NYC Property Sales Price Predictions

by Alberto Fabian & Riazul Islam

Business Understanding

Identify, define, and motivate the business problem that you are addressing. How (precisely) will a data mining solution address the business problem?

Problem Statement

One of the most challenging marketplace actions to take is valuing a real estate property for purchase or sale. How can one put a price on a building, piece of land, or apartment when there is no marker of value other than what another person is willing to pay? Many people have relied on real estate agents, extrapolation from nearby property sales, and “intuition” to estimate how much a property would ultimately be priced at, but these relatively unscientific methods with potential agency problems do not empower buyers or sellers with a strong price predictor to rely on for decision-making.

Improving the buyer/seller perception on the pricing of residential properties for sale would help make the process faster and more transparent. A system that could calculate the expected price of a property would allow for there to be a single source of truth for both buyers and sellers of real estate to assess a property’s true price and ultimately create a more efficient market. In doing so, this would help reduce the time and money spent on the entire price-setting and negotiation process, ultimately saving buyers and sellers both time and money (note: exact measurement of this is outside the scope of the project and incalculable within this set of data as we only are given the final sale date and not the initial listing date).

Currently, there are only a few ways for potential buyers or sellers to estimate property prices. Past property sales prices in the local area is one way to do this, but relatively unscientific given that every property has different characteristics, and pure mental estimation does not systematically address how much a price should change when a variable in the property changes. Real estate brokers, who have extensive experience in the market, can also provide property price estimates, and given their knowledge of many past sales, can better estimate sales prices. However, they do have an interest in inflating sales prices, since they receive a commission on every sale they help a potential seller execute on, incentivizing them to find the highest bidding buyer, rather than helping to sell a property at fair value. They are further incentivized to find

the highest buyer, even if it is not sold at a sustainable price to the buyer, since that will ultimately help push up the price of other properties in the area, which may also be sold by the broker. Finally, companies such as Zillow have developed algorithmic estimation models to price properties that are up for sale on the seller side. These also suffer from similar agency problems as working with real estate agents, but also are not catered toward buyers or people who are earlier in the process and are simply exploring the market to see how much they may be able to make on a sale of their property. Developing a buyer and seller neutral product to analyze and predict the price of a property would eliminate the agency problems and provide a service to avoid unnecessary negotiations on property sales.

The Solution

In this exercise, we used a dataset called NYC Property Sales from the City of New York, featuring a year's worth (September 2016 - September 2017) of NYC property sales, to attempt to accurately predict residential property sales prices across New York City's five boroughs, based on a number of features contained in the dataset. These features that are attached to sales from the 12-month period, include building class category, year built, borough, neighborhood, zip code, year built, tax class, and average sale price in borough/neighborhood/zip code, will be used to determine the most important factors affecting property prices in sales. In addition, we assessed a number of models against predictors of final sale price to attempt to identify the best possible predictor(s) of residential real estate price.

This attempt at modeling price prediction can help both buyers and sellers understand the true price of the property they are trying to buy or sell. As a result, we would anticipated end users saving time on negotiations and having more transparency in understanding pricing expectations from both buyers and sellers. Since this is a solution geared toward both groups (as opposed to Zillow, which is almost entirely only geared towards buyers, for example), it is hoped that both will find this to be a fair and equitable prediction model.

In our business use case, the models that we discuss in our analysis would ultimately be used to power an online tool that would be available to both buyers and

sellers in the market to assist them in making more informed decisions. Based on the features we use to train our model, our business case would be to provide users with an online form where they can enter the same categories of features used to provide them with an estimated sale price prediction. In our deployment section, we will cover more about the opportunities and risks of using this tool, but first we will explore the data and underlying models tested in this exercise.

In the following sections we will provide an overview of what is contained in the dataset, our approach to cleaning up and preparing the data, and our different modeling methodologies. We will also provide our evaluation of how these models stack up against each other in terms of accuracy for predicting price, as well as discuss how they might be evaluated in a business environment. Finally, we will cover how we would expect to deploy these models in a business context and the opportunities and risks associated in doing so, especially if we were to scale this solution.

Data Understanding

Identify and describe the data (and data sources) that will support data mining to address the business problem. Include those aspects of the data that we routinely talk about in class and/or in the homeworks.

The City of New York publishes a public dataset on a rolling 12-month basis containing a record of every building, building unit (apartment, etc.), and lot sold in the New York City real estate market across its five boroughs. This particular dataset found on Kaggle covers a 12-month period from September 2016 to September 2017. The original dataset prior to manipulation includes nearly 85,000 rows across 20 distinct columns, which are labeled as follows:

- Borough
- Neighborhood
- Building Class Category
- Tax Class at Present
- Block
- Lot
- Ease-ment
- Address
- Apartment Number
- Zip Code
- Residential Units
- Commercial Units
- Total Units
- Land Square Feet
- Gross Square Feet
- Year Built
- Tax Class at Time of Sale
- Building Class at Time of Sale
- Sale Price
- Sale Date

The subsequent analysis aims to leverage this data to predict the final sale price of a property given a set of features. As such, we define our target variable as Sale Price, which is provided in the dataset for each property. The remaining columns contain features to be used in subsequent models built to predict the target variable. Some of the features contained within this dataset are represented as nominal variables, which includes features such as Neighborhood and Borough. In order for these to be used in our modeling exercises, it requires that the data be manipulated, which we will cover in more detail in the next section (Data Preparation).

Since we are focusing on only residential properties, it also requires us to clean the data to remove commercial and all other properties not considered residential. Additionally, there are also some columns that contain little to no data, which we discuss manipulating and or removing below.

Another factor we had to consider is features that, though they are represented as integers in the dataset, are actually classes. This includes columns such as Zip Code, Block and Building Class. In understanding the data, it is important that we understand that it may not be apparent at first that some numerical values are actually class features and need to be manipulated for use in building a predictive model and is an important step in our data preparation.

Data Preparation

Specify how these data are integrated to produce the format required for data mining.

As mentioned previously, the dataset is fairly clean but requires some preparation in order for it to be used in the models that will be the basis of this exercise. Some light manipulation has already been done as the version provided on Kaggle is an aggregation of five datasets made publicly available by the City of New York. We have outlined the additional steps taken to clean the dataset in order to be usable in our models below:

1. The dataset is an aggregation of five different datasets (one for each of New York's five boroughs). These original datasets contain a column with numerical values for each row that correspond to the order in which each property appears within the original datasets. As this is an arbitrary numerical value, this column is removed from the dataset. (Dataframe Shape: 84,548 x 22)
2. The next step was to rename all of the columns to shorten naming conventions and remove spacing within multi-word column names. This step makes it easier to later reference column names. (Dataframe Shape: 84,548 x 22)
3. Since the focus of this exercise is on residential properties, we remove all non-residential properties. Building Class Category is represented as a string, which begins with a two-digit code and is followed by a short text description of what the Building Class Code means. The following steps address it (Dataframe Shape: 78,263 x 22):
    a. Created a "Building Class Category Trunc" column that pulls the first two characters (the two-digit code) from each string.
    b. We then converted the data in the new column to integers in order to manipulate the values.
    c. Finally, we removed any rows in the "Building Class Category Trunc" column that were >20, since residential units only includes Building Class codes up to 20 are used to designate residential properties.

4. Next, we removed all " - " values in the Sale Price column, which is due to either the data being unavailable or the transfer of property not being in exchange for any monetary figure. (Dataframe Shape: 65,723 x 22)
5. We also removed all sales prices below $10,000 as there were several entries with $1, $10, or other figures too low to be real residential property sales prices. This is primarily due to estate sales or transfers among families or as charitable donations. The $10,000 threshold helps eliminate the majority of properties that fall under these special cases. (Dataframe Shape: 55,985 x 22)
6. Our next step was to add three columns that calculated the average sale price at the borough, zip code, and block levels for each given row of data. This can serve the purpose of being fed back into the model as a feature to help predict price, as well as help calculate percentiles if needed. (Dataframe Shape: 55,985 x 25)
7. The next step was to create binary code for all the categorical variables so that they could be run through linear and polynomial regression. This was accomplished using the Get Dummies command, which pivoted out each unique class within the categorical variable columns and assigned them either a 1 (if that feature exists) or 0 (if that feature doesn't exist) for each property sale. This was done for all of the following columns (Dataframe Shape: 55,985 x 746):
    a. Neighborhood
    b. Building Class Category
    c. Tax Class at Present
    d. Building Class at Present
    e. Zip Code
    f. Year Built
    g. Tax Class at Time of Sale
    h. Building Class at Time of Sale
8. Lastly, we removed specific columns from our dataframe for the following reasons (Dataframe Shape: 55,985 x 737):
    a. Some features are not possible to be in regression since these are categorical or cardinal and are not helpful as a binary coded variable. In

some cases, these may also be too granular for our purposes. These features include:

i. Block
ii. Lot
iii. Address
iv. Apartment Number
v. Sale Date
vi. Building Class Category Trunc
    1. Note: This column was only created to filter out specific rows by converting the original Building Class Category from a string into a numeric value that could be manipulated. The original Building Class Category column is retained and pivoted using Get Dummies.

b. These columns were removed from our dataframe due to containing little or no data. In a more complete dataset, we would likely keep square footage metrics as we anticipate it to have a strong positive correlation with final sale price:
    i. Ease-ment
    ii. Land Square Feet
    iii. Gross Square Feet

In addition to cleaning up the dataset, we use the scikit-learn module in Python to separate our data into test and training data using *train_test_split* to create training data (80% of the overall dataset) and test data (20% of the overall dataset). For each model type that we'll cover in the sections that follow, we identified "Sale Price" as the target variable we aim to predict.

Modeling

Specify the type of model(s) built and/or patterns mined. Discuss choices for data mining algorithm: what are alternatives, and what are the pros and cons? Discuss why and how this model should "solve" the business problem (i.e., improve along some dimension of interest to the firm).

We identified a number of regression models to test in attempting to best predict residential property prices. In attempting to predict prices, we knew we had to use regression algorithms and not classifiers, so that threw out a number of potential models we could build based on our class learnings. However, many of those classifier models we learned about (such as Decision Tree Classifier) also had regressor equivalents (such as Decision Tree Regressor) that we could instead utilize as part of our modeling and testing.

Initially, we tested on a number of linear regression models, under the belief that we might use a simple model to test for directionality of model elements and a basic understanding of what to see from a regression model on residential property prices. We came into this with certain expectations, based on our (somewhat limited) understanding of the New York City residential property market, but which helped guide our interpretation and "acceptance" of model outputs. For example, we expected one family dwellings (identified using Building Class Category) to yield a lower price than two or three family dwellings. Similarly, we expected properties in Manhattan (identified using the Borough code) to generally yield higher prices than Staten Island.

The initial linear regression models were based on the scikit-learn's Lasso and Linear Regression models.[1] With the Lasso model, we used alphas = 0.01 and 0.1, indicating that we are introducing two different penalty levels for the number of features used in the model as we have 737 columns containing features. These actually yielded very similar results on the coefficients, but since the coefficients were generally in the correct direction (e.g. being in Manhattan yields a higher price than being in Staten Island, as discussed as an expectation earlier), we knew that the dataset would yield some prediction model, through whichever regression algorithm we would ultimately choose, that would answer our initial business question. We also ran a general linear

[1] Scikit-Learn.org. Last accessed April 30, 2019. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html#sklearn.linear_model.Lasso

regression model, also from scikit-learn's Generalized Linear Models module (linear_model), but with a weaker model fitting process, coefficients for each variable that were extremely large relative to the Lasso models, and likely inaccurate given the variance in sales price.[2] This is due to the Lasso Regression model introducing a regularization mechanism by which the greater the alpha, the closer to zero the coefficient for each feature is driven. This helps yield a better fitted model compared to a Generalized Linear Model.

In discussion with Professor Provost, we identified a number of additional models that we could test for improved accuracy versus our initial linear regression models. These included decision tree and k-nearest neighbor regressions. Each of these have different approaches to fitting data, which we can use to compare accuracy and determine, which best addresses our business problem. We also discussed how additional data manipulation could help improve the model by accounting for outliers in the data.

In addition, we were unsure of how to test for accuracy with these models beyond splitting the data into training and test data and looking at the coefficients, so we identified mean absolute error and mean relative error as potential error metrics for accuracy calculation. These would be relatively easy to calculate, but based on our modeling attempts (as will be discussed later), these would be useful for a number of checks, including poor accuracy and overfitting. We also discussed potentially creating high / low price models (with different regression algorithms) based on how well the models predicted values at various parts of the price range, as well as splitting a single regression model into high / low models.

Based on these discussions with Professor Provost, we started by revisiting our lasso models again, but calculated the mean absolute error and accuracy metrics using 80% training and 20% test data. For the alpha=0.01 model, though the coefficients looked correct in direction and magnitude, the mean absolute error was $759,030.13 and the accuracy was -37.99%, extremely poor numbers. However, given the convergence warning thrown by the algorithm, we also tested different thresholds for

[2] "How to Run Linear Regression in Python scikit-Learn." Big Data Made Simple. March 5, 2018. https://bigdata-madesimple.com/how-to-run-linear-regression-in-python-scikit-learn/

the alpha (e.g. alpha=0.1 and alpha=0.5), which yielded almost exactly the same mean absolute error and accuracy, leading us to believe that the lasso models would not be a good fit for predicting residential real estate prices. Even though the normal linear regression earlier was extremely poor just from observing the coefficients, we still calculated a comically poor mean absolute error of $9,941,303,283,799.51 and equally poor accuracy of -4,144,893,311.47%. Adding average price per zip code as an independent variable did not help these regressions either, leading us away from any type of linear regression model as part of our solution.

When running the decision tree regressor model, we initially ran it with a min_samples_leaf value of 5, which led to a mean absolute error of $3,911.41 and an error accuracy (based on mean absolute error) of 99.96%.[3] The latter metric was very alarming, since it seemed to suggest heavy overfitting of the data. As a result, we shifted to using max_depth of 7 instead of min_samples_leaf as our tree limiter. This increased the mean absolute error to $57,360.66 and reduced the accuracy to 85.74%. However, this also seemed high enough on accuracy to drive the possibility of overfitting, so we removed average price for borough, zip code, and block as predictors, which actually resulted in a very similar mean absolute error and accuracy. Finally, we tested with a max_depth of 5, which increased the mean absolute error to $218,194.39 and an accuracy of 47.62%, and a max_depth of 6, resulting in a mean absolute error of $110,581.8 and accuracy of 72.71%.

Next, we tested the k-nearest neighbor model, which allowed us to cluster similar residential properties together to allow us to predict pricing for other similar properties.[4] At first, we ran a standard k-nearest neighbor regression model with 5 nearest neighbors using the algorithm from Scikit-learn neighbors module, which yielded us a mean absolute error of $492,582.17 and an accuracy of 35.3%. However, this model included average price within borough, zip code, and block, and since we believed that the borough level is too broad of a price range and block is too narrow of a data sample, we tried this model with just the average price at the zip code level. This resulted in a

[3] Python Machine Learning Tutorial. https://www.python-course.eu/Regression_Trees.php
[4] “Machine Learning Fundamentals: Predicting AirBnB Prices.” Josh Devlin. DataQuest.io. August, 31, 2017. https://www.dataquest.io/blog/machine-learning-tutorial/

worse mean absolute error of $542,603.27 and an accuracy of 22.27%. Removing the average price at the zip code level actually improved the mean absolute error to $560,425, while reducing accuracy to 8.47%. Increasing the number of k-nearest neighbors to 10 and then 20 increased the mean absolute error and reduced accuracy further, but by marginal amounts. If we reduced the number of neighbors to 3, then the mean absolute error decreases slightly to $551,892.92 and accuracy improves to 22.6%, but these are still very inaccurate results. It seems like the kNN model simply doesn't work very well, possibly because there are too many binary variables in the model to accurately cluster similar properties. Overall, the k-nearest neighbor model seems to have not been very accurate for our analysis.

Since all these models drove extremely large mean absolute errors relative to prices, we decided to try two methods to improve errors: 1. reducing the range of property prices in the dataset to $75,000- $10,000,000, and 2. further splitting that dataset into $75,000-$500,000, $500,000-$1,000,000, and $1,000,000-$10,000,000 ranges. When we first tried the $75,000- $10,000,000 price range, with the kNN and decision tree models, we saw a small but not truly significant gain in mean absolute error and accuracy, so we then split the model into the three smaller ranges mentioned above. For these ranges using the decision tree regression model with a max_depth of 5, we ultimately calculated an average of ~30% mean average error for each of these ranges, with 70-85% accuracy on each.

Ultimately, we chose the decision tree regression model, split into the three aforementioned ranges, with the average price per block as an independent variable. Though the errors were still relatively high (average of ~30% difference from actual price), these three models helped us narrow down to specific ranges that seemed appropriate given that the relative error was consistent across these three. In addition, the ~30% difference in price between predicted and actual on test data was much lower than some models that were much worse in both mean average error and accuracy. In choosing these three models, we believe they can work in tandem to "solve" the question of identifying the estimated fair price for buying/selling a residential property in New York City, since only a few features are required to price a property equitably to both buyer and seller.

Evaluation

Discuss how the result of the data mining is/should be evaluated. How should a business case be developed to project expected improvement? ROI? If this is impossible/very difficult, explain why and identify any viable alternatives.

While the mean absolute error is relatively high at ~30% of price within each selected model's price band, we believe that the models are still quite strong given the relatively few predictors we used to ultimately create predictive models. Given that models we developed using other algorithms yielded mean absolute errors that could be around 100% or more of expected price, our final collection looks to be a relatively strong set given the dataset with which we worked.

Conveniently, Zillow Group's Zestimate tool, used to estimate a market value of an individual home based on public and user submitted data, rates itself a 2 out of 4 stars on estimation accuracy, with 84.1% of Zestimates on sold properties within 20% of the actual sale price. While our models are not as accurate, we believe that, given the fewer data points that we had (see below for additional information on model improvement and deployment methodologies), our models approach a relatively high level of performance in coming close to the Zestimate accuracy.

Expected improvement on this model is relatively straightforward: gather additional and more complete data on metrics such as year built, number of bedrooms, and square footage, and use those metrics to build a much more robust set of decision tree regressor models. In addition, testing with more advanced modeling techniques unexplored here (such as neural network regression) could help drive a more improved model with lower error. Finally, expanding the model beyond New York City could yield further improvements, but with New York City as a unique market, the final models may actually predict lower prices for properties in the city based on a wider market dataset.

There are two potential returns on investment to calculate here: the first is the ROI to the company creating this tool, and the second is the ROI for actual users choosing this tool over others to understand the potential price of their property. For the company that may develop this model into a tool for users, the ROI would be the difference between the revenue generated from the tool and the cost of developing and maintaining the tool. Revenue generation could come from a number of channels,

including advertising on a free version of the tool or offering premium services to paid customers. After developing the tool, the primary costs will be through maintenance, incremental upgrades, and maintaining access to the data necessary to continue predicting property prices.

For users, the ROI is much more challenging to calculate. We believe this would require a market research assessment of the time spent by people who are curious about the cost of the price of a property or properties they are looking to buy or sell, and how much time is saved by using the tool to come to an accurate pricing assessment instead of working with brokers or doing additional self-guided research. In addition, there may be buyers or sellers that find a more favorable price for property in question for purchase or sale, and so additional ROI can be calculated based on the improvement of pricing ultimately obtained through use of the tool for better intel.

Deployment

Discuss how the result of the data mining will be deployed. Discuss any issues the firm should be aware of regarding deployment. Are there important ethical considerations? Identify the risks associated with your proposed plan and how you would mitigate them.

The Product Solution

Our solution aims to address the real estate market from both the buyer and seller perspective. As such, this model would be deployed via an online tool where users can input features that are used in this model into a form. These features would then be fed into a model that would predict an estimated Sale Price for a property given the set of features provided. We can think of this solution through both sides of the market:

- *Sellers*: Online tools created by companies such as Zillow currently provide sale price estimates for sellers. Such tools use a variety of data sources (e.g. historic property sales, mortgage rates, etc.) to build a model that given a set of feature inputs from users provides pricing estimates. One risk to our solution in this side of the market is that current solutions are naturally more competitive relative to our existing model due to their availability of other quantitative and qualitative features not captured in our limited dataset (i.e. number of bedrooms, number of bathrooms, garage (yes/no), etc.). This is something we may want to consider leveraging if we were to expand the scope of this project to include additional or richer datasets.
- *Buyers*: On the buyer side of the market, there isn't a large consumer-facing solution where users can enter features and get an estimate of what a property might cost given these inputs. This is a mostly unaddressed market that could benefit from such a tool to help with planning home-buying. This helps increase buyer power by making them more informed when approaching the real estate market.

By focusing our solution solely on the upper parts of the customer funnel (awareness and consideration), we can position our tool to be more of a research tool

rather than a marketplace tool. This means that users wouldn't transact over our platform. There is also an opportunity to scale and improve the model, which we will address in the current limitations. As a standalone planning tool, in initial deployment it would live on its own website and generate revenue through advertising as the tool pulls in website traffic. At a future point, we could explore the possibility of licensing out this model to existing entities in the real estate space.

Limitations of Existing Solution

By nature of this project being in an academic context, we encounter limitations on the scope and scale of publicly available data. There are multiple data-related factors that we would attempt to address first as part of our deployment strategy in a business context to further refine our model. A few of these considerations include:

- *Volatility*: At present, we are using one year of data, which introduces bias for market conditions in that given year. A scaled solution should include both more recent and more historic data to better normalize our model for variance in market risk factors over time.
- *Additional Feature Categories*: The public dataset made available by the City of New York does not capture qualitative information about properties that could help better zero in on a more accurate price estimate for a given property. Such data might include number of bedrooms, fixtures, appliances, etc.
- *Missing/Limited Data*: The original dataset included columns for data related to square footage, which was included for only a handful of properties. As mentioned previously, we would anticipate this feature to be strongly correlated to sale price. As such, we would explore ways of getting a more complete dataset that contains this metric in order to better train our model.
- *Changes in Market Conditions*: Major recent neighborhood changes would not be reflected in the data. For instance, if Amazon announced that it was opening a new office in a given neighborhood, that cause the property value in that area to spike, but since the model is trained using historical data, it would not adjust for these new market conditions until properties were sold under new market

conditions and fed into the model. In this case, we would want to introduce a mechanism into the model that can account for these changes closer to real time by tapping into an additional data source.

Ethical Considerations

First and foremost, transparency and consumer education on any tool powered by this model should be a priority. From an ethics perspective, we should prioritize ensuring that our end users understand how the price estimator works and that it is not an appraisal of a property, but rather an estimate to use as an anchor point as they begin the process of buying or selling a property.

Another ethical concern in general is inaccuracy due to unavailability of data due to changes in market conditions, which we mention in the section above. Given that the model relies on historic data, we run the risk of artificially suppressing sales price estimates. The model may also be inherently better at predicting price in some geographic areas versus others, which can have repercussions if, for instance, this negatively impacts areas with limited data availability such as lower income neighborhoods for instance. This all ties back to the first point of needing to invest in consumer education in marketing efforts as part of deployment.

Appendix (Contribution of each team member)

Our group worked together on essentially the entire project, working together to understand the business problem and data, clean up the data, and code many of the models we ran. After that, since we both were grounded in the same approach and business/data understanding, we were able to develop and run additional models independently, and write separate parts of the paper for later merging. We were able to split the work equitably, and support each other on the challenges of coding and model development when necessary.

Unfortunately, one of our group members left the class late in the semester, and so we were required to take on his portion of the work. However, he did contribute to our project by helping us select this project, after we had great difficulty finding a suitable project idea and related dataset in the initial going.