Data Science for Business Analytics

NYC Real Estate Price Prediction

Spring 2020 Project Report

Shashank Rajiv Lochan (srl506)

Xiaoxuan Wang (xw2098)

Xingyu Liu (xl2332)

12.05.2020

Introduction

Real estate is the land, all of the natural parts of the land such as trees and water, and all permanently attached improvements such as fences and buildings. People use real estate for a wide variety of purposes, including retailing, offices, manufacturing, housing, ranching, farming, recreation, worship, and entertainment. The success or failure of these uses is dependent on many interrelated factors: economic conditions, demographics, transportation, management expertise, government regulations and tax policy, climate, and topography. The objective of those engaged in the real estate industry is to create value by developing land or land with attached structures to buy, sell, or to lease or by marketing real estate parcels and interests.

The island of Manhattan, whose current land value is estimated at $1.4 trillion, was first purchased for $24. The long and rich history of New York City's real estate market begins in 1609 with its first European settlement. In 1626, the Dutch purchased the island of Manhattan, then known to the Dutch as New Amsterdam, from the indigenous Lenape tribe for 60 guilders' worth of beads and buttons — famously estimated at about $24 in modern currency.

Savills World Research recently conducted a study showing that by the end of 2018, the global value of real estate had reached almost $300 trillion. This makes real estate the largest asset class globally, with a value 3.5 times larger than the entire global GDP.

With so much value tied up in global real estate, understanding property value and the underlying metrics that determine its value is paramount. Investors, financial institutions, and governments who manage their real estate portfolios wisely can generate significant returns for their shareholders and the broader community of people whose livelihoods depend on real estate such as builders, operators, and agents. Valuing a property in a way that aligns with the underlying fundamentals of the income and growth it can generate is essential to reducing risk.

Stakeholders need to understand property value and its underlying drivers. But the drivers of volatility in property value often come from traditional and unconventional data, that in today's time are voluminous for analysts to understand and uncover interesting patterns and aid the decision-making process. Simply put, for all stakeholders the end goal is to maximize returns and in order to do so, one must execute the transaction to buy or sell effectively at the right price to realize profits.

Currently, the property sales market in New York City is massive. To purchase a property in New York City, there is a humongous amount of information to search for and it is very easy to get lost in the ocean of information. From the supply perspective, the real estate company needs to set the property price based on features like the neighborhood, location, and room space, etc.

Only when they have a reasonable and competitive price, they can have a successful sale of the property. The accurate predictions would help to increase sales and in turn generate profits. From any investors' perspective, they need to know the price in advance, so that they can decide whether this project is worthy of any investment and how long they could earn their money back. To sum, this prediction model can help the investors estimate the return on investment on the project in advance.

Taking this as our motivation we decided to apply the principles and techniques of Data Science to predict the property price to aid the data-driven decision making and to develop a deeper understanding of the factors influencing property values.

Implications of Data Science in Real Estate

Undertaking data science and its principles can prove to be a strategic advantage for Real Estate companies to help determine if a property is a good match in terms of investment for each customer.

The real estate sector is a sector whose reach is vast, it is therefore affected by many social, political, and economic factors which means that there is a huge amount of complex data available and it is through analyzing and understanding this data that models can be created which aim to replicate the changes in the sector and evolve in order to anticipate what we might see in the future.

We believe it is essential to establish a predictive model that may take all perspectives of the house sales transaction process into consideration. For a typical property sales transaction, we believe the effects that may influence the price are the features of the sales volume of certain communities were the property's location and the property's features. For the sales volume, since we have New York Cities' past 10 years of property sales data, we will develop predictive models based on the information we have to predict the sales for each borough. We believe it is essential for the people who are going to sell their properties or for the brokers who are selling the properties on behalf of someone to have insights about the sales trends in the communities where their properties are located. With this information, people can list their property with optimally reasonable prices and in turn, increase their probability of selling their property. The property's features are the other group of features which may directly affect the sales price. We are going to establish a predictive model, which will take all property features and the predicted sales volume from the model described above to make the final sales price prediction.

Data Understanding

The success of data science in real estate is however dependent on the quality of the data and how it can be accessed. The volume available is huge, sources are multiple and not always in line with one another. A huge amount of data sits in paper files or PDF documents, so it's not easy to access it even though very relevant. A key challenge during these first stages of data science in real estate is, therefore, understanding how to structure and qualify data in order to eliminate non-relevant or unreliable sources and focus on dependable data that is available for use.

For the purpose of the project we have leveraged the Annualized NYC rolling sales data which is a free, open and downloadable archive of Sale prices of properties in NYC from 2003 through 2015. The features include neighborhood, building type, square footage, and other relevant data. Because rarely is there an exact match with the problem. Historical data often are collected for purposes unrelated to the current business problem, or for no explicit purpose at all. From different dataset options that we came across, we decided to use the above mentioned as it provided enough instances over a period of time for our modelling phase to capture the underlying regularities or trends and also using our intuition for this initial phase that features like square feet, neighborhood, property type etc. could be good predictors of the price of the property.

In total the dataset comprises 1264162 instances and 21 attributes values for each instance as can be seen from the following data description:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1264162 entries, 0 to 1264161
Data columns (total 21 columns):
 #   Column                          Non-Null Count    Dtype
---  ------                          --------------    -----
 0   BOROUGH                         1264162 non-null  int64
 1   NEIGHBORHOOD                    1264162 non-null  object
 2   BUILDING CLASS CATEGORY         1262683 non-null  object
 3   TAX CLASS AT PRESENT            1246996 non-null  object
 4   BLOCK                           1264162 non-null  int64
 5   LOT                             1264162 non-null  int64
 6   EASE-MENT                       10 non-null       object
 7   BUILDING CLASS AT PRESENT       1246996 non-null  object
 8   ADDRESS                         1264155 non-null  object
 9   APARTMENT NUMBER                292906 non-null   object
 10  ZIP CODE                        1264162 non-null  int64
 11  RESIDENTIAL UNITS               1264162 non-null  object
 12  COMMERCIAL UNITS                1264162 non-null  object
 13  TOTAL UNITS                     1264162 non-null  object
 14  LAND SQUARE FEET                1264162 non-null  object
 15  GROSS SQUARE FEET               1264162 non-null  object
 16  YEAR BUILT                      1264162 non-null  int64
 17  TAX CLASS AT TIME OF SALE       1264162 non-null  int64
 18  BUILDING CLASS AT TIME OF SALE  1264162 non-null  object
 19  SALE PRICE                      1264162 non-null  object
 20  SALE DATE                       1264162 non-null  object
dtypes: int64(6), object(15)
```

table1. Data Description 1

We are aware that this is simply based on our intuition and as we progress we will use the power of data mining to uncover the correlations of the different features in the dataset with price which is our class label. Although we incurred no cost for obtaining this data as this was sufficient for the analysis we are performing, we believe there can be various other data sources and features that can influence the price of the property that will often require investment in data that in turn can place an organization with strategic advantage in the market. Empirical studies and experiments have proved empirically that investment in data acquisition leads to informative gains and thus we propose for exhaustive data science, organizations should consider investing in interesting sources of data that has the potential to influence the price of any property.

Time Series Analysis

Consider the possible seasonal impacts or other time-related impacts on the sales of the house. We decided to conduct a time-series analysis. To find out the relation between time and the number of houses sold. We analyzed the time-series of sales. After grouping the sales by date, we got a new data frame including the sale date and the number of houses sold.

SALE DATE	num
2003-01-01	12
2003-01-02	78
2003-01-03	120
2003-01-04	104
2003-01-05	135

table 2. data frame

First, we plot the plots of time and number.

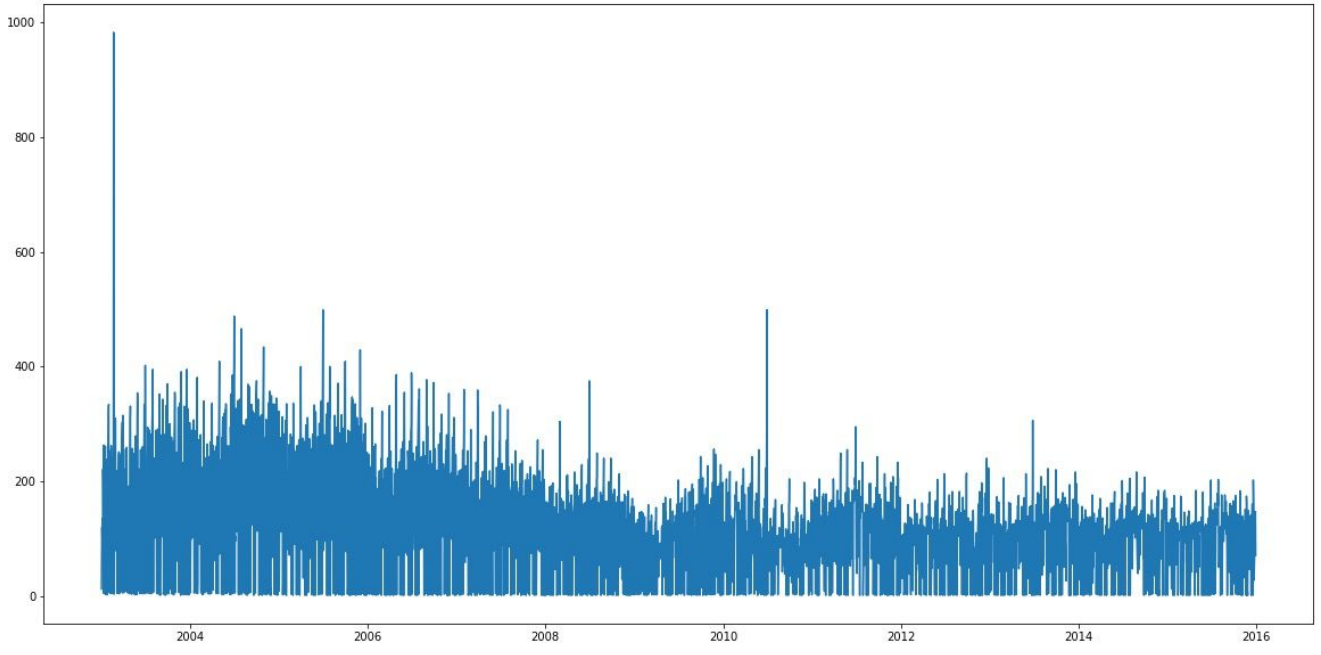

figure 1. time series

It shows some seasonal fluctuation here. And the sales of the house are decreasing generally from 2004 to 2010, but stay relatively constant from 2010 to 2016.

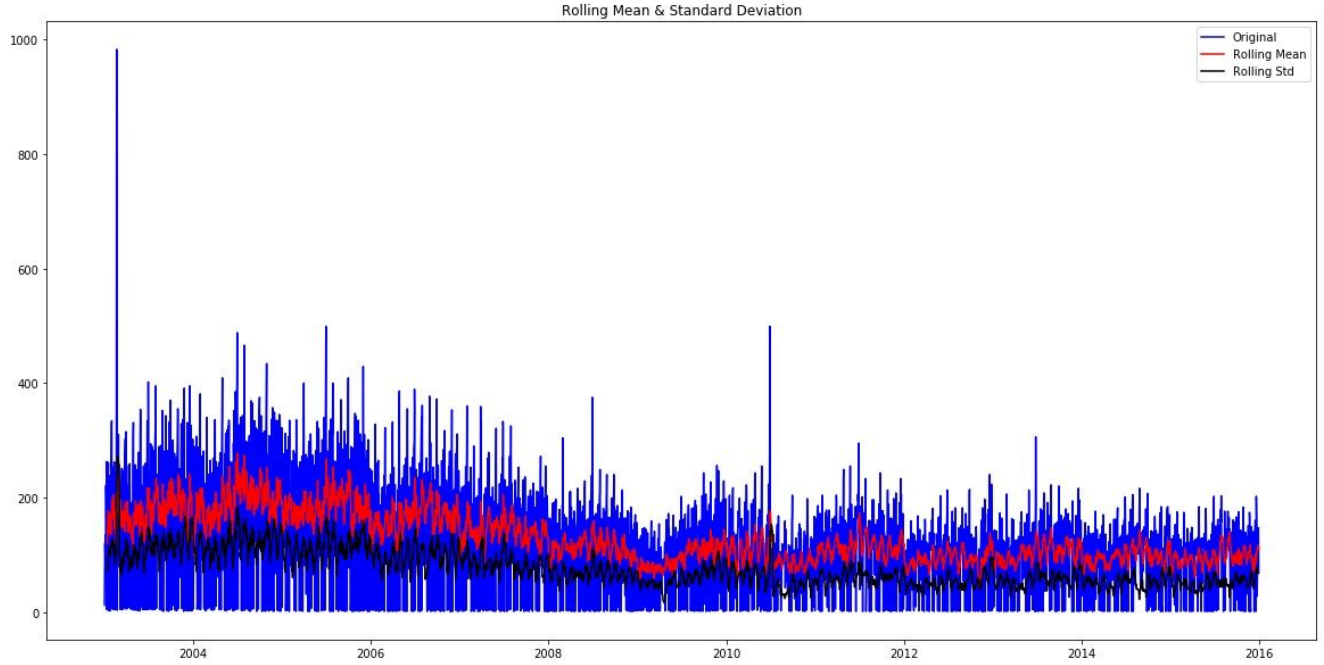

figure 2. time series with rolling statistics

To examine the stability of the time-series, we used the rolling statistics of Mean Difference and Standard Deviation. In this way, we can see the change of the two indicators.

After plotting, we can see that from 2004 to 2010, the sales were decreasing. The mean difference and standard deviation are decreasing over time too. From 2010 to 2016, when the sales were relatively stable. the rolling statistics are more constant too. In general, the time-series is lightly stable.

Data Preparation

Seldom do we find data to be available in the feature vector form that is required as an input for the multitude of data mining algorithms that organizations wish to exploit in order to practice informed decision making for realizing gains. An important stage in the data mining process is the data preparation phase which we believe is analogous to relentless preparation can lead a man to succeed in his endeavors and for the data mining process this couldn't be more true. Majority of successful data mining projects spend most of their time in understanding and preparing the data in a cyclic manner to understand the structure and with that the following steps that must be taken accordingly to ensure that the results of the data mining process actually achieve the business goals. In order to ensure the task of price prediction reaches its

goal of providing the most accurate estimate of the price, we first loaded the dataset from its raw format in the tabular format where every row constitutes a feature vector of the instance and every column defines the feature value and undertook the following data preparation tasks from a dual perspective of comprehensibility and modelling:

A. Data Cleaning

1. Edit Column Name: The data imported from its raw format(csv), contains meaningless column names such as the unnamed_1 that only leads to ambiguity. We changed those column names into understandable and comprehensible ones.
2. We found that most of the columns were of the type Object that does less to really distinguish the type of data that it hosts. We refactored the Column Contents into Numeric Ones for our price prediction which is of numeric nature. The columns, such as land square feet, gross square feet and sales price that are supposed to be numeric values were accordingly refactored and transformed in order to pass those data into our modelling phase. Also, illegal characters inside such as ? and $ were duly removed as they do not contribute to the data mining process and transformed columns to be int or float accordingly.

```
df['SALE PRICE'].value_counts()
```

```
$0                324374
0                  26498
$-                 18667
$10                 7804
? -                 6418
                   ...
$262,900               1
$510,920               1
$1,05,06,000           1
$1,19,250              1
$2,013,868             1
Name: SALE PRICE, Length: 98089, dtype: int64
```

3. Reset the Type of Columns: just as mentioned in the above step, some object features are actually numeric ones. Those should be reset to be int or float types. Apart from that, some columns of int or float type are categorical ones. Taking borough for example, in our dataset, the number 1 in borough features represents Manhattan district in New York City, which means those numbers from 1 to 5 are categorical data type instead of numeric one. Therefore, we should transform those columns to be objective ones.

```
 #   Column                          Non-Null Count    Dtype
---  ------                          --------------    -----
 0   BOROUGH                         1264162 non-null  int64
 1   NEIGHBORHOOD                    1264162 non-null  object
 2   BUILDING CLASS CATEGORY         1262683 non-null  object
 3   TAX CLASS AT PRESENT            1246996 non-null  object
 4   BLOCK                           1264162 non-null  int64
 5   LOT                             1264162 non-null  int64
 6   EASE-MENT                       10 non-null       object
 7   BUILDING CLASS AT PRESENT       1246996 non-null  object
 8   ADDRESS                         1264155 non-null  object
 9   APARTMENT NUMBER                292906 non-null   object
 10  ZIP CODE                        1264162 non-null  int64
 11  RESIDENTIAL UNITS               1264162 non-null  object
 12  COMMERCIAL UNITS                1264162 non-null  object
 13  TOTAL UNITS                     1264162 non-null  object
 14  LAND SQUARE FEET                1264162 non-null  object
 15  GROSS SQUARE FEET               1264162 non-null  object
 16  YEAR BUILT                      1264162 non-null  int64
 17  TAX CLASS AT TIME OF SALE       1264162 non-null  int64
 18  BUILDING CLASS AT TIME OF SALE  1264162 non-null  object
 19  SALE PRICE                      1264162 non-null  object
 20  SALE DATE                       1264162 non-null  object
dtypes: int64(6), object(15)
memory usage: 202.5+ MB
```

table 3. Data Description 2

Initial data types of features that were found to be not in the right format and hence were refactored and transformed. Following figure shows the output of this step.

#	Column	Non-Null Count	Dtype
0	BOROUGH	1264162 non-null	object
1	NEIGHBORHOOD	1264162 non-null	object
2	BUILDING CLASS CATEGORY	1262683 non-null	object
3	TAX CLASS AT PRESENT	1246996 non-null	object
4	BLOCK	1264162 non-null	int64
5	LOT	1264162 non-null	int64
6	EASE-MENT	10 non-null	object
7	BUILDING CLASS AT PRESENT	1246996 non-null	object
8	ADDRESS	1264155 non-null	object
9	APARTMENT NUMBER	292906 non-null	object
10	ZIP CODE	1264162 non-null	int64
11	RESIDENTIAL UNITS	1264162 non-null	float64
12	COMMERCIAL UNITS	1264162 non-null	float64
13	TOTAL UNITS	1264162 non-null	float64
14	LAND SQUARE FEET	1264162 non-null	int32
15	GROSS SQUARE FEET	1264162 non-null	int32
16	YEAR BUILT	1264162 non-null	int32
17	TAX CLASS AT TIME OF SALE	1264162 non-null	int32
18	BUILDING CLASS AT TIME OF SALE	1264162 non-null	object
19	SALE PRICE	1239077 non-null	float64
20	SALE DATE	1264162 non-null	datetime64[ns]

dtypes: datetime64[ns](1), float64(4), int32(4), int64(3), object(9)

table 4. Data Description 3

4. Remove Na values and Unnecessary Columns: Most real world data is noisy and is prone to include values that do not aid the data mining process and it becomes necessary to eliminate these in order to ensure our model is learning from the informative features and their values in the data. Removing Na values is a standard process. As for unnecessary columns, features such as address and apartment units have no contribution to influence sales price. Besides, those columns may function as noise while we are fitting models. Thus removing such columns and irrelevant values becomes important in the data pre-processing stage.
5. Filtering data values: Firstly, some values are illegal. For example, we are dealing with sales data in new york city. Hence, realistically Zip Code should range from 10000 to 20000. Values beyond that range should not be included in our dataset as they are not true representations of the real world data that the model may be used in the future to make predictions. Secondly, although the remaining values may be attributed as legal, their occurrence is so rare that it can be ignored to avoid skewness in the data. And

hence the removal of such values will improve prediction performance due to lack of noise. To be specific, a sales price under 100k is a rare occurrence and not likely to be a regular event in New York City.

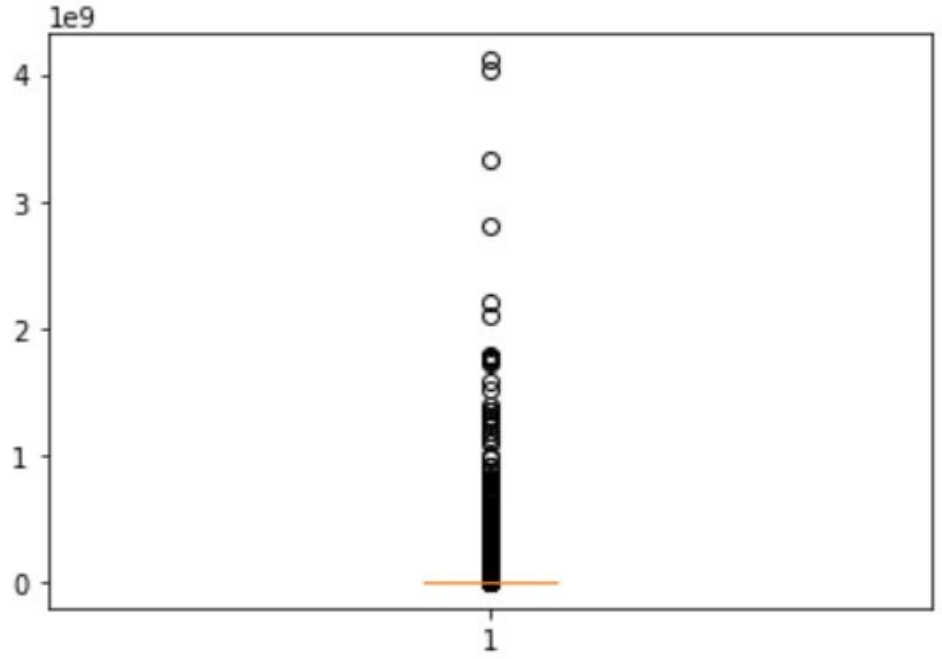

figure2.. Sales price distribution

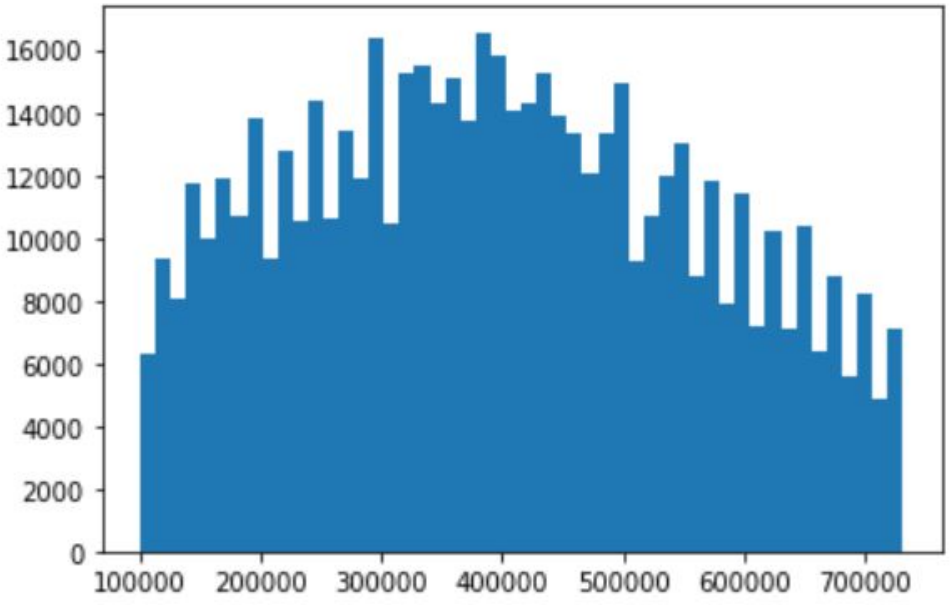

figure3. Sales price after data cleaning

6. Data Description:

Following is the sequence of exploratory data analysis that was performed on the dataset that provide us with the understanding of the different features that could prove to be influential in predicting the price of the property and the most informative amongst them would be found as we proceed with our data mining phases.

Categorical Data:

	BOROUGH	NEIGHBORHOOD	BUILDING CLASS CATEGORY	TAX CLASS AT PRESENT	BUILDING CLASS AT PRESENT	ZIP CODE	BUILDING CLASS AT TIME OF SALE
count	423336	423336	423336	423336	423336	423336	423336
unique	5	257	53	11	157	214	236
top	4	FLUSHING-NORTH	10 COOPS - ELEVATOR APARTMENTS	1	D4	11375	D4
freq	164606	13797	103269	199802	102660	9822	102356

table 5. Categorical Data

Borough Sales Distribution:

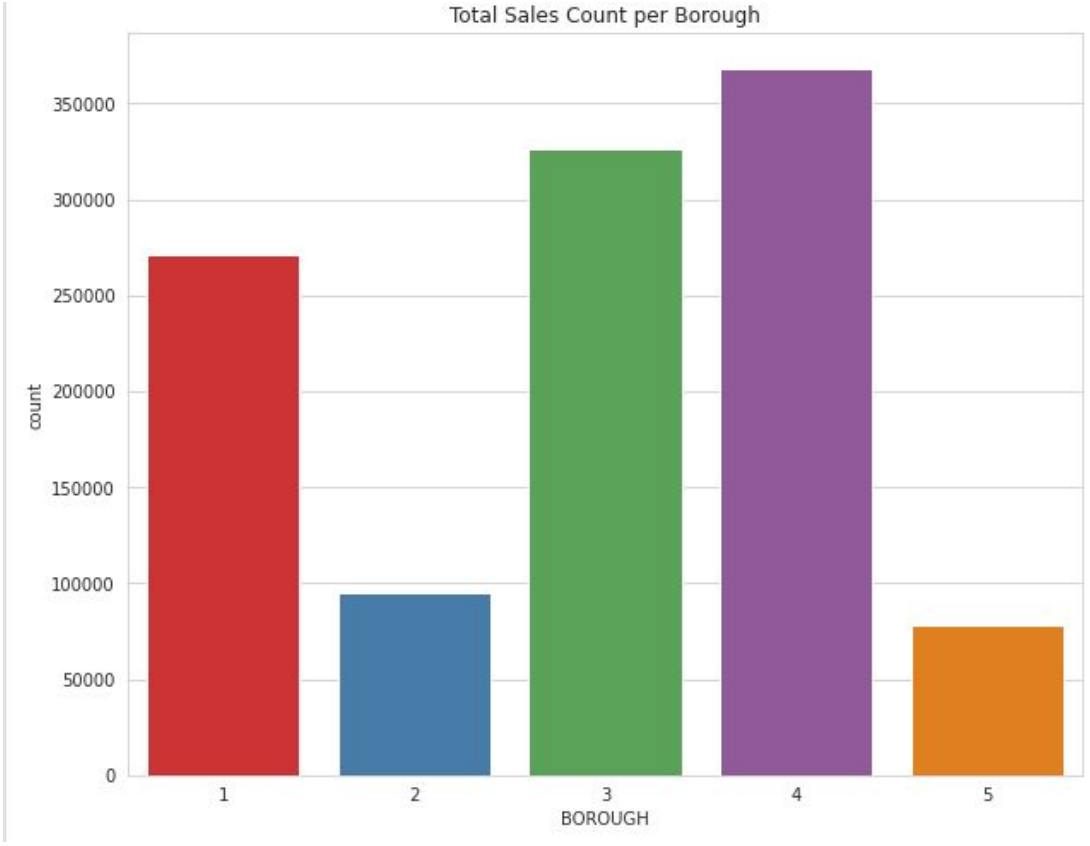

table 6. Borough Sales Distribution

Top 20 costliest neighborhoods:

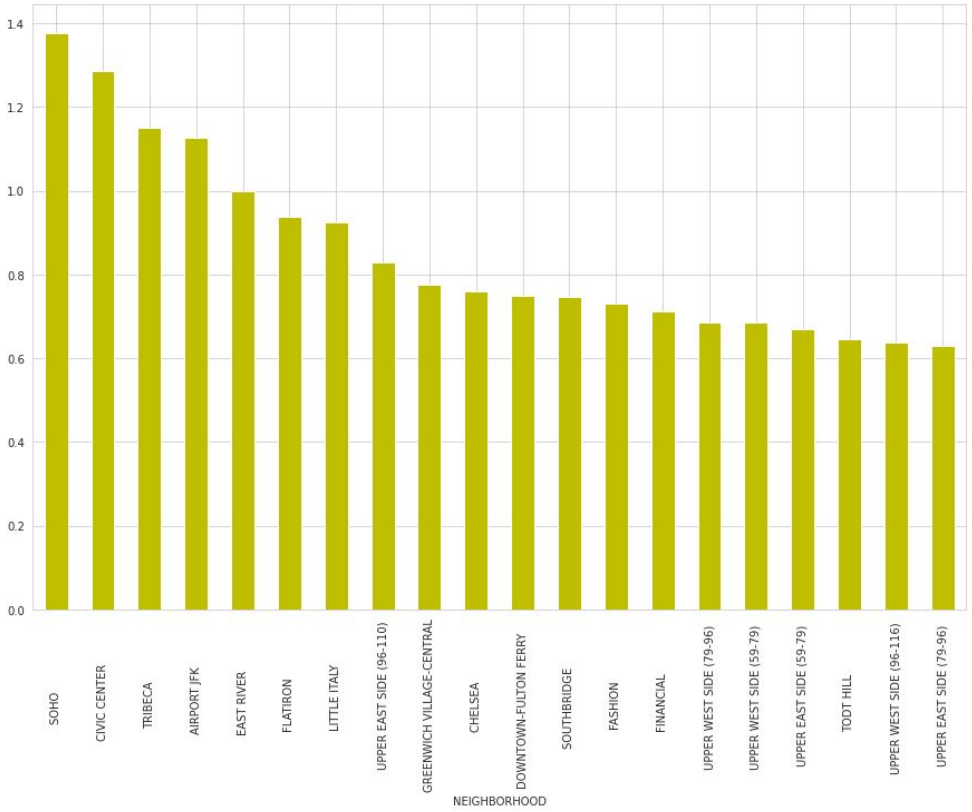

Table 7. Costliest Neighborhood

Tax class at present Distribution:

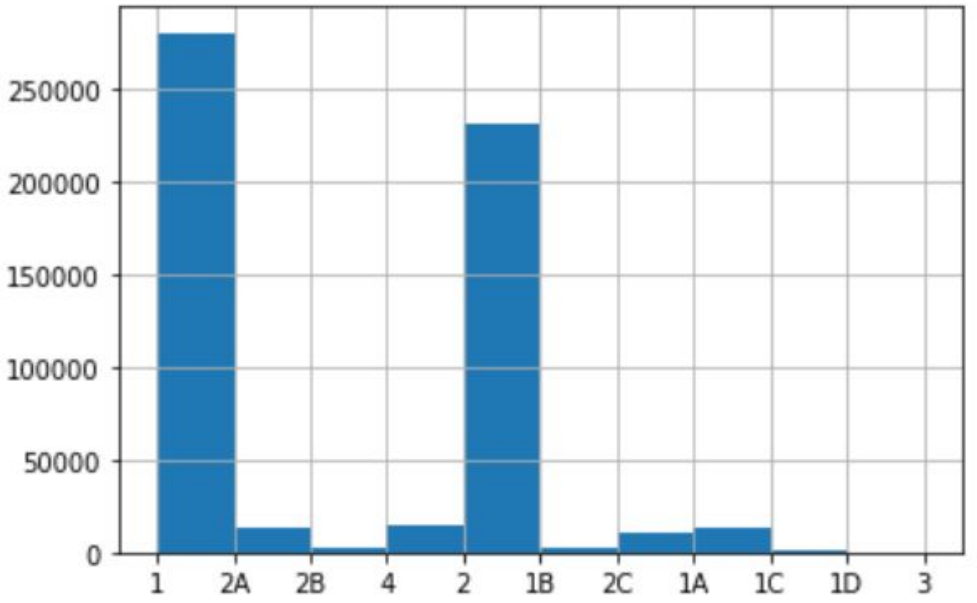

table 8. Tax class at present Distribution

Sales price grouped by Tax class at present:

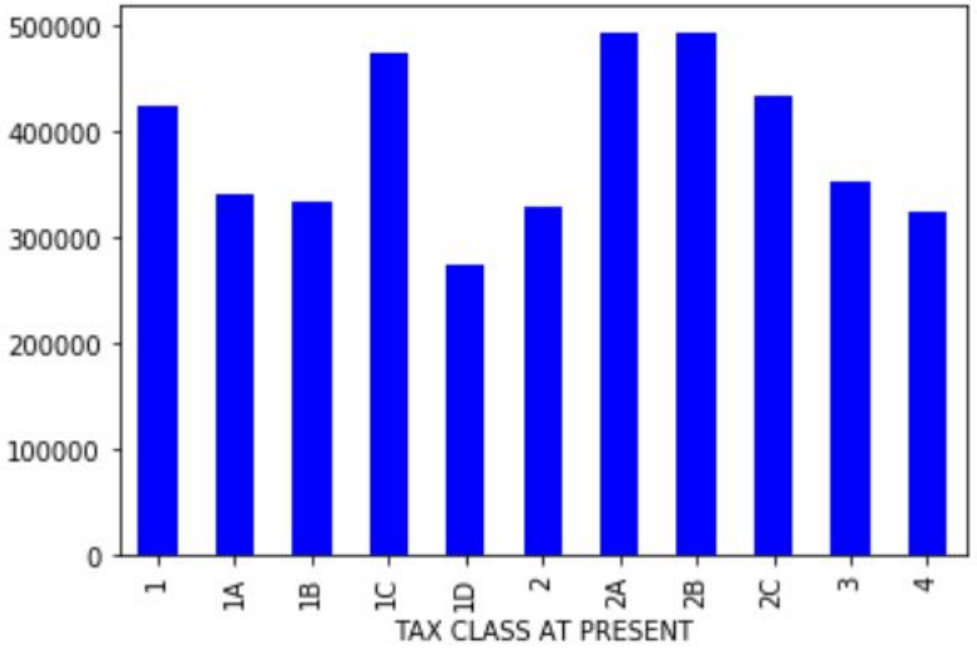

table 9. Sales price grouped by Tax class at present

Sales price grouped by Tax class at time of sale:

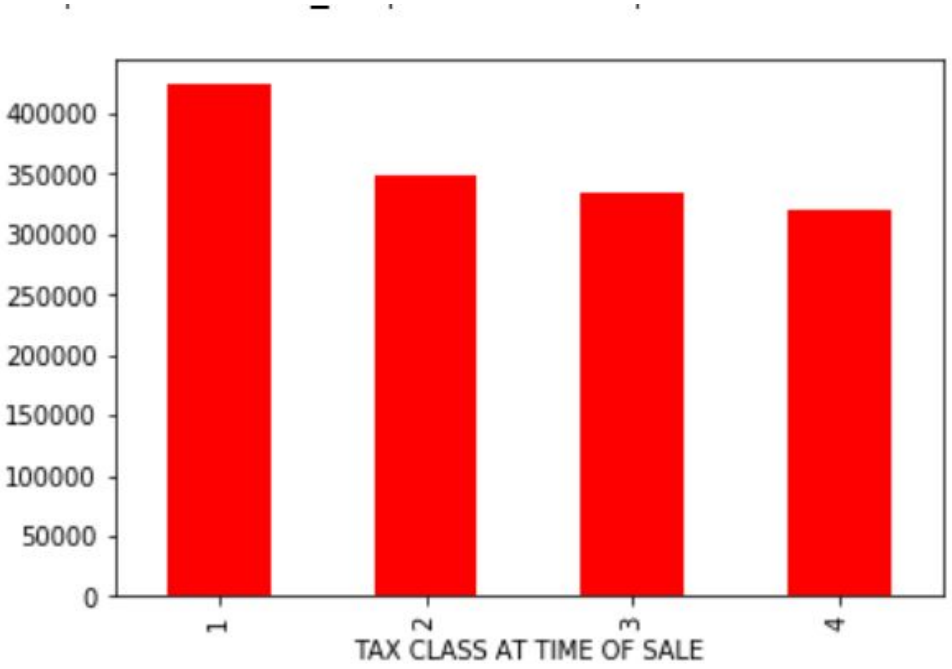

table 10. Sales price grouped by Tax class at time of sale

Sales price grouped by Borough:

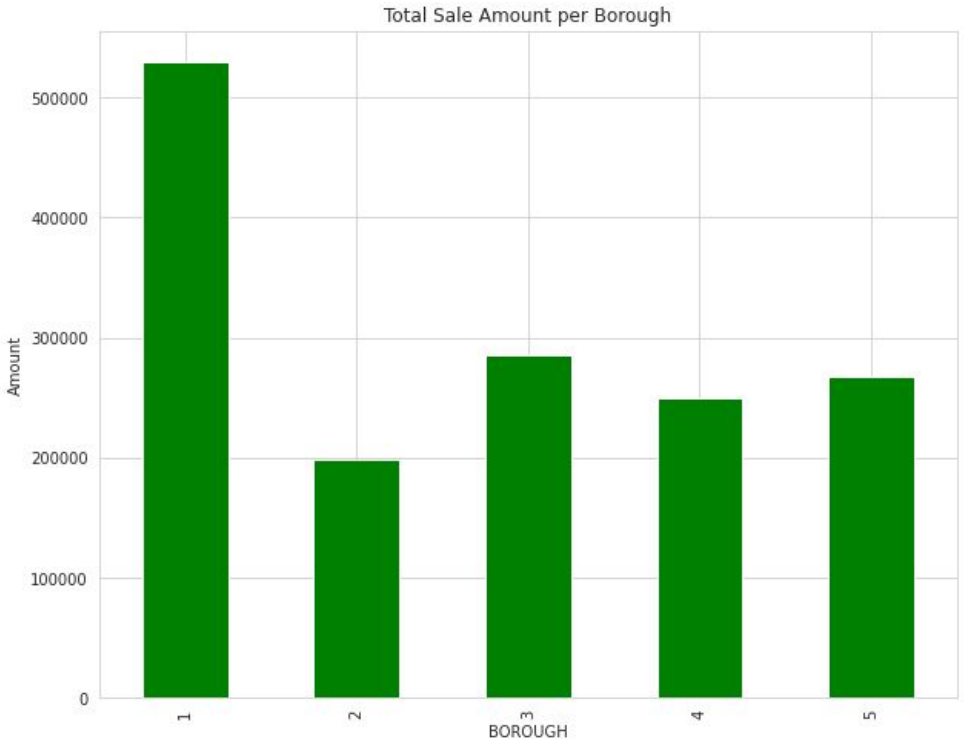

table 11. Sales price grouped by Borough

Sales price grouped by Zip Code:

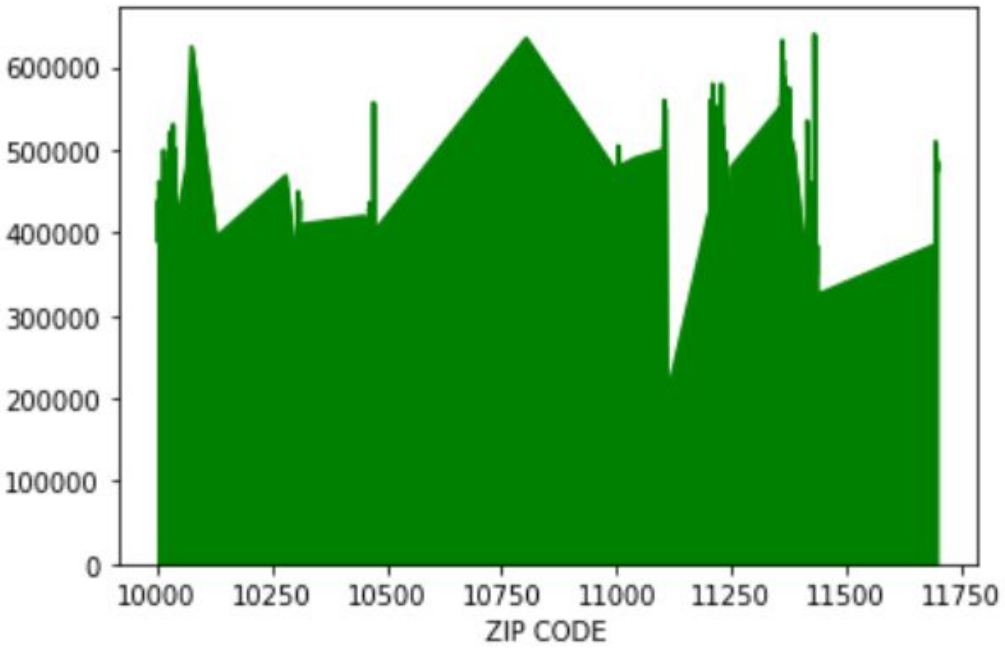

table 12. Sales price grouped by Zip Code

After initial data cleaning, our dataset consisted of 300k pieces of data with 15 features in total. In order to pass our data into the data mining models, more data preparation needs to be carried out as follows:

B. Model Fitting Preparation:

One-Hot Encoding: We notice that objective features are not legal inputs of our built models. In order to solve that problem, we need to do one-hot encoding, a technique to get dummy nodes. For example, we gained dummy nodes from the initial Zip Code feature. The reason we deal with the column is that sales prices in separate regions of NYC may differ a lot. Taking Manhattan for example, a studio in that area may cost around 900k dollars. However, that is as much as the price of a house with over eight bedrooms in the Bronx. So we divide sales data up based on geography (specifically, Zip Code) and model separately on different parts of data. Similar to that method, we also do one-hot encoding on other features such as borough, building class category, tax class at present etc.

```
 #   Column                                                                  Non-Null Count   Dtype
---  ------                                                                  --------------   -----
 0   TAX CLASS AT TIME OF SALE                                               569270 non-null  int32
 1   BOROUGH_1                                                               569270 non-null  uint8
 2   BOROUGH_2                                                               569270 non-null  uint8
 3   BOROUGH_3                                                               569270 non-null  uint8
 4   BOROUGH_4                                                               569270 non-null  uint8
 5   BOROUGH_5                                                               569270 non-null  uint8
 6   ZIP CODE_10001                                                          569270 non-null  uint8
 7   ZIP CODE_10002                                                          569270 non-null  uint8
 8   ZIP CODE_10003                                                          569270 non-null  uint8
 9   ZIP CODE_10004                                                          569270 non-null  uint8
 10  ZIP CODE_10005                                                          569270 non-null  uint8
 11  ZIP CODE_10006                                                          569270 non-null  uint8
 12  ZIP CODE_10007                                                          569270 non-null  uint8
 13  ZIP CODE_10009                                                          569270 non-null  uint8
 14  ZIP CODE_10010                                                          569270 non-null  uint8

217 ZIP CODE_11443                                                           569270 non-null  uint8
218 ZIP CODE_11691                                                           569270 non-null  uint8
219 ZIP CODE_11692                                                           569270 non-null  uint8
220 ZIP CODE_11693                                                           569270 non-null  uint8
221 ZIP CODE_11694                                                           569270 non-null  uint8
222 ZIP CODE_11696                                                           569270 non-null  uint8
223 ZIP CODE_11697                                                           569270 non-null  uint8
224 BUILDING CLASS CATEGORY_01  ONE FAMILY DWELLINGS                         569270 non-null  uint8
225 BUILDING CLASS CATEGORY_01  ONE FAMILY HOMES                             569270 non-null  uint8
226 BUILDING CLASS CATEGORY_02  TWO FAMILY DWELLINGS                         569270 non-null  uint8
227 BUILDING CLASS CATEGORY_02  TWO FAMILY HOMES                             569270 non-null  uint8
228 BUILDING CLASS CATEGORY_03  THREE FAMILY DWELLINGS                       569270 non-null  uint8
229 BUILDING CLASS CATEGORY_03  THREE FAMILY HOMES                           569270 non-null  uint8
230 BUILDING CLASS CATEGORY_04  TAX CLASS 1 CONDOS                           569270 non-null  uint8
231 BUILDING CLASS CATEGORY_05  TAX CLASS 1 VACANT LAND                      569270 non-null  uint8
```

```
271 BUILDING CLASS CATEGORY_44  CONDO PARKING                          569270 non-null  uint8
272 BUILDING CLASS CATEGORY_45  CONDO HOTELS                           569270 non-null  uint8
273 BUILDING CLASS CATEGORY_46  CONDO STORE BUILDINGS                  569270 non-null  uint8
274 BUILDING CLASS CATEGORY_47  CONDO NON-BUSINESS STORAGE             569270 non-null  uint8
275 BUILDING CLASS CATEGORY_48  CONDO TERRACES/GARDENS/CABANAS         569270 non-null  uint8
276 BUILDING CLASS CATEGORY_49  CONDO WAREHOUSES/FACTORY/INDUS         569270 non-null  uint8
277 TAX CLASS AT PRESENT_1                                             569270 non-null  uint8
278 TAX CLASS AT PRESENT_1A                                            569270 non-null  uint8
279 TAX CLASS AT PRESENT_1B                                            569270 non-null  uint8
280 TAX CLASS AT PRESENT_1C                                            569270 non-null  uint8
281 TAX CLASS AT PRESENT_1D                                            569270 non-null  uint8
282 TAX CLASS AT PRESENT_2                                             569270 non-null  uint8
283 TAX CLASS AT PRESENT_2A                                            569270 non-null  uint8
284 TAX CLASS AT PRESENT_2B                                            569270 non-null  uint8
285 TAX CLASS AT PRESENT_2C                                            569270 non-null  uint8
286 TAX CLASS AT PRESENT_3                                             569270 non-null  uint8
287 TAX CLASS AT PRESENT_4                                             569270 non-null  uint8
```

Modeling

We successfully tested our data on several models regarding the sales price prediction. We have tried four models as our baseline models which are: Random Forest Regressor, AdaBoost, LightGBM, and boost. Based on the result, we take several measures so as to improve the prediction performance.

First, We change the evaluation method. The optimized model is measured by the mean absolute percentage error(mean relative error). The reasons why we choose this criterion are that firstly, MAE, although is the most common choice, is not a good indicator in our case. For example, an apartment in Manhattan costs 3 million dollars. And a house in the Bronx costs 200k. The prediction MAE of 90k makes a different impact on those two real estates. Instead of MAE, we chose MAPE as our indicator, which is more persuasive and explainable. Unlike MAE, there are limited open sources functions that we can use. So using MAPE is more challenging since we need to define the MAPE function in different models.

Second, we also apply logarithm to the sales price and retest our models. Specifically, for the XGBoost model, we build two models. With the first model, we train the dataset with initial sales price and compute MAE while testing. With the second one, we compute log function on sales price and pass the log of price into our model. And similar to other models, we chose MAE as an indicator. However, the MAE on log price does not explain too much. In order to make it more straightforward, we convert the MAE of log price to the true MAE of sales price.

Third, we adjust parameters contained in our models to gain better prediction results. For example, we tried our model on different learning rate, n estimators, cv, minimum number of leaves etc.

Pros and Cons:

1. Random Forest:
    a. Advantage:
        i. The prediction performance is much better than other simple data mining algorithms. The MAPE result using Random Forest is around 0.27, which can compete with the best supervised learning algorithms.
        ii. Random forest model we build can provide a reliable feature importance estimate.
        iii. it offers efficient estimates of the test error without incurring the cost of repeated model training associated with cross-validation.
    b. Disadvantage:
        i. It is less interpretable than an indiviral tree or other simpler models.
        ii. It requires lots of memory and time when it is used to train a large number of trees. In other words to say, the cost of using random forest to do data mining work is relatively high.
        iii. Predictions are slower than other models, which brings challenges to applications
2. AdaBoost:
    a. Advantages:
        i. AdaBoost is fast and easy to compute
        ii. It has the complexity to be combined with any machine learning algorithms and we don't need to tune the parameters except for T.
    b. Disadvantages:
        i. It may face problems of overfitting, especially for weak classifiers. Besides, weak classifiers can lead to low margins while predicting.
        ii. It is from empirical evidence and particularly vulnerable to uniform noise.
3. Comparisons of XGBoost and Light GBM:
    a. Overfitting controlling capability:
        i. Learning rate of XGBoost lies between 0.01 to 0.2. And minimum child weight is similar to minimum child leaf. Those two parameters are used to control overfitting.

ii. Maxdepth of Light GBM is set default as 20. And it is important to tune. Besides, the number of leaves in a tree, which should be smaller than 2^max depth, is also an important parameter

b. Speed controlling capability:
    i. Subsample ratio of columns, subsample ratio of training instances, and maximum number of decision trees (n estimators) are all useful parameters in XGBoost that can control speed.
    ii. As for Light GBM, fraction of features to be taken for each criterion, data to be used for each criterion and num iterations are used to control speed
    iii. Pros and Cons:
        1. Advantages of Light GBM:
            a. it is fast in training speed and has higher efficiency. It also buckets continuous feature values into discrete bins which fasten the training procedure.
            b. It requires Lower memory usage because it will replace continuous values to discrete bins
            c. It has better accuracy than any other boosting algorithm since it produces much more complex trees
        2. Disadvantages of Light GBM: it may lead to overfitting
        3. Advantages of XGBoost:
            a. It requires less feature engineering
            b. It is fast
            c. It can handle large data sets well
            d. It is less prone to overfitting
            e. It has better performance
        4. Disadvantages of XGBoost:
            a. XGBoost is much more difficult to interpret
            b. If parameters are not tuned properly, it may lead to overfitting
            c. It is harder to tune since there are many hyperparameters

Based on the analysis of pros and cons we did, XGBoost is the best choice for several reasons. First, we have large datasets, for nearly 300k pieces of data. Speed and overfitting controlling capability are extremely important. Also, we require better performance on it. Taking all these factors into consideration, XGBoost is our best choice.

Evaluation

After testing the four models, we noticed that the random forest model and XGBoost model achieves the best performance with the mean absolute percentage error of 0.25. The rest of the models achieve above 0.27 although we have tried different parameters. As for MAE, we convert log back to dollars and get the mean absolute error of 90k. We believe MAE of 90k is a good result since prices of real estate are always much higher. In our baseline model, the best result of MAPE is around 0.34. That is to say, through a series of taken measures, our optimized model shows better performance than the baseline ones.

From a business perspective, we can predict reasonable price intervals based on the prediction result. To be more specific, if we use the developed XGBoost model to predict the sales price of a given real estate, and the predicted price is 1million dollars. Then for MAE of 90k, we can assume the reasonable price interval to be from 910k to 1090k dollars. Beside, for MAPE of 0.25, the price interval will be from 0.75 million to 1.25 million dollars. As a real estate sales company, we can develop a sales strategy on the basis of that price interval. For example, the sales company can send ads to those targeted customers. Those targeted customers can be people who want to buy a house worth around 0.75 million dollars or 1.25 million dollars. Apart from that, the sales company can develop further promotional strategies according to the price interval.

Deployment

Apart from our endeavor to estimate the prices of real estate in NYC by factoring in past sales data and leveraging data mining techniques to extract patterns and trends to support the data-driven decision making about buying or selling a property, we believe data science will have an important role to play, by itself and in collaboration with many of the upcoming emerging technologies and it will be able to not only improve a business strategy but also improve the way and quality of our lives. Data Science techniques implemented here can be further extended to include additional data points such as consumer behavior, interests, and preferences in order to propose the ideal apartment for each client along with the estimate for the right price.

The IoT is becoming more and more necessary and being integrated with the real estate sector. Sensors that record temperature, air, equipment condition allow for responsive environments that adapt to user's habits and behavior. Such data can have a direct impact on the chosen place of residence and investment criteria and will increasingly influence buying and selling habits in the near future and would thus need to be included in the feature set in order to further perfect the estimates along with personalization with respect to the investor. Such risks are always

relevant in the realm of data science which is highly experimental and organizations need to continuously experiment with varied features and continuously invest in data acquisition that could potentially provide them an edge in the market. For us to implement the same, we could expand our feature set to include these parameters and in a cyclic manner understand their influence on the data mining process and continue to reinvent the solution with new features that prove to be influential with experimentation.

Although with respect to our dataset and means of collection there are no ethical concerns but organizations must know where to draw the line in their data acquisition strategy in order to not invade the privacy of the investor for the sake of providing more and more personalization service as their selling point and differentiator.

Also, how emerging technologies such as blockchain and cryptocurrency could revolutionize the way transactions take place in the industry. One such interesting project that we came across was the New York City Real Estate Coin(NYCREC). It is intriguing to think about the prospects of applying data science for assisting buyers and sellers to probabilistically predict the right price to transact to sell or purchase a property and also in a transparent way through an immutable ledger in an industry where how transactions are made possible only represents the tip of the iceberg.

Appendix

1. A comprehensive beginner's guide to create a Time Series Forecast (with Codes in Python and R): https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/
2. Data Source: NYC Calendar Sales (Archive)

   https://data.cityofnewyork.us/Housing-Development/NYC-Calendar-Sales-Archive-/uzf5-f8n2
3. New York City Real Estate Coin: NYCREC
4. Data Science in Real Estate Implications:

   https://www.realestate.bnpparibas.com/how-data-science-transforming-real-estate
5. WikiPedia:

   https://en.wikipedia.org/wiki/Real_estate

   https://en.wikipedia.org/wiki/Category:Real_estate_industry

Group Effort

1. Baseline model & Optimized model: Xingyu Liu, Xiaoxuan Wang
2. Data Visulization : Shashank rajiv Lochan
3. Data cleaning & Data Preparation: Xiaoxaun Wang, Shashank rajiv Lochan
4. Time Series Analysis: Xingyu Liu
5. Report: : Xingyu Liu, Xiaoxuan Wang, Shashank rajiv Lochan