# NYC House Sale Price Prediction Report

Yuan Ling
Email: yl4452@nyu.edu

Biqi Lin
Email: bl2379@nyu.edu

Runyu Yan
Email: ry787@nyu.edu

Dajiang Liu
Email: dl3502@nyu.edu

***Abstract*—Figuring out how to predict the property sales is always one of the most important economical topics on the table. As we all know, House price trends are not only the concern of buyers and sellers, but it also indicates the current economic situation. However, there are various factors that might influence the property market and the house price. We compare three different regression models, LASSO, K Nearest Neighbors (KNN), and Random Forest (RF), to predict the house prices in New York City, USA. The experimental results show that The Random Forest gives the best performance with the minimal RMSE = 0.7297. Zip code, block and gross square feet were the top 3 important features. The results revealed the fact that the location of the building and the total area of the property are the most significant factors to the sale price.**

## I. Introduction

Prices of real-estate properties is critically linked with economy. Good estimation of property prices is crucial for real-estate agencies, tax regulators, banks, insurance companies, as well as individual home-owners. Several studies have produced positive results in predicting housing prices in different locations around the world (Park & Bae, 2015; Saunders, Gammerman, & Vovk, 1998; Kontrimas & Verikas, 2011; Antipov & Pokryshevskaya, 2012; Fan et al., 2006; McGeal et al., 1998). In these studies, different machine learning algorithms (Support Vector Machine, Multilayer Perceptron, Nave Bayesian, Random Forest (RF), etc.) have been compared. Many of the results showed RF provided more accurate predictions. The current study aimed to offer further exploration into the performance of several mainstream machine learning regression algorithms in predicting housing prices. The algorithms we compared were K Nearest Neighbors (KNN), Lasso Regression (Lasso), and RF. The statistical benchmarks we used to mark the performance were Mean Squared Error (MSE). We further discussed several common practices in dealing with certain real estate sales features. Programming was done in Jupyter Notebook with Python 3 Kernel.

## II. Methodology

### A. Data pre-process

The data we used were online-archived real estate sales data of New York City from 2016 to 2017, retrieved from the website of New York Citys Department of Finance. The data consists of 20 explanatory features and the target variable (Sale Price). There are in total 84,548 rows of data. Firstly, we deal with the missing values problem. All rows with missing target variables or with extreme outliers in the target variable were dropped. Then the features with 60% or higher missing values were dropped. The rest missing values were conducted imputation. Specifically, we imputed the missing values for numerical features with the mean, and for categorical features with the mode. Secondly, we dropped the near-zero variance features. Lastly, the skewness was checked for the target variable. We found that the target variable was significantly skewed to the right (skewness = 114.14). Therefore, we applied log transformation for the target variable, and resulting variable reached good normal distribution (skewness = -1.047). After the data pre-process, the dataset contains p = 17 features, and n = 58852 observations.

### B. Exploration data analysis

All features were separated into numerical group and categorical group to do the Exploration Data Analysis. For the numerical features, the scatter plot matrix, correlation matrix and boxplot were checked. For categorical features, the bar plots between each of them and sale price were checked, respectively. The results shown that firstly, there was no clear linear relationship between the numerical features and the sale price; secondly, the col-linearity problem and outlier problem existed among numerical features; thirdly, all categorical features had different degrees of impact on the sale price. Based on these findings, three models were employed as the candidate models in this study: LASSO, KNN and RF Regression. Since this is a regression problem, RMES was employed as the criterion to compare the models predictive performance.

### C. Feature design and selection

Some methods like LASSO have a hard time dealing with multicollinearity. Therefore, for the pairs with correlation higher than 0.7, we remove one of them so that the correlation between each two remaining features is less than 0.7. We end up with 5 numerical and 11 categorical features after this filtering. Principle component analysis (PCA) has the potential to deal with highly correlated variables and extract most informative features. Therefore, we make the comparison on the LASSO Model with or without the PCA method.

### D. Models

**LASSO** LASSO is a special version of linear regression. Compared with the ordinary least squares (OLS) linear regression, it enhances the prediction accuracy by performing both variable selection and regularization during the fitting procedure. LASSO relies on the linear model but adds some shrinkage penalty to do the fitting procedure for estimating the coefficients. The new procedure is much like a best subset selection. Some useless features will be filter out from the

the final model automatically. The tuning parameter for the LASSO is $\lambda$. Selecting a good value of $\lambda$ for the LASSO is critical.

**KNN** KNN is a completely non-parametric approach where no assumptions are made about the shape of the decision boundary. KNN is an instanced- based learning technique, which means instead of learning a model, it calculates the similarity among instances for prediction. KNN has simple implementation and only uses local information, which can yield highly adaptive behavior. The tuning parameter is $K$ for KNN, whose value influences the models performance significantly.

**Random Forests** RF is a modified version of bagging trees that build a large set of de-correlated trees. It runs efficiently on large dataset, and can handle thousands of input variables without variable deletion. Furthermore, it avoids over-fitting problem. To apply random forest, positive integer $B$ regression trees are built concurrently using $B$ bootstrapped training sets. The final prediction is the average of the $B$ regression trees in the forest. Different from the bagging tree, when building each tree, the split is selected from a random subset of all predictors. That means if there are $p$ predictors in total, only $m$ predictors ($m < p$) will be consider as the candidate splits. One split is selected from those m predictors. A new fresh sample of $m$ predictors is taken for selecting each next split. The tuning parameters for random forest are $B$ and $m$.

## III. Results

The whole data set was randomly divided into the training data and the testing data with the ratio 7:3. The training set was used to apply 10-fold cross-validation with the Mean Standard Error (MSE) for selecting the optimal tuning parameter for each model.

### A. KNN

For KNN, the number $k$ of nearest neighbors was tuned from the odd numbers between 0 and 20. Fig. 1 shows the MSE varying with $k$. The optimal $k$ with minimal MSE was determined to be 11.

### B. LASSO

For LASSO, the tunning parameter $\lambda$ is chosen from $\{0.1, 0.05, 0.025, 0.01, 0.001, 0.0001, 0.00001\}$. Fig. 2 shows the effect of different values of $\lambda$ with MSE. We found the larger $\lambda$, the smaller value of MSE. The optimal $\lambda$ is determined to be 0.1.

Highly correlated variables may decrease the prediction accuray of LASSO method. So consider two means to reduce correlation in the training set of our LASSO method. Firstly, we apply PCA on the training data and pick up top ten principle components. Alternatively,we select the top 5 important features {ZIP CODE, BLOCK, GROSS SQUARE FEET, LOT, BOROUGH} ranked by the random forest regression model to train our model. However, we did not observe performance improvement of either methods. The method comparison is in Fig 3.

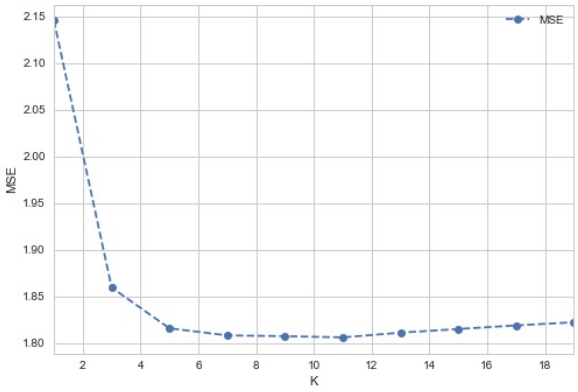


Fig. 1: Mean Standard Error as a function of the number of nearst neighbours (K) for KNN.
Results are from 10-fold cross-validation.

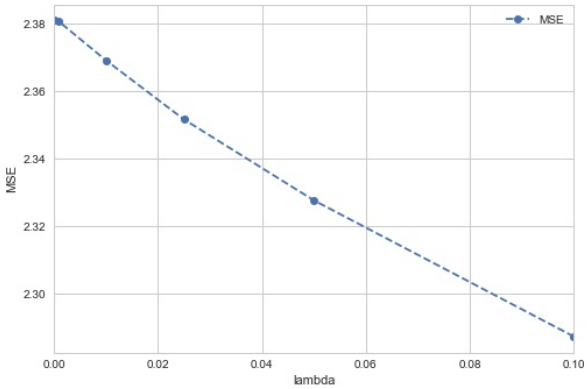


Fig. 2: Mean Standard Error as a function of ($\lambda$) for LASSO
Results are from 10-fold cross-validation.

### C. Random Forest Tree

For Random Forest, considering the time cost, in this project we only tune the number of trees $B$, which is chosen from $\{100, 200, 300, 400, 500\}$. The parameter $m$ (the number of randomly selected predictors which were considered as the split candidates) was set to be $m = \sqrt{p} = \sqrt{18} \approx 4$.. Fig. 4 showed the value of MSE with different $B$ values. The optimal $B$ was determined to be 500.

The feature importance was calculated in the Random Forest model where $B$ is set to be the optimal value 500 (Fig. 5). Zip code, block and gross square feet were the top 3 important features. The results revealed the fact that the location of the building and the total area of the property are the most significant factors to the sale price.

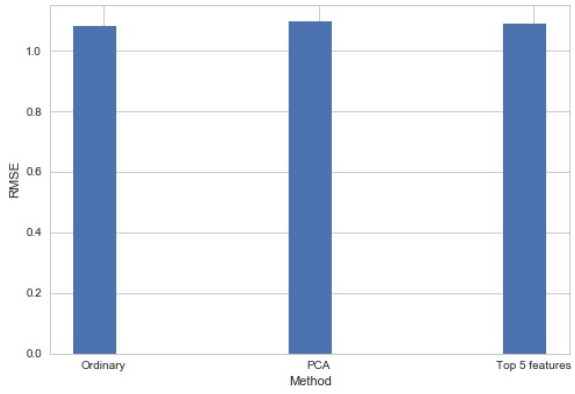


Fig. 3: Root Mean Standard Error with different training set for LASSO.

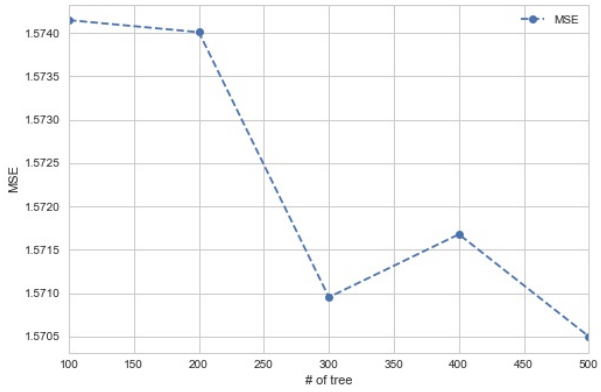


Fig. 4: Mean Standard Error as a function of the number of trees for the Random Forest Regression.
Results are from 10-fold cross-validation

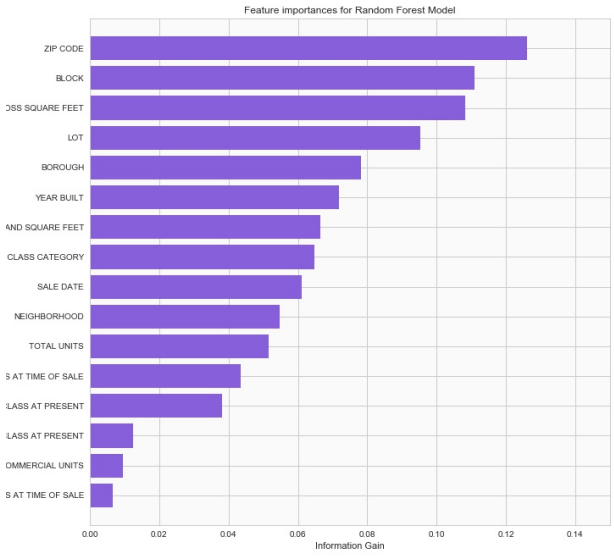


Fig. 5: Feature Importance Ranking by Information Gain for the Random Forest Regression.

### *D. Discussion*

*1) Model Comparision:*
Models with their optimal tuning parameter were used to perform prediction based on the testing data set. The predicted sale prices and the actual sale price were used to compute the RMSE. The model with the minimal RMSE was selected as the best model. We found that KNN and Random Forest Regression perform significantly better than LASSO. And the result in Table I shows that Random Forest is the best model, with the least RMSE 0.7297. We expect that this dramatic different in performance is due to the possible non-linear relation between the target and the features in our model. In particular, the sale price has unlikely to be linearly dependent on the geographical variables such as "ZIP CODE", "BLOCK", which are among the top important features suggested by the Random Forest model. As a linear regression model, LASSO cannot account for these nonlinear interactions. On the other hand, the Random Forest model is able to discover more complex dependencies. This may also explain why some feature engineering operations (see Fig. 3) like PCA and selecting top 5 features don't improve the prediction accuracy in LASSO Model. KNN performs much better than LASSO since it can distinguish linear or non-linear variables as the Random Forest Model, but it is very sensitive to the noise data. And we expect that our date set contains a non-negligible amount of noise. Random Forest model, as an ensemble method, can overcome this deficiency. In total, Random Forest turns out to be the best model to predict sale price.

TABLE I: The performace comparsion with different models

| Model | LASSO | KNN | Random Forest |
|---|---|---|---|
| RMSE | 1.0806 | 0.8695 | 0.7297 |

*2) Processing speed:*
One possible drawback of the current study is the training speed of the models. The current study did not use speed boosting methods such as GPU-processing or parallel processing. Therefore, the processing speed of training certain models such as RF was slightly long for the size of the dataset. There are several programming packages (Theano, Tensorflow, etc.) available in Python environment which support GPU or parallel processing. Future studies could apply speed boosting methods in training more complicated models such as RF, Deep Neural Network, and SVM, or when the size of the dataset is comparatively larger.

*3) Future Work:*

In this project, we used three different models to predict the sale price. It has been reported that some machine learning models such as XGBoost, Neural Network regression perform well in similar tasks. One future direction is to explore these methods in our setting.

Another important future direction is obtaining better data set. One possibility is to get more historical data. (Our current date set is collected from 2016 to 2017). Another possible direction is to find better data cleaning methods.

In this paper we use 16 features were eventually selected to predict the sale price. The asking price is not involved in the analysis. There are two interesting questions one can consider once the asking price is involved. Replace the sale price by the asking price of the house as the target. The practical application of solving such a problem is to determine whether a house is over priced or under priced. Another interesting question is to include asking price as a feature and predict the sales price. We believe that the sale price is strongly correlated with the asking price, much stronger than any other features. Therefore we expect a different method.

## IV. Conclusion

We apply three regression models to predict New York house sale price: LASSO, KNN, Random Forest Regression. The Random Forest Regression gives the best performance, with RMSE$= 0.7297$. We believe it is due to the nonlinearity interaction between the target and certain features, as well as the noise in our data set. Our result can serve as a reference for house sellers.

## V. Appendix

The dataset used for this project and the code used for the analysis can be found at the link below:

https://github.com/nyu-ds-HousePricePred

## References


[1] ntipov, E. A., & Pokryshevskaya, E. B. (2012). Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics. Expert Systems with Applications, 39(2), 1772-1778.

[2] an, G. Z., Ong, S. E., & Koh, H. C. (2006). Determinants of house price: A decision tree approach. Urban Studies, 43(12), 2301-2315.

[3] ontrimas, V., & Verikas, A. (2011). The mass appraisal of the real estate by computational intelligence. Applied Soft Computing, 11(1), 443-448.

[4] cGreal, S., Adair, A., McBurney, D., & Patterson, D. (1998). Neural networks: the prediction of residential values. Journal of Property Valuation and Investment, 16(1), 57-70.

[5] ark, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. Expert Systems with Applications, 42(6), 2928-2934.

[6] aunders, C., Gammerman, A., & Vovk, V. (1998). Ridge regression learning algorithm in dual variables. Proceedings of the 15th International Conference on Machine Learning, ICML 98.