

NYC Condominium Building Price Estimation And Neighborhood Classification

Introduction

Price prediction and classification based on common characteristics are classic statistical and machine learning problems. This project examines these problems in the context of gaining an understanding of the structure of condominium prices in New York City in addition to developing an alternative neighborhood classification system for buildings based on their characteristics. This is accomplished by applying a series of machine learning algorithms to the NYC Open Data set.

The first goal of predicting prices was approached by utilizing a Kernel Least Square Regression model with cross validation on a training set to develop parameter estimates for different factors. These factors could then be used to calculate a predicted price for any given building. The second goal of classifying buildings into neighborhoods based on common characteristics was approached by applying a K-Nearest Neighbors algorithm to the dataset.

The results of applying the two machine learning algorithms were then used to construct two maps. The first map expresses the results of the price prediction algorithm by placing green dots on the geo-coordinate location of a building if the predicted price is significantly greater than the actual value, and a red dot if the predicted price is significantly less than the actual value. The second map expresses the results of the neighborhood classification algorithm by displaying the buildings in the dataset in different colors based on the neighborhood assigned to them by the algorithm.

Data Set

The data set utilized to train and test the algorithms were obtained from the NYC Open Data set. Specifically, the Department of Finance's Condominium Comparable Rental Income (Manhattan) data sets from the years 2008 - 2009, 2009 - 2010, 2010 - 2011, and 2011 - 2012 were chosen. The initial data set contained 36 separate features for each building. In order to reduce the dimensionality of the problem, the number of features was trimmed to nine. The nine features retained for the new subset included, for each building, the neighborhood, building classification, total units in the building, year built, gross square footage, gross income, gross income per square foot, full market value, and full market value per square foot. In addition, due to some neighborhoods containing only a few data

points, the decision was made to choose only the ten largest neighborhood sets for the final data set. This decision was made because the small sample size for the smaller neighborhoods would not provide sufficient data for training the algorithm effectively. Finally, the neighborhood and building classification features were mapped from strings to integers. This was done to simplify the process of applying the algorithm to the data.

Furthermore, in order to account for the effects of price growth between the years, the data sets needed to be normalized. A normalizing factor was applied to the market value per square foot, full market value, gross income, and gross income per square foot. In order to calculate the normalizing factor, the 2011-2012 year was selected as the baseline. That is, the 2008 – 2010 data sets were price adjusted to match the scale of the 2011 – 2012 year. In addition, to account for the differences in growth rates within the different neighborhoods of New York City, a separate normalizing factor was calculated for each neighborhood in New York City. For each neighborhood, identifying all buildings that appeared in the data sets of adjacent years and calculating the average price increase for those buildings enabled the determination of the year-to-year growth rate for that particular neighborhood. The resulting year-to-year growth rates are presented in Figure 1. The normalizing factor was then found by calculating the product of the growth rates for each neighborhood to rescale the year to the 2011 - 2012 baseline. The resulting normalizing factors are in the table presented in Figure 2. Note that in Figure 2, the first column's values are an integer mapping of the neighborhoods names presented Figure 1 (i.e. 0 is Chelsea, 1 is East Village, etc.).

Growth Rate	2008	2009	2010	2011
CHELSEA		10.94%	-3.20%	20.10%
EAST VILLAGE		7.58%	-0.61%	22.42%
GREENWICH VILLAGE		5.10%	0.36%	14.75%
HARLEM		5.88%	-1.61%	-6.26%
MIDTOWN EAST		7.78%	3.11%	10.65%
MIDTOWN WEST		4.62%	0.36%	9.92%
SOHO		0.44%	-3.70%	9.85%
TRIBECA		-0.25%	6.92%	6.70%
UPPER EAST SIDE		8.44%	3.41%	12.81%
UPPER WEST SIDE		5.90%	-3.32%	14.82%

Figure 1: Year-to-Year Growth Rates for NYC Neighborhoods

Normalizing Growth Rate	2008	2009	2010
0	128.97%	116.25%	120.10%
1	130.89%	121.67%	122.42%
2	121.04%	115.16%	114.75%
3	97.65%	92.23%	93.74%
4	122.96%	114.09%	110.65%
5	115.42%	110.32%	109.92%
6	106.25%	105.78%	109.85%
7	113.80%	114.09%	106.70%
8	126.51%	116.66%	112.81%
9	117.55%	111.01%	114.82%

Figure 2: Normalizing Growth Rates for NYC Neighborhoods

Lastly, in order to attain the geo-coordinate locations for the buildings that would be used to render the final visual presented in the results section, a Python program was written that mapped the building addresses to latitude-longitude coordinates. The resulting data was stored in a separate table and was not used by the algorithm.

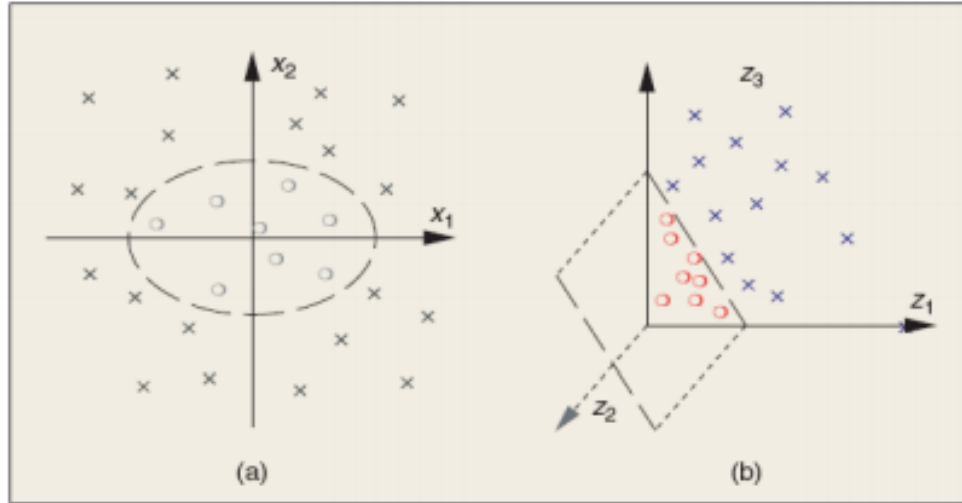
Technique

1. Price Prediction

The initial plan involved utilizing the Simple Least Square Regression model to determine the values of the hyper-parameters. But upon realizing that the data set is not linear separable (i.e. there is not a linear relationship between the features and the final price), a new technique was sought. From research, the Kernel method was identified as an alternative candidate for solving the problem. The Kernel method describes a mechanism for turning a non-linear separable data set into a linear separable one. The principle behind the Kernel method is that a data set can be mapped into a higher-dimensional vector space where linear relations exist within the data. The mapping can be accomplished via numerous algorithms. In this project, a Gaussian kernel function, shown as Figure 3, was selected to build the kernel matrix, K .

$$K(x, x') = e^{-\frac{\|x - x'\|^2}{\sigma^2}}, \sigma > 0$$

Figure 3: The Gaussian kernel function



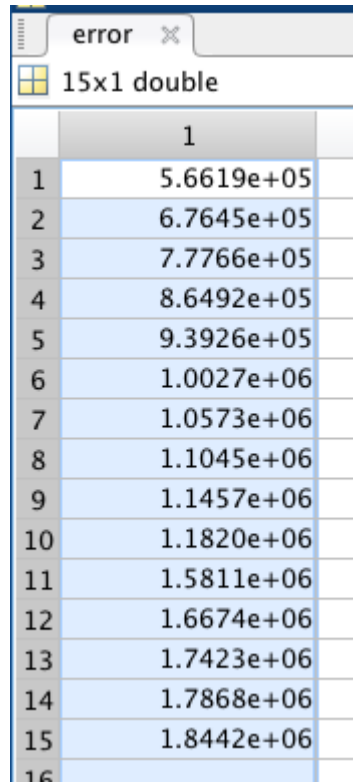
▲ 1. Effect of the map $\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ (a) Input space X and (b) feature space H .

Figure 4: Example of Hyper-plane Kernel Mapping

With K constructed, a cross validation method was utilized in order to obtain a more accurate result. The cross validation technique was utilized to compare a series of σ values and identify the optimal value. The series of σ values tested are presented in the array below:

$m = [0.02 \ 0.04 \ 0.06 \ 0.08 \ 0.1 \ 0.12 \ 0.14 \ 0.16 \ 0.18 \ 0.2 \ 1 \ 2 \ 5 \ 10 \ 20]$

Upon executing the cross validation strategy, the error rates found for the tested σ values are presented below in Figure 5:



	1
1	5.6619e+05
2	6.7645e+05
3	7.7766e+05
4	8.6492e+05
5	9.3926e+05
6	1.0027e+06
7	1.0573e+06
8	1.1045e+06
9	1.1457e+06
10	1.1820e+06
11	1.5811e+06
12	1.6674e+06
13	1.7423e+06
14	1.7868e+06
15	1.8442e+06

Figure 5: Error Results of Cross Validation

Thus, 0.02 was identified as the optimal σ , because it yielded the smallest error rate.

Upon completing the mapping, any linear algorithm can then be applied to the new space. For this project, a simple linear regression algorithm was chosen, though other alternatives such as Support Vector Machines (SVM) exist. In order to attain the regression result for the new data (testing data), the following formula needs to be applied:

$$f^*(x) = \sum_i \alpha_i K(x_i, x)$$

where $K(x_i, x)$ can be used with the Gaussian kernel function to calculate α_i . Note that different linear regression algorithms provide different formulas for determining α_i . In the case of the Least Squares Regression algorithm, α_i can be calculated with the following expression:

$$\alpha = (K + \lambda I)^{-1} \mathbf{y}.$$

Presented below in Figure 6 is a table containing the average percentage by which the predicted values for the price per square foot differed in each neighborhood as compared to the actual values for the 2011 - 2012 data set. Chelsea (neighborhood 0) was the most accurate with an average percentage difference of 0.32%; whereas Harlem (neighborhood 3) had the greatest disparity with an average percentage difference of 32.45%.

Neighborhood	Percent Error
0	0.32%
1	12.66%
2	11.50%
3	32.45%
4	1.38%
5	3.52%
6	17.65%
7	9.89%
8	1.83%
9	8.88%

Figure 6: Average Error for the Price Prediction Algorithm by Neighborhood

The results of the price prediction algorithm are displayed visually in the results section below.

2. Neighborhood Classification

The neighborhood classifier simply utilized the k-nearest neighbors algorithm (k-NN) with the first 3-years of normalized data composing the training data set. As each training data entry already possessed a neighborhood classification, this allowed for the algorithm to be easily applied. A simplified description of the 2-dimensional case (1-NN) of the k-NN algorithm is depicted below in Figure 7.

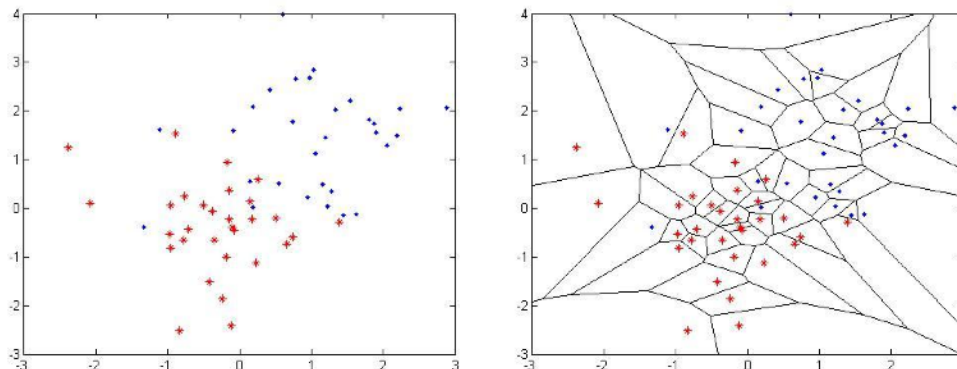


Figure 7: The segmentation of the space in 1-NN

In the training part, all the training data are placed on the data space similar to the plot on the left in Figure 7. Subsequently, the data space is segmented into the pattern shown in the right-hand plot of Figure 7. Each segment of the space only contains one training datum, which is the one nearest neighbor within the segment. During the learning process, when the testing datum is added into the trained space, it will always fall into one segment. At this stage it will be classified into the class of the segment's pinpoint.

Empirically, all the training data is inputted into the k-NN classifier from the Matlab library, which automatically completes the prediction along with the bagging procedure to enhance the prediction accuracy.

The resulting predictions are shown in the table below:

Neighborhood	Initial Size	Final Size
0	79	404
1	44	34
2	96	84
3	109	67
4	52	32
5	42	32
6	37	3
7	57	20
8	194	133
9	171	72

Figure 8: Table of Initial (Actual) and Final (k-NN) Neighborhood Size

Note that Chelsea is the largest clustering with a final size of 404 compared to an actual size of 79 (obtained from the original 2011 - 2012 data set). From this it is apparent that many buildings throughout New York City contain characteristics that are similar to the profile of a Chelsea condominium building. This is in contrast to the SoHo cluster which only contains three points. As this represents a significant reduction from the actual number obtained from the 2011 - 2012 data set, it is evident that few buildings actually meet the profile of a SoHo condominium building. The results presented in the above table are presented visually in the results section below.

Results

Presented below in Figure 9 is a map of New York City overlaid with dots on geo-coordinates of the examined data. The green dots, of which there are 166, represent good investment opportunities because the actual market price of those properties is less than the predicted market price by at least 20%. This indicates that the property is undervalued, and thus that it is a good investment. The red dots, of which there are 21, represent bad investment choices because the actual market price of those properties is greater than the predicted market price by at least 20%. This indicates that the property is overpriced, and thus that it is not a good investment. The grey dots represent the properties that fall within the 20% tolerance of the actual price, thus they are properties that are appropriately priced according to the price prediction algorithm.



Figure 9: Price Prediction Results Map of NYC

Figure 10 contains the data points from the actual 2011 – 2012 data set overlaid on New York City. The ten major neighborhoods are depicted without significant overlap.



Figure 10: Map of NYC Overlaid with Neighborhoods (Actual Data)

Upon running the k-NN algorithm, a new set of clusters was identified. The resulting set is depicted below in Figure 11. From the results it is apparent that many of the condominiums in New York City are very similar and differ in few characteristics. This is evident because of the large size of the Chelsea cluster, which contains 404 entries along with the significant overlap in the resulting clusters. Due to the large overlap, clear delineations between the clusters are hard to identify. From this it can be concluded that much of the New York City condominium is homogenous in terms of the characteristics analyzed by the algorithm.

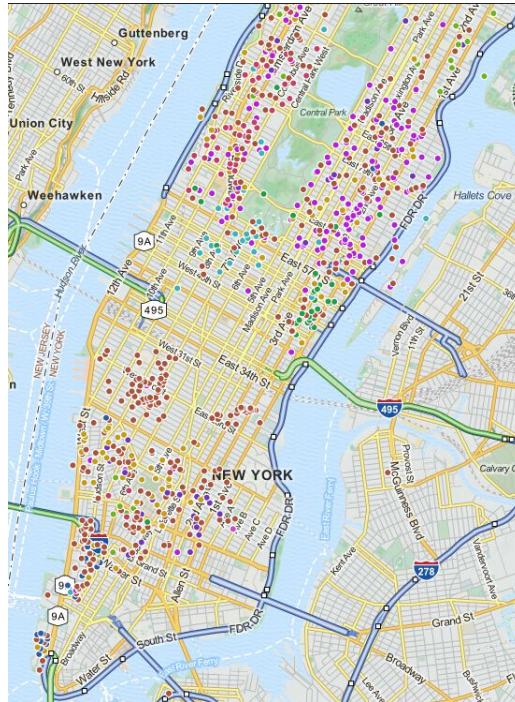


Figure 11: Map of NYC Overlaid with Neighborhoods (k -NN Data)

References

1. { HYPERLINK "<https://nycopendata.socrata.com/>" \h } { HYPERLINK "<https://nycopendata.socrata.com/>" \h }
2. { HYPERLINK "<http://people.cs.pitt.edu/~milos/courses/cs3750-Fall2007/lectures/class-kernels.pdf>" \h } { HYPERLINK "<http://people.cs.pitt.edu/~milos/courses/cs3750-Fall2007/lectures/class-kernels.pdf>" \h }
3. { HYPERLINK "http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm" \l "Algorithm" \h } { HYPERLINK "http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm" \l "Algorithm" \h }
4. { HYPERLINK "http://en.wikipedia.org/wiki/Kernel_method" \h } { HYPERLINK "http://en.wikipedia.org/wiki/Kernel_method" \h }
5. { HYPERLINK "http://en.wikipedia.org/wiki/Gaussian_process" \l "Applications" \h } { HYPERLINK "http://en.wikipedia.org/wiki/Gaussian_process" \l "Applications" \h }

Suhas Gudhe, Peixuan Jiang, Yiran (Lawrence) Luo, Jun Wang
April 30, 2015
CSE 5522 – Alan Ritter

6. { HYPERLINK "http://en.wikipedia.org/wiki/Cross-validation_%28statistics%29" \h }
validation_%28statistics%29" \h }
7. { HYPERLINK "https://mangomap.com/" } { HYPERLINK
"http://en.wikipedia.org/wiki/Cross-validation_%28statistics%29" \h }