Estimation of Sale Price of Residential Properties in Manhattan: with Supervised Learning Approaches

Young Jin Park: yjp228@nyu.edu
Soojin Kim: sk5291@nyu.edu
Sujeong Cha: sjc433@nyu.edu

***Abstract*— Unless you are a real-estate guru, purchasing your first home, or even selling one is a very painful experience. Unlike everyday grocery shopping, "house shopping" does not occur as often so that "learning from past experience" does not work. If you are purchasing your home, there are several features that you appreciate which may or may not align with the features which the seller appreciates and closing prices are often met somewhere in the middle. However, how do you know that you are paying or receiving the "Fair" market value of the property of interest? How do you even set a "starting point" for your dream purchase or sale? You might have a friend who recently purchased his/her home or a broker working for you to tell you that number based on their recent experience or even from haunch. Our goal is to assist those who are in need of such "starting point" by providing them the estimated "market" value of the property of interest via leveraging data and machine.**

I. BUSINESS UNDERSTANDING

Although one might expect the New York real estate market to be notoriously competitive, Manhattan homes face longer and longer wait times to reach the closing. According to StreetEasy Market Reports, Manhattan homes in particular spent a median of 117 days on the market in February 2019[1]. Even well-positioned buyers hesitate to buy homes, walking away from deals that are not attractive because they are often uncertain about whether the listing price represents the actual value of the house. To price the homes sensibly and still take advantage of the important features that affect housing prices and get the highest return, the paper explores how various location-specific factors, on top of the house features, influence the selling price. By data mining diverse information about the location such as crime rate, pest control, noise complaints, distance to park, we can identify the most important features that predict the selling price of Manhattan homes.

Our data mining solution will help both buyers and sellers to make informed decisions. Since our model identifies favorable features as well as risk factors related to house prices, buyers can leverage

[1] McDonald, Emily. "Manhattan Homes Linger on Market, Forcing Sellers to Cut Prices." *StreetEasy*, 21 Mar. 2019, https://streeteasy.com/blog/february-2019-market-reports/.

the information to better understand the attributes that influence the house price and negotiate with the seller to settle a reasonable price. Moreover, sellers or real estate agents can make better decisions about the listing price to sell their houses quickly because a property that was on the market for a long period of time tends to lose value.

II. DATA UNDERSTANDING

A. *Data collection and Feature Extraction*

To predict the selling price of a new house in Manhattan based on historical data, house features and location-specific features such as crime rate, pest control and noise complaints, we combined data from various sources into a single cohesive dataset.

B. *Historical Sales Data*

The NYC Departments of Finance has made the housing sales records available to the public on the NYC Open Data portal. This dataset displays sales information of properties sold in New York City. The information includes neighborhood, building type, address, year built, sale price, sale date and more (21 features in total). We have set the time frame of data to 57 months, starting from Jan. 2015 to Sep. 2019 in order to fully reflect the historical trends in housing prices. Moreover, we have zoomed into Manhattan borough only as the price variance across five different boroughs might harm the accuracy of our prediction.

C. *Number of Bedroom and Bathroom Data*

Unfortunately, NYC Department of Finance did not have information about the number of bedroom and bathroom, the features that are highly correlated with sale price of a house. We were able to locate partial bedroom/bathroom counts per unit from Zillow website (most of the past sales posting seemed to not have bed/bathroom count which seems to be deleted after the deal was closed).

D. *Complaints / Crime Data*

We have hypothesized that a house in undesirable neighborhood conditions is likely to be valued at a lower price compared to the houses with the same features but more well-maintained surroundings. In order to proxy the environmental factor, we have collected data on complaints received by NYC 311 and NYPD. Both datasets are available through the NYC Open Data portal. The raw records of complaints were cumulative from 2010 and were

highly intractable (i.e. NYC 311 complaints data contained 22M rows and 41 columns); filtering on conditions was necessary but filtered data was not readily available to download as a CSV. It had to be exported with Socrata API which is a data access tool generally used on data from governmental, non-profits, and NGO bodies. Using Socrata API, we have filtered the raw data out on 'borough = Manhattan' and 'created_date = 2015.01 ~ current'. Of many complaint types, we have restricted 311 complaints to those under 'Rodent', 'Sidewalk Condition', and 'Noise - Commercial' types (chosen by heuristic), and crime complaints to 'Felony' crimes which can be comparatively far more critical. In addition, a (longitude, latitude) tuple was added to each instance in order to count the number of civil/crime complaints in nearby area of a house.

E. Selection Bias

Since the property data is entered manually by humans, there was missing and evidently flawed information. For example, a large number of records had typos in addresses and missing values for number of residential units in the building (which we ended up not using due to the poor quality of the data). Also, the dataset included the transactional data on the entire building purchase, as opposed to a single unit transaction, and countless transactions with unreasonably low sales price. For the former case, we dropped transactions with extremely large sale price, number of units, type of building and missing apartment number. For the latter case specifically, we interpreted these values as errored or assumed it to be non-market transactions (such as inheritance) and tried to eliminate such instances by dropping all the instances with sale price of $20,000. The lack of information, whether it was not available or incorrect, reduces the representativeness of the samples and thus, could have introduced bias in the prediction. Moreover, when we merged historical sales data with Zillow's bedroom and bathroom data, luxurious houses were excluded because many of luxurious houses are sold "off-market", not through online real estate marketplace (i.e. StreetEasy, Zillow, etc.). As a result, the final reduced dataset is non-representative of all the houses that were sold in Manhattan.

III. DATA PREPARATION

A. *Data Instance and Data Integration*

Our data instance is an individual housing unit in Manhattan area with corresponding characteristics. However, there was no comprehensive dataset that contained the features of our interest. Therefore, as explained in II. Data Understanding, we pulled data from multiple sources such as Zillow, NYC Department of Finance, NYC 311, and NYPD, and integrated them into one. In the process of merging these multiple datasets, *Historical Sales Data* from NYC Department of Finance served as a base.

- **Merging *Historical Sales* with *Zillow* data:**

By using the address of an instance ('street number' + 'street name' + 'apartment number') as a unique key, we inner joined *Historical Sales* data with *Zillow* data. The initial sales data contained 67,349 instances. After merging by the address, the number was reduced to 10,453 instances.

- **Adding features with Google Maps API and geopy package:**

For each instance in our processed dataset, we converted each address into geocodes (longitude, latitude) by using GoogleMaps API.

After converting all the addresses into geocodes, we used "distance" method from geopy package and calculated and merged the following features for each instance: 'number of complaints', 'number of felonies', 'minimum distance to park', 'minimum distance to subway', '6-month average housing price within 0.5 mile' Missing values in the feature '6-month average housing price within 0.5 mile' were imputed with the median value.

- **Converting categorical to numerical:**

As *"scikit-learn works on any numeric data stored as numpy arrays or scipy sparse matrices"* [2], categorical features (Zipcode and Building Categories) were converted to numerical by using the *get_dummies()* function in Pandas.

B. *Target Variable*

The target variable in our integrated dataset is the final sale price for each housing unit. As stated earlier, these values were originally obtained from

[2] "Frequently Asked Questions." *Scikit*, https://scikit-learn.org/stable/faq.html.

the *Historical Sales Data* provided by NYC Department of Finance.

C. *Correlation Between Final Features*

Before moving onto the modelling stage, we computed correlation coefficients between variables to grasp the sense of feature importance. Substantiating our naïve guess in the initial stage, 'Beds' and 'Baths' are the most highly correlated factors with 'SalePrice'. Interestingly, however, 'Baths' has a marginally more correlation with the price than 'Beds' does, which slightly contradicts with our assumption that the number of 'Beds' is the most critical variable in determining the price of a house.

IV. MODELLING & EVALUATION

Since our business problem is predicting the sale price of a house in Manhattan which is a numerical value, this is a regression problem. We chose to focus on some of the more common predictive modeling techniques which support this type of problem: linear regression, polynomial regression, decision tree regression, support vector regression and random forest regression. Based on the exploratory data analysis findings, along with basic knowledge on real estates, we set our baseline as median price per number of bedroom.

Choosing the best algorithm was a two-part approach. First, we evaluated the baseline model which obtained the R2 score of 0.518 and selected

	TaxClass	Baths	Beds	YearBuilt	#Complaints	#Felonies	ParkDist	SubDist	6MoAvg	Condos	Coop	OtherType	SalePrice
TaxClass	1	-0.41	-0.4	-0.021	-0.019	-0.0015	0.029	0.015	-0.013	0.091	0.25	-0.99	-0.39
Baths	-0.41	1	0.78	-0.14	-0.041	-0.033	-0.019	0.048	0.092	0.29	-0.42	0.41	0.62
Beds	-0.4	0.78	1	-0.067	-0.037	-0.031	-0.028	0.0024	0.096	0.23	-0.36	0.4	0.59
YearBuilt	-0.021	-0.14	-0.067	1	0.0091	0.018	0.015	-0.038	-0.07	-0.38	0.36	0.024	-0.041
#Complaints	-0.019	-0.041	-0.037	0.0091	1	0.24	0.024	-0.084	-0.052	-0.086	0.077	0.019	-0.066
#Felonies	-0.0015	-0.033	-0.031	0.018	0.24	1	-0.018	-0.21	0.014	-0.078	0.075	0.0016	-0.051
ParkDist	0.029	-0.019	-0.028	0.015	0.024	-0.018	1	0.37	-0.075	0.047	-0.035	-0.03	-0.034
SubDist	0.015	0.048	0.0024	-0.038	-0.084	-0.21	0.37	1	-0.23	0.11	-0.1	-0.016	-0.00048
6MoAvg	-0.013	0.092	0.096	-0.07	-0.052	0.014	-0.075	-0.23	1	0.18	-0.18	0.013	0.22
Condos	0.091	0.29	0.23	-0.38	-0.086	-0.078	0.047	0.11	0.18	1	-0.94	-0.092	0.23
Coop	0.25	-0.42	-0.36	0.36	0.077	0.075	-0.035	-0.1	-0.18	-0.94	1	-0.25	-0.36
OtherType	-0.99	0.41	0.4	0.024	0.019	0.0016	-0.03	-0.016	0.013	-0.092	-0.25	1	0.4
SalePrice	-0.39	0.62	0.59	-0.041	-0.066	-0.051	-0.034	-0.00048	0.22	0.23	-0.36	0.4	1

Figure 1. Correlation Matrix

top 2 models with the highest R2 scores. We used R2 score as evaluation metric because it has the advantage of being scale-free meaning that whether the output values are very large or not, the R2 is always going to be between -∞ and 1, and thus allows us to easily quantify how much our model is better than the baseline model[3]. Once top 2 models were selected, we used MAE to determine which one was the best. Although the most simple and common metric for regression evaluation is MSE, we chose to use MAE because MAE is more robust to outliers. For example, if we make a single very bad prediction, MSE will make the error even worse by squaring the error and given our target value is in millions, it becomes hard to judge how well the model is performing.

A. Model Specification - Algorithm

- **Linear Regression:**

We decided to use linear regression as the model attempts to fit a straight hyperplane to dataset line with minimum error from all data points. In addition, it is easy to interpret as we have more transparency on the weight of each feature. However, it assumes there is a straight-line relationship between them which is incorrect sometimes and is prone to outliers.

- **Polynomial Regression:**

Similarly, polynomial regression is a simple and common way to provide a non-linear fit to the data. It produces non-linear curves while still estimating the coefficients using least squares. However, as we increase the degree of the polynomial, we may end up choosing a wrong polynomial degree which will result in a bad bias/variance trade-off. As a result, in selecting the best polynomial degree, we compared the R-squared scores for training and testing dataset. Below figure shows that using polynomial regression to higher powers than required (in this case, greater than 2) results in overfitting and thus, a significant drop in R2 score for testing data. Therefore, we chose to proceed with polynomial degree of at most 3.

[3] Drakos, Georgios. "How to Select the Right Evaluation Metric for Machine Learning Models: Part 1 Regression Metrics." *Medium*, Towards Data Science, 5 Dec. 2018, https://towardsdatascience.com/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-1-regrression-metrics-3606e25beae0.

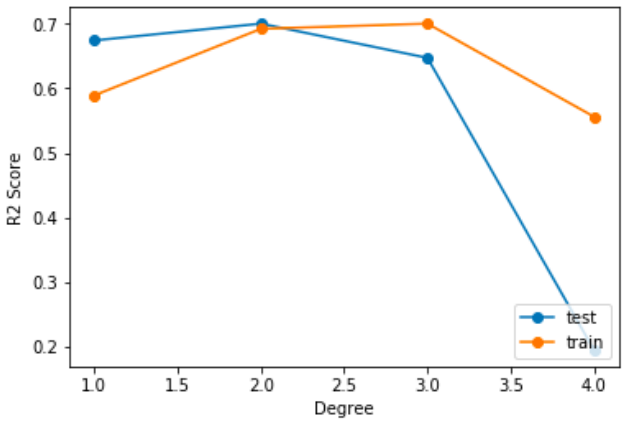

Figure 2. Comparing R2 Scores on Train and Test Data

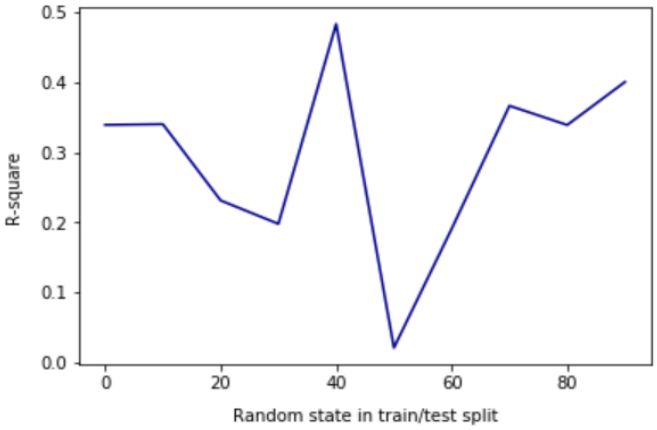

Figure 3. High variance of Decision Tree Regression

- **Decision Tree Regression (DT):**

Decision Tree algorithm uses a tree-like structure that recursively divide-and-conquer the data into two groups in a way that those two groups are heterogeneous at most. One great advantage of a decision tree is that it is easy to interpret and visualize. However, the algorithm can produce unstable results depending on small variations in the training set, resulting in high variance[4]. As the following graph shows, R^2 square fluctuates dramatically depending on the random state numbers when splitting train/test set.

- **Random Forest Regression (RF):**

One possible improvement over the instability of Decision Tree is to use bootstrapping aggregation, commonly known as bagging[5]. A Random Forest is an ensemble technique that employs multiple decision trees where sampling is done with replacement. As stated, the definite advantage of random forests is the stability of prediction results. Moreover, the cost associated with cross-validation is unnecessary as each training set is already sampled with replacement. On the other hand, the biggest disadvantage is the low interpretability which inherently comes from the fact that it is an ensemble,

[4] Drakos, Georgios. "Decision Tree Regressor Explained in Depth." *GDCoder*, GDCoder, 2 June 2019, https://gdcoder.com/decision-tree-regressor-explained-in-depth/.

[5] Krishni. "A Beginners Guide to Random Forest Regression." *Medium*, Data Driven Investor, 5 June 2019, https://medium.com/datadriveninvestor/random-forest-regression-9871bc9a25eb.

a mixture of different trees. This trait lenders the model less attractive as we aim to provide customers with some rationale along with the price predictions.

- **Support Vector Regression (SVR):**

Support Vector Regression is a regression version of the support vector machine classifier. A main difference from a linear regression, which finds a line that minimizes the least square errors, is that SVR tries to find a decision boundary along the original hyper plane so that all data points are within the boundary line. The disadvantage of SVR is that "it does not directly provide probability estimates; rather, these are calculated using an expensive, fivefold cross-validation[6]". However, the support vector regression, just as its corresponding classifier SVM, performs effectively in high dimensional spaces. This model can be suitable for our dataset as we have comparatively large number of variables and small volume of instances.

B. Model Specification - Features

To improve upon our baseline model and minimize the prediction error, we conducted a series of feature engineering steps as follows:

- Adjusted "min_dist_park" to "minDistSelectedParks" to only include Central Park, Bryant Park, Battery Park and Riverside Park
- Dropped "TaxClassAtTimeOfSale" feature
- Handled time related variable "YearBuilt" by converting it into three dummy variables Post-2000, 1980-2000, Pre-1980
- Changed "halfMileAvgPrice6mo" to "AvgPriceZipCode2YrBeds"
- Added new promising feature variables such as "LEASED" and "Floor_Proxy" based on domain knowledge

The major ideas/logic behind them are respectively:

- Since the original "min_dist_park" did not have much significance, we selected parks that have an impact on the sale price.
- We reduced the number of categorical variables to eliminate multicollinearity which may result in unstable coefficient estimates.
- We binned the variable appropriately as new-developments in general do not have conditions that require renovation or better amenities. If it's not a new-development, individual unit may vary in terms of the condition (may require severe renovation/may have already been renovated which is included in the sale price). The addition of "IsXXRenovated" (Where XX

[6] K. Roy, S. Kar, R. N. Das, "Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment", pp. 227, 2015.

being bathroom, kitchen) as features if such data is available may also improve the accuracy of the model.

- Although "halfMileAvgPrice6mo" represents location specific market sentiment, there were too few units to assess this value and did not have much significance. Therefore, we changed this variable to "AvgPriceZipCode2YrBeds". Logic is that sale price is contingent on the average price of n-bedroom units that have been sold within the specified window.
- Original dataset did not have any data on the floor, which is deemed to be correlated with the property value and was extracted by parsing the "AptNum" and applying specific rules to estimate the floor number. Also, buildings on land-lease tends to have lower property values because it is an undesirable trait by buyers. We gathered as much information we could get on the list of land-lease buildings and flagged the units in the land-lease buildings.

Type	Feature Name
Building-specific	Baths
	Beds
	Floor_Proxy
	BuildCat_CONDOS
	BuildCat_COOPS
	BuildCat_OTHERS
	BY_Pre_1980
	BY_1980_2000
	BY_Post_2000
	LEASED
Location-specific	min_dist_subway
	minDistSelectedParks
	NumComplaints3moPointOneMile
	NumFelonies3moPointOneMile
Market sentiment	SalesVolume
	AvgPriceZipCode2YrBeds
Target variable	SalePrice

Figure 4. List of final features

Besides feature engineering, we also improved upon the baseline by removing considerable sale price outliers (i.e. very expensive houses) and instances with number of bedroom greater than 5. The logic behind this is that many of such "luxury" houses with number of bedroom greater than 5 are not sold "on-the-market". Rather, most of them are sold "off-market" through renowned real estate agents. Moreover, in general, there are very few houses with more than five bedrooms in Manhattan. Therefore, instances with extremely high sale price and many bedrooms were excluded from our model. With this further reduced dataset, we obtain the following results from each model as shown in Figure 2. The results will be further discussed in Section C.

	Pre feature engineering	Post feature engineering
Linear Regression	0.445	0.674
Poly (deg=2)	0.536	0.700
Poly (deg=3)	0.526	0.647
Decision Tree	0.339	0.319
Random Forest	0.490	0.642
Support Vector	0.506	0.701

Figure 5. R2 Scores before and after feature engineering

C. Choosing the Optimal Algorithm

- **Ranking in terms of R2 Score:**

SVR ≈ Poly 2 > Linear > Poly 3 ≈ RF > DT

The R^2 scores of SVR and Poly (deg=2) were approximately equivalent; however, we have forgone Poly (deg=2) because some predictions from the poly model were negative while SVR produced strictly positive prediction for every instance. For instance, a house whose true price was $609,500 had the predicted price of -$1,609,535. Moreover, it was hard to interpret which variable resulted in such an error with large magnitude.

· # Baths:	1
· # Beds:	2
· Floor:	7
· Leased Land:	No
· Building Category:	Condos
· Year Built:	Post 2000
· # Complaints:	0
· # Felonies:	0
· Distance to Subway:	0.19
· Distance to Park:	2.84
· Avg Sales Volume:	865
· Avg Selling Price:	443,222
· Sale Price (true):	609,500
· Sale Price (prediction):	**-1,609,535**

Figure 6. Negative Prediction from Poly (deg=2)

The negative predictions account for 0.6% of the test set, which can be deemed as not influential; however, as our primary objective is to present a "price", we should avoid such negative predictions however rare it may be.

Note: As stated above, polynomial models produced negative predictions for some instances. This might be due to the fact that the size of our training dataset is not large enough to offset the effects of outliers (or what is deemed as outliers currently due to lack of data in the neighborhood of such data point). If we gather more and more data, it might be plausible for the poly models to reduce the impact of outliers and produce more "safe" predictions.

- **Comparing SVR and Linear Regression:**

After excluding Poly (deg=2) for the above rationale, we proceeded to compare SVR with the next best model, Linear Regression. Considering that the difference in R^2 score is only 0.03, Mean Absolute Error (MAE) was used as a metric. The following figure shows that for SVR, the range of MAE is narrower, and more instances are within the error rate of 50%.

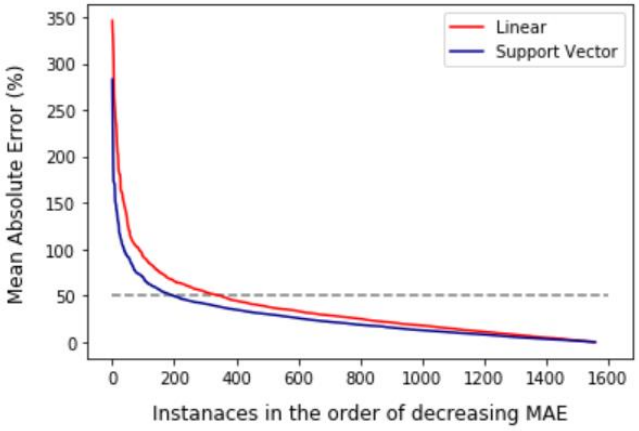

Figure 7. MAE of Linear and Support Vector Regression

Subsequently, we have decided our optimal algorithm among the six models analyzed to be Support Vector Regression.

D. Model Specification - Hyperparameters

From the previous section, we have decided on the optimal algorithm and features. Now, we proceed to increase the performance of our SVR through hyperparameter tuning. According to scikit-learn, C in SVR represents the regularization parameter. "The strength of the regularization is inversely proportional to C. Must be strictly positive. The penalty is a squared l2 penalty"[7]. In addition, scikit-learn provides various kernel functions as options ('rbf' (Gaussian), 'linear', 'poly') which are "used to map the original dataset (linear/nonlinear) into a higher dimensional space with a view to making it linear dataset"[8]. The performance results for each combination of C and kernel type, with all the other options not changing, are presented below.

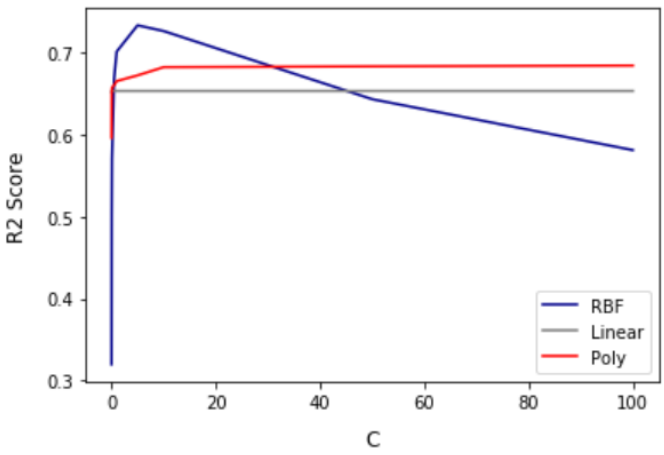

Figure 8. Tuning Hyperparameter C and Kernel

The best combination of all was when C=5 and kernel type is Gaussian (R^2 = 0.733). The amount of increase in R^2 score compared to our initial model seems marginal, but at least we have confirmed that this combination produces the best result.

E. Final Evaluation

- **Generalizability**

As a final stage, we performed cross-validation to confirm whether our model performs equally well on the unknown test set and hence generalizable. This is to reduce bias as we are using most of the data for training the model; moreover, it also reduces variance as we are making use of various training

[7] *"Sklearn.svm.SVR." Scikit, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html.*

[8] *Diffference between SVM Linear, Polynmial and RBF Kernel.* https://www.researchgate.net/post/Diffference_between_SVM_Linear_polynmial_and_RBF_kernel.

data set, decreasing the effects of outliers being included in a specific training set. Our final model to use for k-fold validation is:

regressor = SVR(kernel='rbf', C=5)

The resulting average score for k=5 is 0.60, and for k=10 is 0.62.

- **Analyzing Outliers:**

a) Error Rate of 282.82%

· # Baths:	1
· # Beds:	2
· Floor:	0
· Leased Land:	No
· Building Category:	Co-op
· Year Built:	Post 2000
· # Complaints:	6
· # Felonies:	47
· Distance to Subway:	0.029
· Distance to Park:	0.308
· Avg Sales Volume:	1132
· Avg Selling Price:	1,461,170
· Sale Price (true):	390,000
· Sale Price (prediction):	1,677,560

Considering the most important factors, 'baths' and 'bed', this house should be sold around $990,881 which is the mean price of the housings with 2 bedrooms and 1 bathroom. The actual transaction price of $390,000 is doubtfully low, and our predicted price of $1,677,560 seems more reasonable.

b) Error Rate of 216.50%

· # Baths:	4
· # Beds:	5
· Floor:	0
· Leased Land:	No
· Building Category:	Others
· Year Built:	Pre 1980
· # Complaints:	8
· # Felonies:	0
· Distance to Subway:	0.23
· Distance to Park:	0.444
· Avg Sales Volume:	928
· Avg Selling Price:	7,090,579
· Sale Price (true):	1,825,000
· Sale Price (prediction):	7,669,993

The 'average selling price' here is exceptionally high which seems to have a huge impact on the predicted price. The feature was calculated from the prices of housings with the same Zipcode and the number of bedrooms. However, most of the instances in our dataset has 0 (studio) to 3 bedrooms, with only 0.05% instances having more than 4 bedrooms. That is, the average selling price for this instance is highly sensitive to outliers and thus not reliable. If we were able to collect more data on houses with more than 4 bedrooms, our prediction would have been less error-prone.

V. DEPLOYMENT

We assume that there exists a strong demand for our model from all the parties involved in the residential property transactions. Once the business deploys our model and starts providing its customers with approximate value of the property, the business can draw even more customer traffic to their platform which would directly be translated into the boost in sales. The more proprietary data the business owns for each property, there exists more potential for significant improvement in the model. Business can provide a "free" and "paid" version of APIs by differentiating features included in each API (i.e. features based on publicly available data vs. features based on exclusive data).

Furthermore, regulators could also leverage our model to probe the current status of market and assess the impact of any new real estate regulations by training the model with hypothetical data.

Sections following will elaborate on actual deployment of model which is a crucial step to realize values out of such business cases.

A. Platform

The most suited platform to deploy this model would be real estate database companies such as Zillow Group. After its acquisition of StreetEasy in 2013, Zillow group is deemed to have the most extensive database on NYC's residential properties. Currently, Zillow's web UI provides users so-called "Zestimate" which is a reference point for a predicted FMV of the property. Our model could be considered as an enhancement to the prediction of "Zestimate Model". Also, current "Zestimate" is only available for existing listings on Zillow. We would like to further improve the UI so that users can pass in hypothetical features (i.e. Number of Bedrooms, ZipCode, Distance to the nearest park, etc.) and get the price prediction for the property with such features which could be the reference price for users' property purchase/sale endeavors.

B. Infrastructure and System Process

- **Database**

a. Master Property Identifier DB

The company needs to have all of its property inventory to be assigned with proper unique IDs, so

that adding/deleting and modifying the features are easily manageable.

b. Feature DB

Each instance should be stored as a separate row with a unique ID attached from above. Since the table is going to be heavily queried, it needs to be properly indexed and is recommended to have frequent caching in the output pipelines.

c. Predicted Value DB

For each unit with distinct ID, we should have a table that stores the current estimated value (updated daily) based on our model. We also should have a history table which is copied from current estimated value table with run ids so that we do not lose our predictions once the current table is updated. The current table is the table which user will be querying directly against for any housing units that exist in the master DB.

Updates to Property ID, Feature table, and Predicted Value table should occur overnight so that the user is less impacted by the DB update. (In addition, DB output pipelines should not have NOLOCK as part of their DB query).

- **Code Deployment**

The model should be deployed into a container where each image is tagged and built separately. This will provide the company functionality to control the version of code that gets deployed and ability to revert when necessary.

- **Development Environment**

All DBs and containers should have DEV/STG/QA/PROD environment. Initial development will be done in DEV and will be deployed to STG for testing. STG DB should be synced with PROD DB once a week so that the developers would have actual data for testing and debugging. QA environment is exclusively for QA and would have the latest copies of both code and DB of PROD. QA process will be elaborated in the latter sections.

C. Model and Data Governance

There are fairly straight forward methods to evaluate the performance of our model. NYC Department of Finance updates the property transaction data every month. We can simply compare our model's prediction to the actual transaction data and measure how accurate our

predictions are. Once we observe any data points which significantly disrupt our model's prediction, we could conduct more extensive exploratory analysis on such data points to extract features that affects the prediction of our models.

Also, Zillow provides a channel where the owner of the property can send in request for corrections of his/her property's "Zestimate" with supporting arguments. The company can compile such requests and analyze them periodically to come up with actions to remediate the discrepancies if necessary (i.e. add more features such as whether the kitchen is renovated or not).

Furthermore, thorough QA processes should be introduced to govern the integrity of the data. As discussed above, extensive QA should be conducted to spot outliers and such outliers should be looked into. Also, since all the units in master inventory are assigned unique IDs, we should monitor any changes in the feature of the existing unit (from the new listings) and should be carefully verified (i.e. previous sales posting indicates the unit had 3 bedrooms but the current active listing says the unit has 4 bedrooms). Such cases should be manually probed to verify that the change is actual (i.e. owner has done renovation) or a manual error by the listing agent or the owner.

For the governance of the significant model changes, we should avoid selecting more randomized or complicated models over simpler models simply because the evaluation metrics are marginally better. Most of the features used in the model is numerical rather than categorical and we expect for such numerical (ordinal) features to have somewhat monotonic relationship with the sales price. For example, as the number of bedrooms increases, we expect the sale price will concurrently increase (while the marginal price increase per unit of bedroom may vary). Such result is displayed in the comparison between polynomial regressions with different degrees in previous sections. Figure 2 shows that polynomial regression with degree 3 had higher R-squared score on training data compared to that of degree 2 but had significantly lower score in testing data. However, we would like to avoid the situation where selection bias of training and testing dataset affects such evaluation metrics on testing

data and making the premature decision to deploy more complicated model to production.

D. Limitations of the model

The limitation of the current model is that the model cannot capture the individual context of a sale. It would be very challenging to collect data on such features which may have meaningful impacts on the final sale price. Such features may include whether the seller is under severe financial stress; potential buyers competition unreasonably raising the bid, etc. Hence, the usage of the model should be limited to serving the purpose of providing the reference price under "normal" transactional situations without such context.

Also, extreme market conditions could severely affect the outcome of our model such as financial crisis of 2008. Such extreme cases should be identified and should not be implemented in the model. If deemed necessary, we should build a separate model for stress scenarios.

Since the model is limited to predicting the "market value" of the property under "normal" conditions, we need to make sure that the training data does not include any "abnormal" instances. We have observed many unreasonable transactional data from the Department of Finance databases. Since the database records all transactions without distinction, it is difficult to distinguish "market transaction" vs "non-market transaction". An example of non-market transaction would be inheritance of property in the form of transactions. We have looked into some cases where sale price was 0 or significantly low and the actual filings indicated that the seller and the buyer shared the same last name (which is an indication for inheritance). Before updating the features and training the model, we should carefully assess any additional incoming data and examine so that such data does not corrupt the training of the model.

E. Ethical Considerations

The company should be wary of privacy issues regarding user data. The target variable and some of the features are public records but others are not. The company needs to make sure that the user understands and agrees to data usage clause in the service agreement.

Also, the company should carefully consider the potential social issues related to the model when

deploying. One of the closely related social issues under is NIMBY-ism which is a social phenomenon which the residents in a certain neighborhood opposing any changes which would negatively impact the value of the properties and the quality of their life. According to Been, "Americans of nearly all sociodemographic or socioeconomic statuses" are "now half as likely to have moved in [2017] as their counterpart in 1950" [9]. If we engineer features that are known to be linked with NIMBY-ism, we may get more accurate predictions but might have negative social impacts such as widening the socioeconomic gap between different groups of people[10].

VI. CONCLUSION

The goal of the model is to provide a service which could help people when they are making one of the most important financial decisions of their life. Potential buyers or sellers could leverage our model and use the predicted price as the reference point in the initial stage of their long and painful journey of home purchase or sale.

Our model at current stage, given the restrictions of available data, does not satisfy our standard to be productionized as is; however, we do see ample room for improvement once proprietary data are used for further feature engineering and training of the model. The examples of proprietary data which we expect to improve the performance of the model include monthly maintenance, renovation status and exact square-footage and etc. Even with such enhancement, our model will potential limitations and ethical considerations as previously discussed. Yet, we believe that such issues could be addressed and overcome or remediated through iterations of collaborations between domain experts and data scientists.

All of the codes used in our paper can be found in our Github repository

[9] Been, Vicky, "City NIMBYs", *Journal of Land Use, vol 332*, (Spring 2018), p.236. https://furmancenter.org/files/City_NIMBYs_Vicki_Been_JLUELv33.2.pdf

[10] Been, Vicky, "City NIMBYs", *Journal of Land Use, vol 332*, (Spring 2018), p.231. https://furmancenter.org/files/City_NIMBYs_Vicki_Been_JLUELv33.2.pdf

REFERENCE

1. McDonald, Emily. "Manhattan Homes Linger on Market, Forcing Sellers to Cut Prices." *StreetEasy*, 21 Mar. 2019, https://streeteasy.com/blog/february-2019-market-reports/.
2. "Frequently Asked Questions." *Scikit*, https://scikit-learn.org/stable/faq.html.
3. Drakos, Georgios. "How to Select the Right Evaluation Metric for Machine Learning Models: Part 1 Regression Metrics." *Medium*, Towards Data Science, 5 Dec. 2018, https://towardsdatascience.com/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-1-regrression-metrics-3606e25beae0.
4. Drakos, Georgios. "Decision Tree Regressor Explained in Depth." *GDCoder*, GDCoder, 2 June 2019, https://gdcoder.com/decision-tree-regressor-explained-in-depth/.
5. Krishni. "A Beginners Guide to Random Forest Regression." *Medium*, Data Driven Investor, 5 June 2019, https://medium.com/datadriveninvestor/random-forest-regression-9871bc9a25eb.
6. K. Roy, S. Kar, R. N. Das, "Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment", pp. 227, 2015.
7. *"Sklearn.svm.SVR." Scikit, https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html.*
8. *Diffference between SVM Linear, Polynmial and RBF Kernel.* https://www.researchgate.net/post/Diffference_between_SVM_Linear_polynmial_and_RBF_kernel.
9. Been, Vicky, "City NIMBYs", *Journal of Land Use, vol 332*, (Spring 2018), p.236. https://furmancenter.org/files/City_NIMBYs_Vicki_Been_JLUELv33.2.pdf
10. Been, Vicky, "City NIMBYs", *Journal of Land Use, vol 332*, (Spring 2018), p.231. https://furmancenter.org/files/City_NIMBYs_Vicki_Been_JLUELv33.2.pdf

CONTRIBUTION

Young Jin Park:
Data cleaning
Feature extraction (Google Map API, Geopy)
Feature engineering

Soojin Kim:
Data preparation (Zillow Web Scraping)
Modeling & Evaluation
(Linear, Polynomial Regression)
Final write-up formatting

Sujeong Cha:
Data preparation (Socrata API)
Modeling & Evaluation
(Decision Tree, Random Forest, SVR)
Hyperparameter Tuning