTIAL UNITS: 2.0237
          TOTAL UNITS: -1.8690
     LAND SQUARE FEET: 0.7707
    GROSS SQUARE FEET: -1.8882

R squared for the training data: 0.708

R squared for the test data: 0.674

Mean square error (MSE): 0.004103750128621069

Cross Validation r2 score: 0.6983

Out[53]: 'DONE'
```

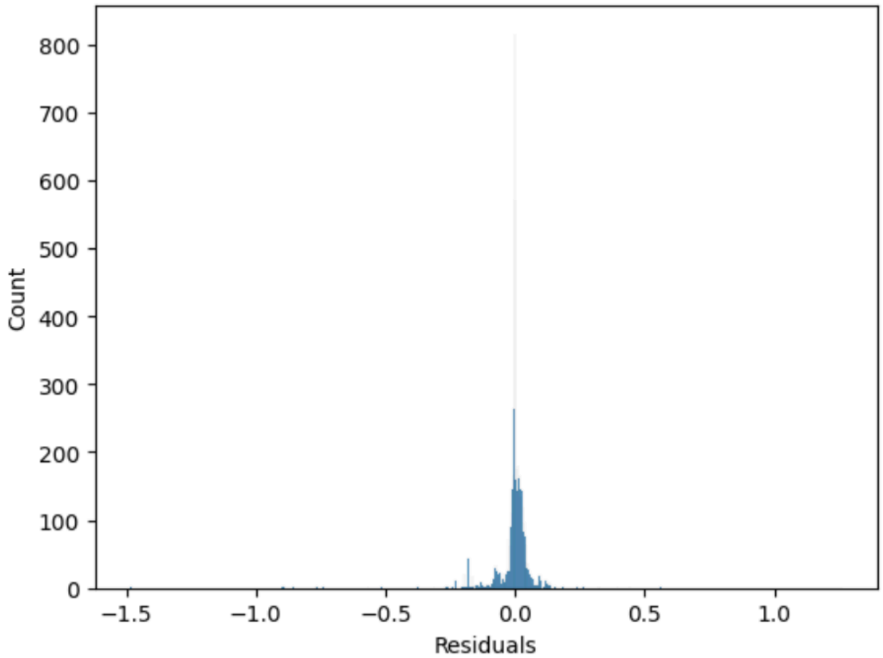

Figure 11: A screen capture of the output of the improved linear regression model when using RFE for feature selection in the improved linear model.

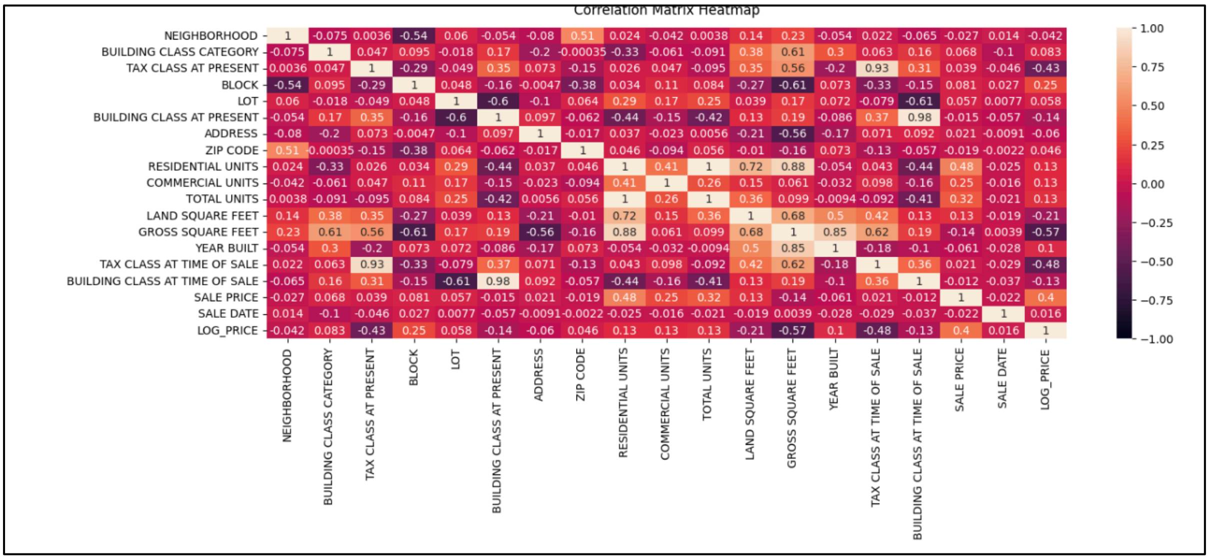

Figure 12: Correlation matrix heatmap for the imputed dataframe with categorical variables turned into numeric.

```
Y-axis intercept 0.8069

Coefficients:
 TAX CLASS AT PRESENT: 0.0022
     LAND SQUARE FEET: 0.6595
    GROSS SQUARE FEET: -0.3946
 TAX CLASS AT TIME OF SALE: -0.1271
                BLOCK: 0.0000

R squared for the training data: 0.417

R squared for the test data: 0.400

Mean square error (MSE): 0.0074908727238997535

Cross Validation r2 score: 0.4127
```

Out[56]: 'DONE'

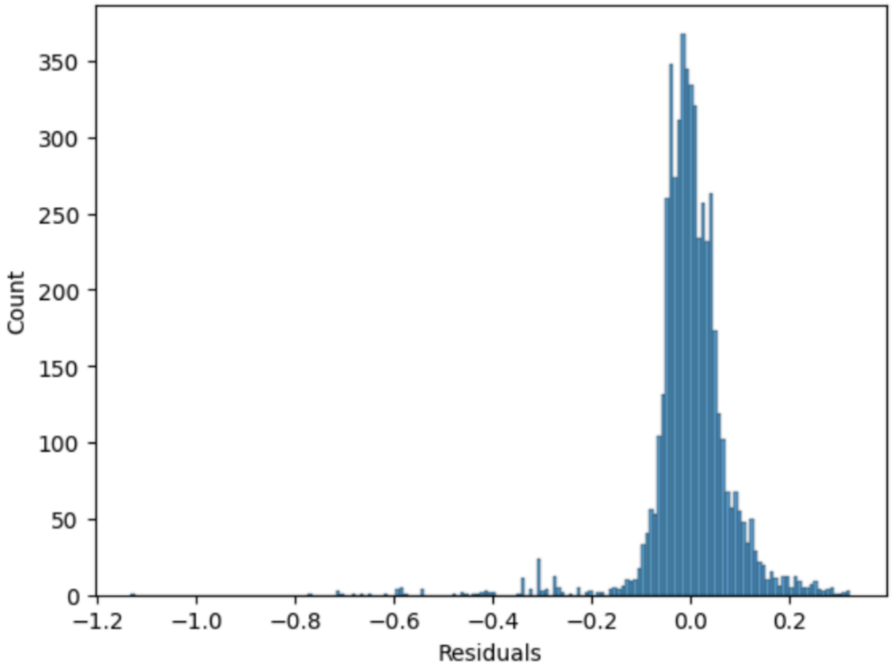

Figure 13: A screen capture of the output of the improved linear regression model when personally selecting the features to train the model.

```
Creating model...
Y-axis intercept 0.7758

Coefficients:
         NEIGHBORHOOD: 0.0020
 BUILDING CLASS CATEGORY: 0.0234
 TAX CLASS AT PRESENT: 0.0164
                BLOCK: -0.0001
                  LOT: 0.0000
 BUILDING CLASS AT PRESENT: 0.0007
              ADDRESS: -0.0000
             ZIP CODE: -0.0042
    RESIDENTIAL UNITS: -66.2765
     COMMERCIAL UNITS: -3.3786
          TOTAL UNITS: 65.4056
     LAND SQUARE FEET: 1.1526
    GROSS SQUARE FEET: 0.4812
           YEAR BUILT: 0.0006
 TAX CLASS AT TIME OF SALE: -0.1987
 BUILDING CLASS AT TIME OF SALE: -0.0028
            SALE DATE: 0.0001

R squared for the training data: 0.652

R squared for the test data: 0.659

Mean square error (MSE): 0.004229592591948757

Cross Validation r2 score: 0.6510

Out[9]: 'DONE'
```

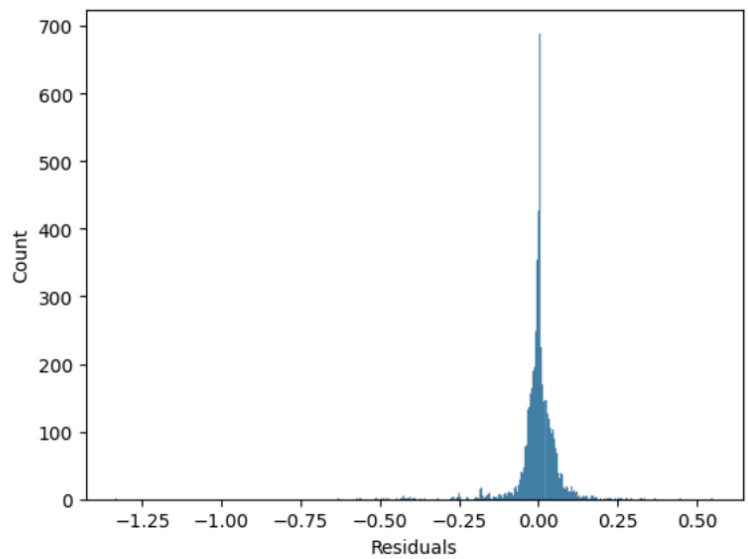

Figure 14: A screen capture of the output of the improved linear regression model when using all the features available.

```
Creating model...
These are the features used to predict:
['RESIDENTIAL UNITS', 'COMMERCIAL UNITS', 'TOTAL UNITS', 'LAND SQUARE FEET', 'GROSS SQUARE FEET']
DONE

Y-axis intercept 0.6409

Coefficients:
    RESIDENTIAL UNITS: -64.5722
     COMMERCIAL UNITS: -3.4850
          TOTAL UNITS: 63.8097
     LAND SQUARE FEET: 0.7824
    GROSS SQUARE FEET: 0.4449

R squared for the training data: 0.745

R squared for the test data: 0.796

Mean square error (MSE): 0.0025139959106576463

Cross Validation r2 score: 0.7541
```

```
Out[16]: 'DONE'
```

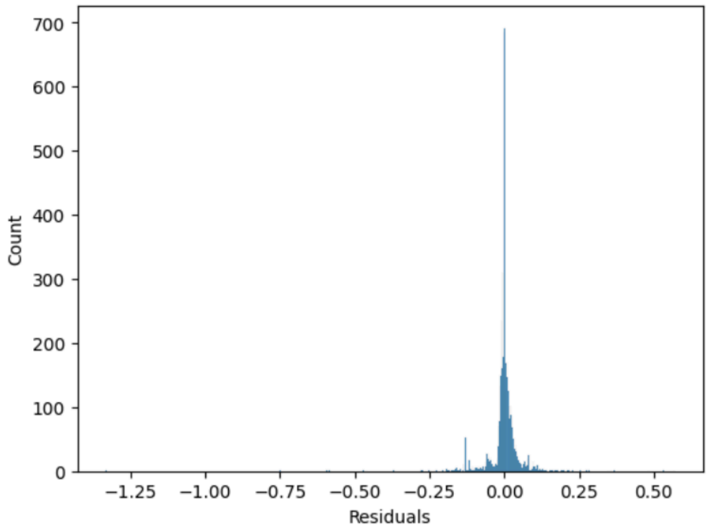

Figure 15: A screen capture of the output of the improved linear regression model when using all the numerical columns.

```
Creating model...
Y-axis intercept 1.1746

Coefficients:
    RESIDENTIAL UNITS: 93.8682
     COMMERCIAL UNITS: 4.7573
          TOTAL UNITS: -92.5699
     LAND SQUARE FEET: 0.7945
    GROSS SQUARE FEET: -4.8742

R squared for the training data: 0.552

R squared for the test data: 0.558

Mean square error (MSE): 0.0064517448847151255

Cross Validation r2 score: 0.5252

[25]: 'DONE'
```

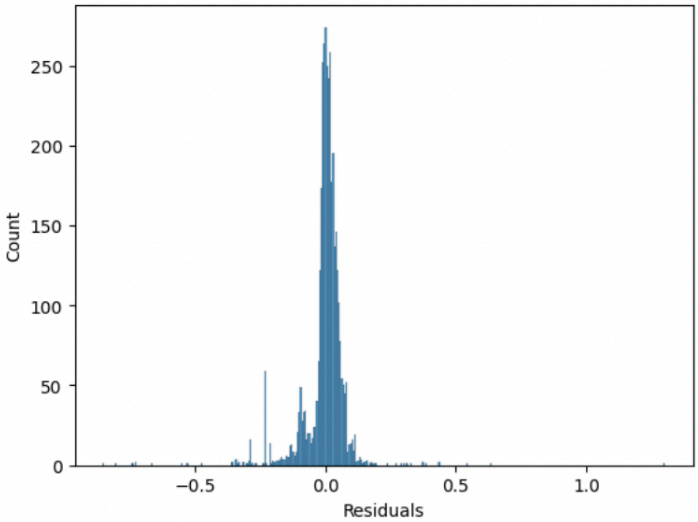

Figure 16: A model built without predicting SALE PRICE or LOG_PRICE

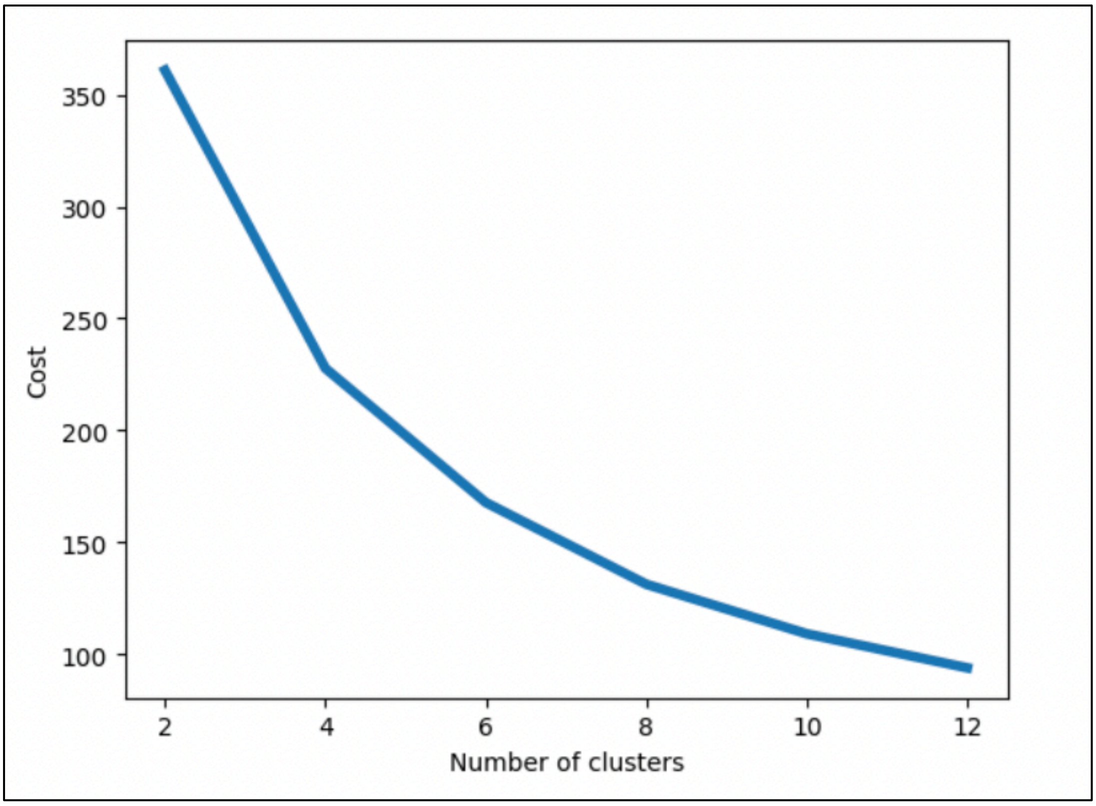

Figure 17: Elbow graph

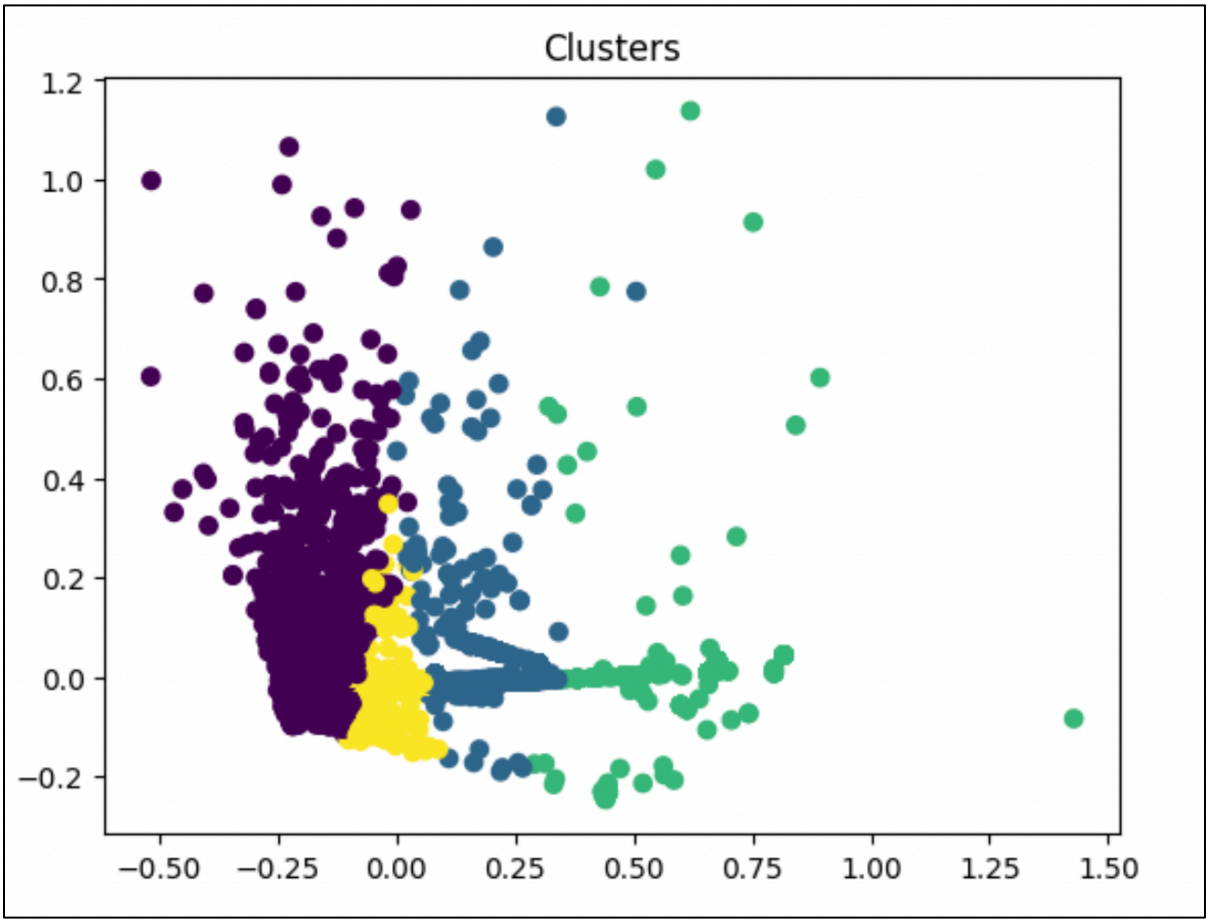

Figure 18: Clusters produced by the K-Means algorithm.

```
R squared value for cluster 0 using cross-validation: 0.4150502283503271
R squared value for cluster 1 using cross-validation: 0.2107788184654248
R squared value for cluster 2 using cross-validation: 0.7516527521015863
R squared value for cluster 3 using cross-validation: 0.02742152872042028

Mean of r squared values for local linear regressors: 0.3512258319094396

Minimum and maximum r squared value are: 0.02742152872042028 & 0.7516527521015863
```

Figure 19: Output of the cluster-based local linear regressor.