Predicting Housing Prices in NYC

Helen Ma (hm385), Shane Racey (str45), Ivan Xie (ix24)

Orie 4741: Big Messy Data

Final Project

May 10th, 2024

Table of Contents

Abstract

New York City is home to one of the most dynamic and competitive housing markets in the United States, with hundreds of sales and property rentals being listed daily. Our dataset is taken from real estate company StreetEasy's Data Dashboard, encompassing the history of listing prices from 2010 to the present across metrics from median asking price to inventory. This project serves to leverage trends in the housing market in developing robust regression models, ultimately encouraging our clients in the real estate industry to predict future prices of various property types within NYC and iteratively optimize their pricing strategies.

1. Data Quality Analysis

1.1 Raw Data Overview

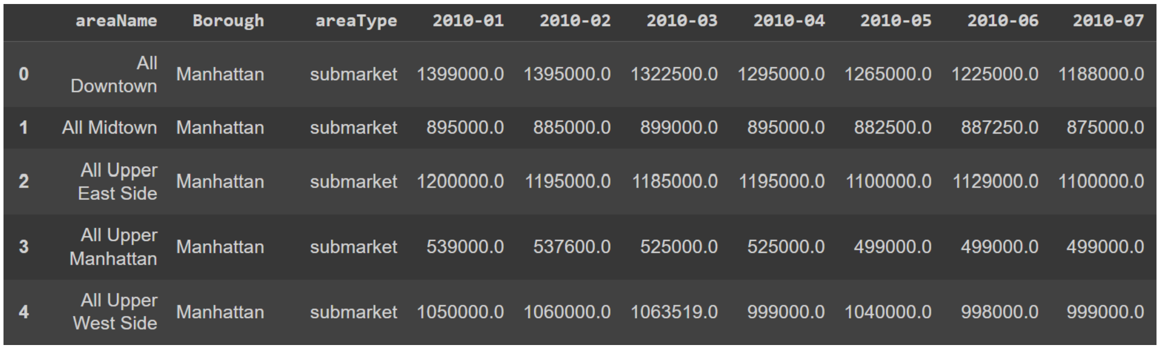
The folder consists of multiple CSV files, each containing monthly information over time for various property details. The data is collected from January 2010 to February 2024. Prices lower than $10,000 are not included in the data. Records on home sales transactions come from the New York City Department of Finance. We initially chose to focus on the following datasets listed below with their definitions:

- Median Asking Price: the exact middle asking price among all asking prices of available listed homes during the month
- Median Asking Rent: the exact middle asking rent among all asking rental prices of available listed homes during the month
- Median Sales Price: the exact middle sales price among all recorded sales prices of homes that closed during the month
- Recorded Sales Volume: the number of properties sold in that month
- Days On Market: the number of days that the property has been listed before it was sold
- Sale List Ratio: the ratio of the sale price to list price

We wish to predict how a real estate group should price their property given the physical characteristics of the property such as location and size, along with the previous price trends of a typical property in that location. In each dataset, there are 198 rows and 173 columns. Each row is indexed by a unique area name. Columns labeled "Borough" and "areaType" categorize the area name into broader region descriptions. The rest of the columns are numerical prices, sales volume, number of days, or ratios, by date, according to the title of the respective data set.

1.2 Data Visualization

To get a better understanding of the datasets that will be used for predictive modeling, visualizations will help give a general sense of what prices we should expect. Looking at the median asking price dataset, we can see that each row contains information about the location of the property and a history of the price of the property for each given month. We noticed that there are some values that aggregate certain areas of NYC like ‘All Downtown’ and ‘All Upper West Side’ and after making more visualizations we ran into more grouped areas like all of NYC. These visualizations help us find outliers in the corrupted data so that we can get better results when modeling the data.



	areaName	Borough	areaType	2010-01	2010-02	2010-03	2010-04	2010-05	2010-06	2010-07
0	All Downtown	Manhattan	submarket	1399000.0	1395000.0	1322500.0	1295000.0	1265000.0	1225000.0	1188000.0
1	All Midtown	Manhattan	submarket	895000.0	885000.0	899000.0	895000.0	882500.0	887250.0	875000.0
2	All Upper East Side	Manhattan	submarket	1200000.0	1195000.0	1185000.0	1195000.0	1100000.0	1129000.0	1100000.0
3	All Upper Manhattan	Manhattan	submarket	539000.0	537600.0	525000.0	525000.0	499000.0	499000.0	499000.0
4	All Upper West Side	Manhattan	submarket	1050000.0	1060000.0	1063519.0	999000.0	1040000.0	998000.0	999000.0

Table 1: Dataset for median asking price

The first figure is a density map to help visualize the price distribution of properties across the 5 boroughs. The results of mapping the prices aligned with our domain knowledge with Manhattan and Brooklyn having average higher prices costing around 1 million. Additionally, Queens, Bronx, and Staten Island have lower price ranges with an average sale price of 612k, 529k, and 305k respectively. The location of the sales property was a significant feature when making predictive pricing models since the area in each borough may impact the price of an apartment.

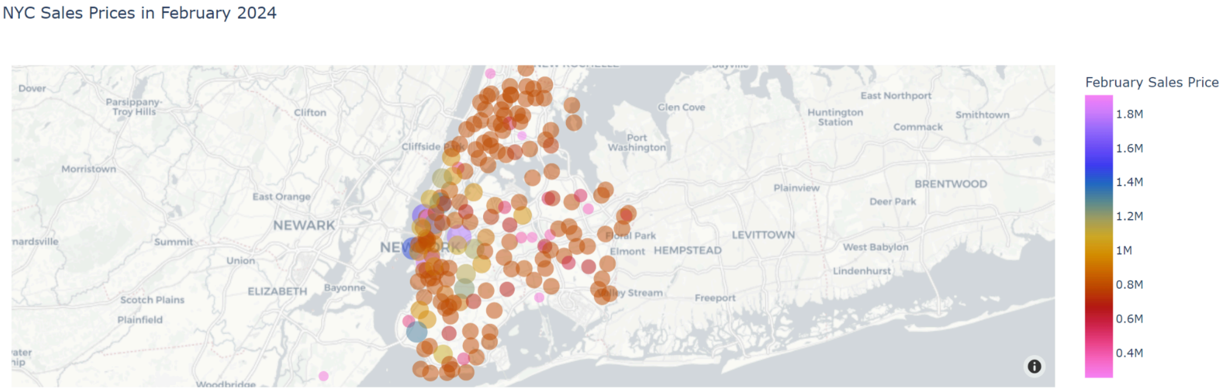
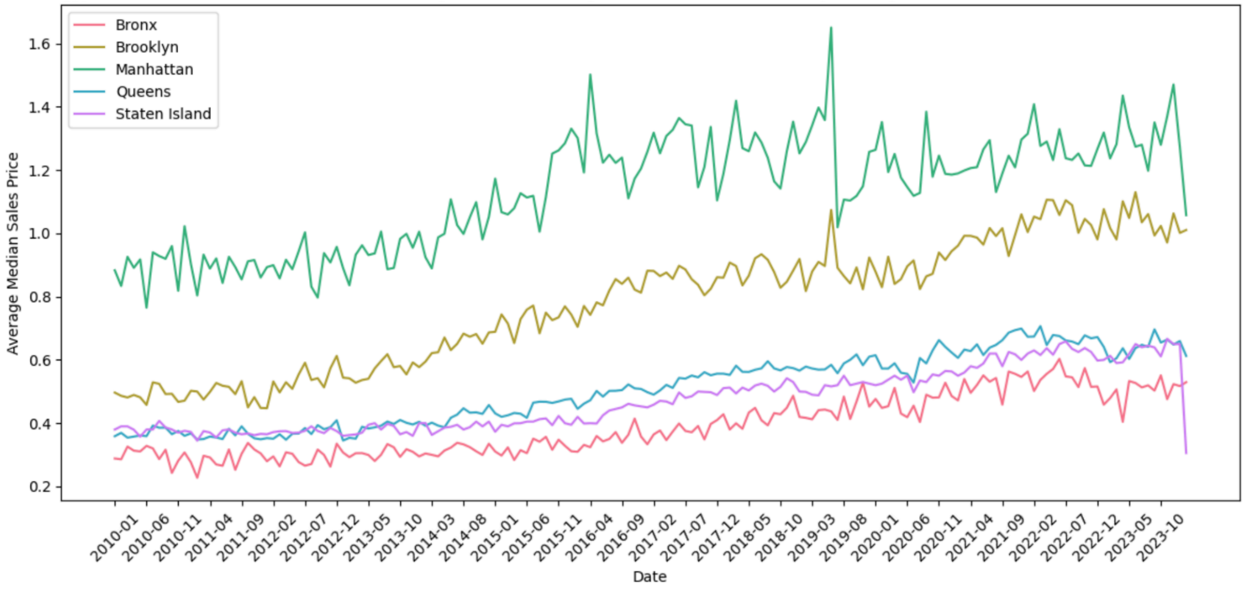


Figure 1: Density map of February sales prices

Our second plot shows the average median sales price in millions of dollars from 2010-2023 for the five different boroughs in NYC. We can see that the different boroughs vary a significant amount in monetary magnitude, so we kept the borough category as a predictor in our model analyses moving forward. There seems to be an overall increase spanning the most recent decade.



Graph 1: Change in NYC housing sales price over time by borough

1.3 Data Cleaning

Our raw datasets contain numerous NA values across both time and area name, highlighting missing entries where data is not available. Potential explanations for the presence of NAs pertain to lower-income neighborhoods in the Bronx or Brooklyn, where the greater proportion of public housing and long-term government projects detracts from opportunity for commercial real-estate. Smaller gaps in data may simply be attributed to external factors from the recording process. The numbers of missing values (NAs) across our 6 datasets are as follows: 9563 in "medianAsking Price_All .csv", 11989 in "medianAskingRent_All.csv", 12595 in "medianSalesPrice_All.csv", 1 in "recordedSales Volume_All.csv", 23654 in "daysOnMarket _ All.csv", and 25157 in "saleList Ratio_All.csv". We did not observe any zero values or any other placeholders holding no significant meaning in the context of our data.

Our data filtering first consisted of identifying nearly empty rows of NAs such that there was not enough information over time to predict neighboring or future data. In our code, clean_data(...,drop_threshold,...) takes in a user-specified proportion of NAs in a particular row (excluding id columns) such that if the row exceeds this proportion, this row is removed. In our case, we set our threshold to be 0.5. For relatively sparse rows that still satisfied our threshold constraint, we imputed the missing values for each area name by taking the median value in the nearest preceding and proceeding 2 years where data exists. Our clean_data(..., interval) also takes in a user-specified range of years centered about the "pivot" year to calculate the median around. Rather than taking the mean or median of the entire row, zoning into the 4-year range accounts for our data's temporal aspect. We affirmed that data from our earliest year in 2010 may not be representative of the housing market now in 2024, but then we assume that data close in time are similar. Median prices may also be more useful to consider than mean prices due to their robustness against price outliers, crucial in combatting volatile periods in the housing market.

1.4 Data Reorganization

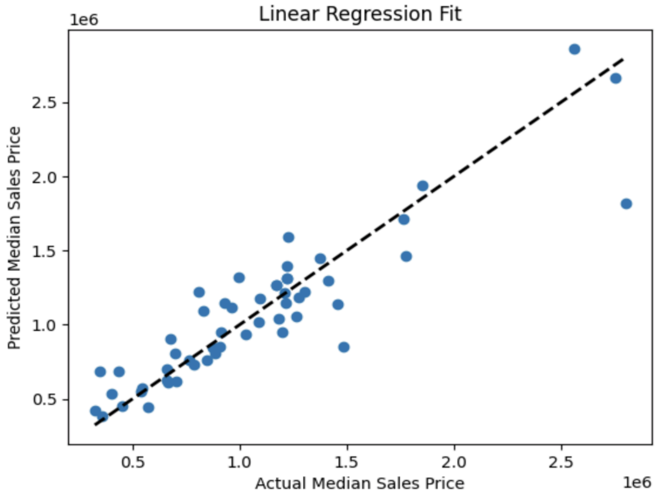
Finally, since our data is given in 6 separate datasets with values collected over a wide range of time, we wanted to combine them and look at all 6 predictors to see the effects between each, rather than seeing the trend of one predictor over time. To do this, we implemented a function that randomly generates a date, or column from the original datasets, to pull values on all 6 predictors. The resulting data frame still has one row for each "areaName", but each of the columns now contains information on one predictor, with the last column being the response, median sales price. Also, to include the borough as a numerical predictor, we mapped each of the five boroughs to the values 0-4, inclusive.

2. Model Selection

We experimented with our data on several regression models to examine which fit made reliable and valid predictions for median sales prices.

2.1 Least Squares Multiple Linear Regression

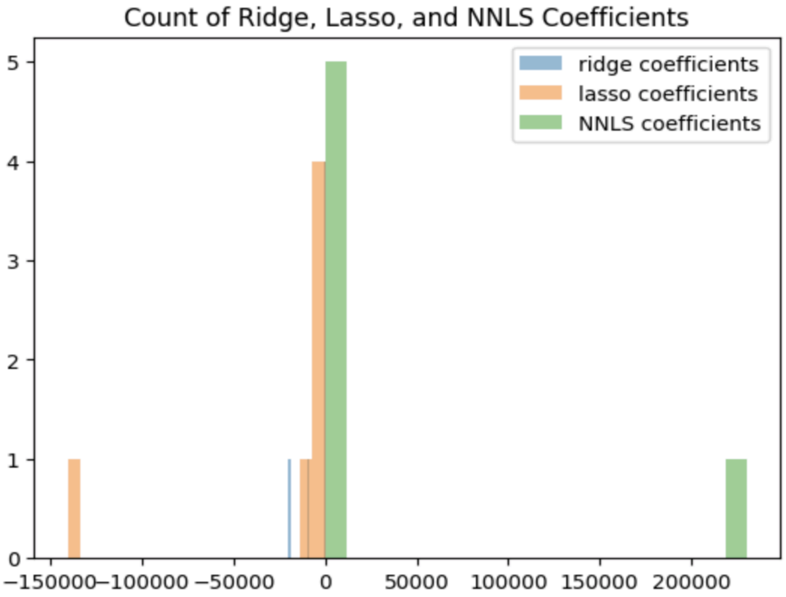
We fitted a least squares regression on the compiled data which includes 6 predictors and the response. This was done 10 times on randomly generated datasets selected from the past 10 months for relevancy purposes. From Figure 3, we could see that the predicted price is generally accurate along a wide range of prices. The R-squared value for the given plot on one dataset was 0.820, while the adjusted r-squared was 0.797. The MSE was 5.264e+09. The data seems prone to outliers, so we considered the L2 norm instead, which was 1.670e+06. We considered more models moving forward for comparison.



Graph 2: Linear regression fit for predicted vs actual sales price

2.2 Regularized Regression

To see which model might potentially fit our dataset better, we compared ridge, lasso, and nonnegative least squares regression. Ridge regression, or L2 regularization, puts weights on the coefficients. Lasso regression scales coefficients to 0, favoring the outcome of a sparse model. The nonnegative least squares regression also fits a sparse model, as it restricts all coefficients to be nonnegative. After fitting the three models, we generated a histogram to visualize the range of coefficients.



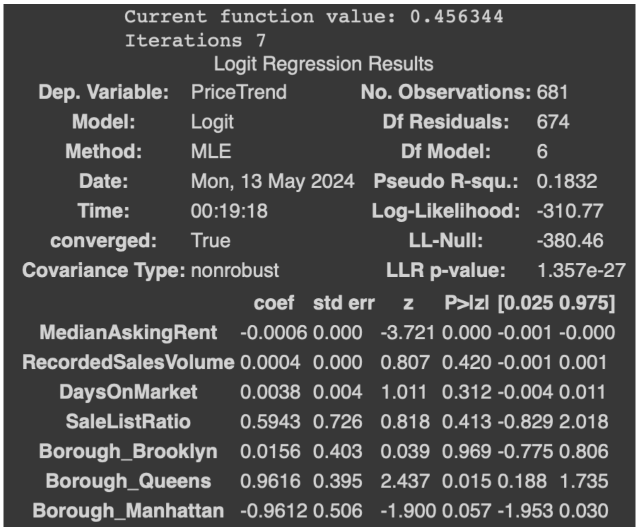
Graph 3: Histogram of coefficients for ridge, lasso, NNLS

From the plot, we can see that the nonnegative least squares model was the most sparse for our data. The coefficients were 0.752 for MedianAskingPrice, 66.470 for MedianAskingRent, 2.300e+05 for SaleListRatio, and 0 for Borough, RecordedSalesVolume, and DaysOnMarket. This means that only the first three features listed were used in the prediction. However, this model had the highest test MSE among the three models, suggesting that a sparse model may not be the best for predicting the median sales price. The model producing the best fit was the ridge regression, with a test MSE of 1.017e+011 and a L2 norm of 2.321e+06. The R-squared value was 0.565. The resulting coefficients from the ridge model were -9.829e+03 for Borough, 7.497e-01 for MedianAskingPrice, 6.817e+01 for MedianAskingRent, -2.297e+01 for RecordedSalesVolume, and -4.485e+02 for DaysOnMarket, and -2.004e+04 for SaleListRatio. Negative coefficients may be necessary for a better fit on the data if features are inversely related to the outcome. The lasso regression produced very similar coefficients with the ridge, without scaling any down to 0.

2.3 Logistic Regression

We aimed to answer the following question utilizing a logistic regression model: which areas across NYC are more “profitable” for a seller to market their property listings? Logistic regression is a form of supervised machine learning used in classification settings where the objective is to predict the probability that an outcome belongs in some binary class or not. We can think of logistic regression as an extension for linear regression. While linear regression predicts continuous real-valued outcomes, logistic regression predicts class probabilities which enables classification decisions. Logistic regression assumes that data is well-modeled by the sigmoid function $\sigma(x) = 1/(1+e^{-x})$, and that our predictors are not highly correlated.

In defining how we deemed an area to be opportunistic, we benchmarked the price at which the buyer purchased the property in that area, “medianSalesPrice”, against the lowest price at which the seller was willing to sell the property, “medianAskingPrice”. Higher median sales prices than median asking prices characterize a seller’s market, as the buyer must compensate for low supply of such property and pay some above-typical price. Otherwise, we have a buyer’s market where property prices are below typical and sellers must drive their prices down to adjust for low demand. Lower sales prices than asking prices could also indicate negotiation from the buyer’s end, where sellers set high prices at listing time but do not expect to end up realizing its entire revenue. Thus, the projections we looked to output from our model aim to answer whether the sentiment regarding real estate in a given area is generally bullish or bearish.



```
         Current function value: 0.456344
         Iterations 7
                        Logit Regression Results
   Dep. Variable:     PriceTrend          No. Observations: 681
       Model:         Logit                 Df Residuals:   674
      Method:         MLE                    Df Model:      6
       Date:          Mon, 13 May 2024   Pseudo R-squ.:     0.1832
       Time:          00:19:18            Log-Likelihood:   -310.77
    converged:        True                  LL-Null:        -380.46
 Covariance Type:     nonrobust             LLR p-value:    1.357e-27
                        coef    std err   z     P>|z|  [0.025  0.975]
  MedianAskingRent    -0.0006  0.000   -3.721  0.000  -0.001  -0.000
 RecordedSalesVolume   0.0004  0.000    0.807  0.420  -0.001   0.001
    DaysOnMarket       0.0038  0.004    1.011  0.312  -0.004   0.011
    SaleListRatio      0.5943  0.726    0.818  0.413  -0.829   2.018
  Borough_Brooklyn     0.0156  0.403    0.039  0.969  -0.775   0.806
   Borough_Queens      0.9616  0.395    2.437  0.015   0.188   1.735
 Borough_Manhattan    -0.9612  0.506   -1.900  0.057  -1.953   0.030
```

Table 2: Logistic regression results

We observed that on our test set across months of the last year (February 2023 - 2024), our model returned an accuracy of 0.8488 and at face value, logistic regression seemed to perform reasonably well. However, upon further inspection, we saw in our confusion matrix that skew may exist when comparing our true positive (the event where our model correctly predicts that a property was sold at a higher price than its ask) and true negative rates (lower price than its ask). Our model's precision on "Above Ask" was 0.76, while that of "Below Ask" was 0.86, suggesting that our model was better at simply identifying unprofitable areas for realtors to steer away from rather than pinpointing areas of unusual opportunity, i.e. "diamonds in the rough". The model was bullish about future real estate in Queens, bearish about Manhattan, and neutral about Brooklyn. It was surprising to find that the model does not have a strong sentiment regarding the DaysOnMarket predictor, as one would expect that longer times for a property to stay on the market indicate that either buyers do not perceive this property to be valuable or there is not a strong demand for properties from this area in general. The pseudo-R-squared value of 0.1832 indicated that 18.32% of the variance in the probability of a bullish prediction was explained by the model. Overall, the logistic model gave sensible predictions with respect to its predictors, but exhibited bias when it came to its success with "Below Ask" predictions due to the abundance of these labels in our raw datasets.

2.4 Random Forest Regression

The next model we looked at was a random forest regression. Random forest is an ensemble learning method predominantly used for classification and regression tasks. It operates by constructing a multitude of decision trees during the training phase and outputting the mean or average prediction of the individual trees for regression tasks. Another reason for choosing this model is because it is a form of bootstrap aggregating (bagging) of decision trees, where each tree is built from a sample drawn with replacement from the training set. Since we had a limited number of features from our dataset, this bootstrap aggregation helped with adding more data into the algorithm to make our model more robust. In the random forest model that we created, we had 20 trees in our forest with a maximum depth of 10. Below is a figure of one of the trees that was produced from our forest selecting between the different features. Because decision trees tend to keep splitting until there is zero training error, we did not have to worry about underfitting our data. Additionally, since we only had 6 features from our dataset and limited the maximum depth of each tree, it was hard for the decision tree to overfit the data.

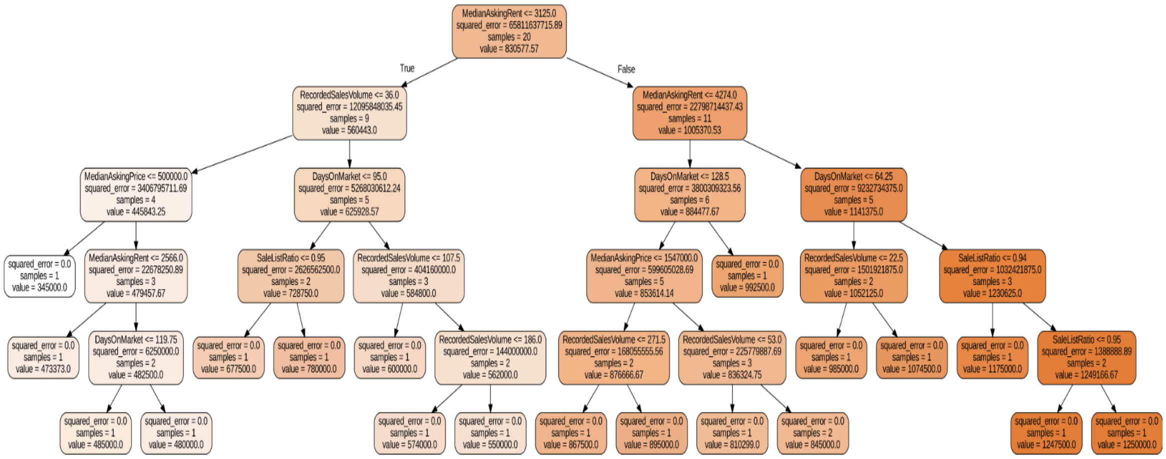

MedianAskingRent <= 3125.0 squared_error = 65811637715.89 samples = 20 value = 830577.57
True
False
RecordedSalesVolume <= 36.0 squared_error = 12095848035.45 samples = 9 value = 560443.0
MedianAskingRent <= 4274.0 squared_error = 22798714437.43 samples = 11 value = 1005370.53
MedianAskingPrice <= 500000.0 squared_error = 3406795711.69 samples = 4 value = 445843.25
DaysOnMarket <= 95.0 squared_error = 5268030612.24 samples = 5 value = 625928.57
DaysOnMarket <= 128.5 squared_error = 3800309323.56 samples = 6 value = 884477.67
DaysOnMarket <= 64.25 squared_error = 9232734375.0 samples = 5 value = 1141375.0
squared_error = 0.0 samples = 1 value = 345000.0
MedianAskingRent <= 2566.0 squared_error = 22678250.89 samples = 3 value = 479457.67
SaleListRatio <= 0.95 squared_error = 2626562500.0 samples = 2 value = 728750.0
RecordedSalesVolume <= 107.5 squared_error = 404160000.0 samples = 3 value = 584800.0
MedianAskingPrice <= 1547000.0 squared_error = 599605028.69 samples = 5 value = 853614.14
squared_error = 0.0 samples = 1 value = 992500.0
RecordedSalesVolume <= 22.5 squared_error = 1501921875.0 samples = 2 value = 1052125.0
SaleListRatio <= 0.94 squared_error = 1032421875.0 samples = 3 value = 1230625.0
squared_error = 0.0 samples = 1 value = 473373.0
DaysOnMarket <= 119.75 squared_error = 6250000.0 samples = 2 value = 482500.0
squared_error = 0.0 samples = 1 value = 677500.0
squared_error = 0.0 samples = 1 value = 780000.0
squared_error = 0.0 samples = 1 value = 600000.0
RecordedSalesVolume <= 186.0 squared_error = 144000000.0 samples = 2 value = 562000.0
RecordedSalesVolume <= 271.5 squared_error = 168055555.56 samples = 2 value = 876666.67
RecordedSalesVolume <= 53.0 squared_error = 225779887.69 samples = 3 value = 836324.75
squared_error = 0.0 samples = 1 value = 985000.0
squared_error = 0.0 samples = 1 value = 1074500.0
squared_error = 0.0 samples = 1 value = 1175000.0
SaleListRatio <= 0.95 squared_error = 1388888.89 samples = 2 value = 1249166.67
squared_error = 0.0 samples = 1 value = 485000.0
squared_error = 0.0 samples = 1 value = 480000.0
squared_error = 0.0 samples = 1 value = 574000.0
squared_error = 0.0 samples = 1 value = 550000.0
squared_error = 0.0 samples = 1 value = 867500.0
squared_error = 0.0 samples = 1 value = 895000.0
squared_error = 0.0 samples = 1 value = 810299.0
squared_error = 0.0 samples = 2 value = 845000.0
squared_error = 0.0 samples = 1 value = 1247500.0
squared_error = 0.0 samples = 1 value = 1250000.0

Figure 2: Random forest regression tree

To measure the results of our data, we made sure to create a test/train split so that we only trained the model on 70% of all data. The other 30% of our data was used to test the predictions of our random forest model against and make evaluations to see if it was a good fit or not. From our results, we obtained a mean squared error (MSE) of 87821590355, a mean absolute error (MAE) of 230267, and an R-squared (R2) score of 0.53.

Unfortunately, the random forest model did not perform as well as we thought it would have. One reason why this may not have been a good fit is because of the lack of features from our dataset. Random forests excel when there are lots of features to choose from and can aggregate multiple decision trees based on the selected features in each tree. Since we only had a limited number of features, each tree may not have been as unique as the others.

Conclusion

The models we chose are commonly used throughout industry and academia with the least squares multiple regression being the best. However, over recent years, these models have been adapted and upgraded to be more precise and accurate depending on the dataset and industry. For this reason, we would like to explore deeper models before using them in production to change how we would make decisions for a company.

After analyzing our project we have concluded that we have not created a weapon of math destruction. The outcomes of the models are simple to measure such that it is clear if a model is a good fit or not. The predictions made from the models are not likely to hurt anyone whether the predictions are accurate or not. If the predictions from the models are used to set prices for listings, it would simply cause some confusion about what the property would actually be priced at since there are multiple checkpoints before a property is sold or rented.

In terms of fairness, we only considered price and location so that there is virtually no room for bias or discrimination. Since our data only deal with features of property listings, the models are fair in price evaluation.

Appendix

Contributions

1) Shane: density map visualization, random forests model

2) Helen: line graph visualization, linear regression, regularized regression

3) Ivan: data cleaning and aggregation, logistic regression

References

[1]https://streeteasy.com/blog/data-dashboard/?agg=Total&metric=Inventory&type=Sales&bedrooms=Any%20Bedrooms&property=Any%20Property%20Type&minDate=2010-01-01&maxDate=2024-02-01&area=Flatiron,Brooklyn%20Heights

[2]https://github.com/Helen029/Orie4741Project