Methodology

Our goal is to compare the use of spatial lags as features in a machine learning predictive model against traditional feature engineering techniques. We will create three modeling data sets:

- Base
- Zip Code
- Spatial Lag

We will then create 2 predictive models for each modeling data set, using a different outcome variable for each:

1) Probability of Sale. The probability that a given property in New York City will sell in a given year
2) Amount of Sale. Given that a property sells, how much is the sale value?

There will be six predictive models built in total, as follows:

#	Model	Model Type	Data	Outcome Variable	Outcome Var Type	Evaluation Metric
1	Probability of Sale	Classification	Base	Building Sold	Binary	AUC
2	Probability of Sale	Classification	Zip Code	Building Sold	Binary	AUC
3	Probability of Sale	Classification	Spatial Lags	Building Sold	Binary	AUC
4	Sale Price	Regression	Base	Sale Price per SF	Continuous	RMSE
5	Sale Price	Regression	Zip Code	Sale Price per SF	Continuous	RMSE
6	Sale Price	Regression	Spatial Lag	Sale Price per SF	Continuous	RMSE

To accomplish this, we combine three open-source data repositories provided by New York City via nyc.gov and data.cityofnewyork.us. Our base modeling data set includes all building records and associated sales information from 2003-2017.

Following the creation of the base modeling data, we create two additional data sets through feature engineering: a "Zip Code features" data set and a "Spatial Lag features" data set. The primary goal of this study is to compare the predictive power of the spatial lags vs. the base and zip code features.

Data

The New York City government makes available an annual data set which describes all tax lots in the five boroughs. The Primary Land Use and Tax Lot Output data set, known as PLUTO, contains a single record for every tax lot in the city along with a number of building and tax-related attributes such as Year Built, Assessed Value, Square Footage, number of stories, and many more. At the time of this writing, NYC has made this data set available for all years between 2002-2017, excluding 2008. For convenience, we also exclude the 2002 data set from our analysis because sales information is not available for that year. Importantly for our analysis, the latitude and longitude of the tax lots are also made available, allowing us to locate in space each building and to build geospatial features from the data.

Ultimately, we are interested in sales transactions–both frequency, and amount. Sales transactions are also made available by the New York City government, known as NYC Rolling Sales Data. At the time of this writing, sales transactions are available for the years 2003-2017. The sales transactions data contains additional data fields describing time, place, and amount of sale as well as additional building characteristics. Crucially, the sales transaction data does not include geographical coordinates, making it impossible to perform geospatial analysis without first mapping the sales data to PLUTO.

Prior to mapping to PLUTO, the sales data must first be transformed to include the proper mapping key. New York City uses a standard key of Borough-Block-Lot to identify tax lots in the data. For example, 31

West 27th Street is located in Manhattan, on block 829 and lot 16, therefore, its Borough-Block-Lot (BBL) is 1_829_16 (the 1 represents Manhattan). The sales data contains BBL's at the building level, however, the sales transactions data does not appropriately designate condos as their own BBLs. Mapping the sales data directly to the PLUTO data results in a mapping error rate of 23.1%. Therefore, the sales transactions data must first be mapped to another data source, the NYC Property Address Directory, or PAD), which contains an exhaustive list of all BBLs in NYC. Once the sales data is combined with PAD, the data can be mapped to PLUTO with an error rate of 0.291%.

After the Sales Transactions data has been mapped to PAD, it can then be mapped to PLUTO. The sales data is normalized and filtered so that only BBLs with less than or equal to 1 transactions in a year occur. The final data set is an exhaustive list of all tax lots in NYC for every year between 2003-2017, whether that building was sold, for what amount, and several other additional variables.

Only building categories of significant interest are included in the data. The following building types are included:

Category	Description
A	ONE FAMILY DWELLINGS
B	TWO FAMILY DWELLINGS
C	WALK UP APARTMENTS
D	ELEVATOR APARTMENTS
F	FACTORY AND INDUSTRIAL BUILDINGS
G	GARAGES AND GASOLINE STATIONS
L	LOFT BUILDINGS
O	OFFICES

The data is further filtered to include only records with equal to or less than 2 buildings per tax lot. The global filtering of the data set reduces the base modeling data from 12,012,780 records down to 8,247,499.

Feature Engineering

Base modeling data

The base modeling data set is enhanced to include additional features. A summary table of the additional features are presented below:

Feature	Min	Median	Mean	Max
has_building_area	0	1.00	1.00	1.00
Percent_Com	0	0.00	0.16	1.00
Percent_Res	0	1.00	0.82	1.00
Percent_Office	0	0.00	0.07	1.00
Percent_Retail	0	0.00	0.04	1.00
Percent_Garage	0	0.00	0.01	1.00
Percent_Storage	0	0.00	0.02	1.00
Percent_Factory	0	0.00	0.00	1.00
Percent_Other	0	0.00	0.00	1.00
Last_Sale_Price	0	312.68	531.02	62,055.59
Last_Sale_Price_Total	2	2,966,835.00	12,844,252.00	1,932,900,000.00
Years_Since_Last_Sale	1	4.00	5.05	14.00
SMA_Price_2_year	0	296.92	500.89	62,055.59
SMA_Price_3_year	0	294.94	495.29	62,055.59
SMA_Price_5_year	0	300.12	498.82	62,055.59

Feature	Min	Median	Mean	Max
Percent_Change_SMA_2	-1	0.00	685.69	15,749,999.50
Percent_Change_SMA_5	-1	0.00	337.77	6,299,999.80
EMA_Price_2_year	0	288.01	482.69	62,055.59
EMA_Price_3_year	0	283.23	471.98	62,055.59
EMA_Price_5_year	0	278.67	454.15	62,055.59
Percent_Change_EMA_2	-1	0.00	422.50	9,415,128.85
Percent_Change_EMA_5	-1	0.06	308.05	5,341,901.60

A binary variable is created to indicate whether a tax lot has a building on it (i.e., whether it is an empty plot of land or not). In addition, building types are quantified by what percent of the square footage belongs to the major property types: Commercial, Residential, Office, Retail, Garage, Storage, Factory and Other.

Importantly, two variables are created from the Sales Prices: A price-per-square-foot figure ("Last_Sale_Price") and a total Sale Price ("Last_Sale_Price_Total"). Sale Price per Square foot eventually becomes the outcome variable in one of the predictive models, even though it is referred to as Sale Price. Further features are derived which carry forward the previous sale price of a tax lot, if there is one, through successive years. Previous Sale Price is then used to create Simple Moving Averages (SMA), Exponential Moving Averages (SMA), and percent change measurements between the moving averages. In total, 69 variables are input to the feature engineering process and 92 variables are output. The final base modeling data set is 92 variables by 8,247,499 rows.

Zip code modeling data

The first of the comparative modeling data sets is the Zip code modeling data. Using the base data as a starting point, several features are generated to describe characteristics of the zip code where the tax lot resides. A summary table of the Zip code level features is presented below.

Feature	Min	Median	Mean	Max
Last_Year_Zip_Sold	0.00	27.00	31.14	112.00
Last_Year_Zip_Sold_Percent_Ch	-1.00	0.00	Inf	Inf
Last_Sale_Price_zip_code_average	0.00	440.95	522.87	1,961.21
Last_Sale_Price_Total_zip_code_average	10.00	5,312,874.67	11,877,688.55	1,246,450,000.00
Last_Sale_Date_zip_code_average	12,066.00	13,338.21	13,484.39	17,149.00
Years_Since_Last_Sale_zip_code_average	1.00	4.84	4.26	11.00
SMA_Price_2_year_zip_code_average	34.31	429.26	501.15	2,092.41
SMA_Price_3_year_zip_code_average	34.31	422.04	496.47	2,090.36
SMA_Price_5_year_zip_code_average	39.48	467.04	520.86	2,090.36
Percent_Change_SMA_2_zip_code_average	-0.20	0.04	616.47	169,999.90
Percent_Change_SMA_5_zip_code_average	-0.09	0.03	341.68	113,333.27
EMA_Price_2_year_zip_code_average	30.77	401.43	479.38	1,883.81
EMA_Price_3_year_zip_code_average	33.48	419.11	479.95	1,781.38
EMA_Price_5_year_zip_code_average	29.85	431.89	472.80	1,506.46
Percent_Change_EMA_2_zip_code_average	-0.16	0.06	388.90	107,368.37
Percent_Change_EMA_5_zip_code_average	-0.08	0.07	326.17	107,368.38
Last_Sale_Price_bt_only	0.00	357.71	485.97	6,401.01
Last_Sale_Price_Total_bt_only	10.00	3,797,461.46	11,745,130.56	1,246,450,000.00
Last_Sale_Date_bt_only	12,055.00	13,331.92	13,497.75	17,149.00
Years_Since_Last_Sale_bt_only	1.00	4.78	4.30	14.00
SMA_Price_2_year_bt_only	0.00	347.59	462.67	5,519.39
SMA_Price_3_year_bt_only	0.00	345.40	458.50	5,104.51
SMA_Price_5_year_bt_only	0.00	372.30	481.09	4,933.05

Feature	Min	Median	Mean	Max
Percent_Change_SMA_2_bt_only	-0.55	0.03	600.10	425,675.69
Percent_Change_SMA_5_bt_only	-0.33	0.02	338.15	188,888.78
EMA_Price_2_year_bt_only	0.00	332.98	442.79	5,103.51
EMA_Price_3_year_bt_only	0.00	332.79	443.02	4,754.95
EMA_Price_5_year_bt_only	0.00	340.57	436.70	4,270.37
Percent_Change_EMA_2_bt_only	-0.47	0.06	377.17	254,462.97
Percent_Change_EMA_5_bt_only	-0.34	0.06	335.17	178,947.30

In general, the base model data features are aggregated to a zip code level and attached to the individual observations, including SMA and EMA calculations. Additionally, a second set of features are added, denoted as "bt_only", which filter only for tax lots of the same building type. In total, the Zip code feature engineering process inputs 92 variables and outputs 122 variables.

Spatial Lag modeling data

Spatial lags are variables created from physically proximate observations. For example, taking the average building age from all buildings within 100 meters of the tax lot in question would be a spatial lag. Creating spatial lags presents both advantages and disadvantages in the modeling process. Spatial lags allow for much more fine-tuned measurements of a building's surrounding area. Knowing the average sale price of all buildings within 500 meters of a building can be much informative than knowing the sale prices of all buildings in the same zip code. However, building spatial lags is computationally expensive.

To build spatial lags for all 8,247,499 observations in our modeling data, we created a spatial indexing technique that sped up the process by allowing for parallelization of the operation. Since tax lots rarely if ever move, we reduced the indexing task to 514,124 points (the number of unique tax lots in New York City). Then, for each point, we calculated and cached every other tax lot within 500 meters of that building. The result was an origin-destination relationship graph that connected each tax lot to its surrounding tax lots.

Next, we used the spatial index to create spatial lag features. One advantage to using spatial lags is the rich number of potential features which can be created. Spatial lags can be weighted based on a distance function, i.e., physically closer observations can be given more weight. For our modeling purposes, we created two sets of features: distance weighted features (denoted with a "_dist") and simple average features (denoted with"_basic"). SMA and EMA as well as percent changes were also calculated. In total, the spatial lag feature engineering process input 92 variables and output 194 variables. A summary of the Spatial Lag features are presented below:

Feature	Min	Median	Mean	Ma
Radius_Total_Sold_In_Year	1.00	20.00	24.00	20
Radius_Average_Years_Since_Last_Sale	1.00	4.43	4.27	14.
Radius_Res_Units_Sold_In_Year	0.00	226.00	289.10	2,9
Radius_All_Units_Sold_In_Year	0.00	255.00	325.94	2,9
Radius_SF_Sold_In_Year	0.00	259,403.00	430,891.57	8,6
Radius_Total_Sold_In_Year_sum_over_2_years	2.00	41.00	48.15	25
Radius_Average_Years_Since_Last_Sale_sum_over_2_years	2.00	9.25	8.70	26.
Radius_Res_Units_Sold_In_Year_sum_over_2_years	0.00	493.00	584.67	3,3
Radius_All_Units_Sold_In_Year_sum_over_2_years	1.00	555.00	660.67	4,2
Radius_SF_Sold_In_Year_sum_over_2_years	2,917.00	580,947.00	872,816.44	14,
Radius_Total_Sold_In_Year_percent_change	-0.99	0.00	0.27	77.
Radius_Average_Years_Since_Last_Sale_percent_change	-0.91	0.13	0.26	8.0
Radius_Res_Units_Sold_In_Year_percent_change	-1.00	-0.04	Inf	Inf
Radius_All_Units_Sold_In_Year_percent_change	-1.00	-0.04	Inf	Inf

Feature	Min	Median	Mean	Ma
Radius_SF_Sold_In_Year_percent_change	-1.00	-0.02	Inf	Inf
Radius_Total_Sold_In_Year_sum_over_2_years_percent_change	-0.96	-0.03	0.03	15.
Radius_Average_Years_Since_Last_Sale_sum_over_2_years_percent_change	-0.72	0.12	0.17	2.5
Radius_Res_Units_Sold_In_Year_sum_over_2_years_percent_change	-1.00	-0.04	Inf	Inf
Radius_All_Units_Sold_In_Year_sum_over_2_years_percent_change	-0.99	-0.04	0.12	84.
Radius_SF_Sold_In_Year_sum_over_2_years_percent_change	-0.98	-0.04	0.18	361
Percent_Com_dist	0.00	0.04	0.07	0.5
Percent_Res_dist	0.00	0.46	0.43	0.6
Percent_Office_dist	0.00	0.01	0.03	0.4
Percent_Retail_dist	0.00	0.02	0.02	0.0
Percent_Garage_dist	0.00	0.00	0.00	0.2
Percent_Storage_dist	0.00	0.00	0.01	0.2
Percent_Factory_dist	0.00	0.00	0.00	0.0
Percent_Other_dist	0.00	0.00	0.00	0.0
Percent_Com_basic_mean	0.00	0.04	0.07	0.5
Percent_Res_basic_mean	0.00	0.46	0.43	0.6
Percent_Office_basic_mean	0.00	0.01	0.03	0.4
Percent_Retail_basic_mean	0.00	0.02	0.02	0.0
Percent_Garage_basic_mean	0.00	0.00	0.00	0.2
Percent_Storage_basic_mean	0.00	0.00	0.01	0.2
Percent_Factory_basic_mean	0.00	0.00	0.00	0.0
Percent_Other_basic_mean	0.00	0.00	0.00	0.0
Percent_Com_dist_perc_change	-0.90	0.00	0.00	6.1
Percent_Res_dist_perc_change	-0.50	0.00	0.03	36.
Percent_Office_dist_perc_change	-1.00	0.00	Inf	Inf
Percent_Retail_dist_perc_change	-0.82	0.00	Inf	Inf
Percent_Garage_dist_perc_change	-1.00	0.00	Inf	Inf
Percent_Storage_dist_perc_change	-1.00	-0.01	Inf	Inf
Percent_Factory_dist_perc_change	-1.00	0.00	Inf	Inf
Percent_Other_dist_perc_change	-1.00	0.00	Inf	Inf
SMA_Price_2_year_dist	0.00	400.01	496.30	3,8
SMA_Price_3_year_dist	0.00	396.94	492.00	3,8
SMA_Price_5_year_dist	8.83	425.55	515.29	3,8
Percent_Change_SMA_2_dist	-0.13	0.03	552.33	804
Percent_Change_SMA_5_dist	-0.09	0.02	317.46	322
EMA_Price_2_year_dist	0.00	378.63	475.54	3,4
EMA_Price_3_year_dist	8.83	382.25	476.05	3,2
EMA_Price_5_year_dist	7.88	386.34	468.91	2,8
Percent_Change_EMA_2_dist	-0.09	0.06	346.51	480
Percent_Change_EMA_5_dist	-0.02	0.06	303.55	273
SMA_Price_2_year_basic_mean	0.02	412.46	496.75	2,5
SMA_Price_3_year_basic_mean	0.02	409.00	492.43	2,5
SMA_Price_5_year_basic_mean	17.16	443.34	515.67	2,6
Percent_Change_SMA_2_basic_mean	-0.13	0.04	543.51	393
Percent_Change_SMA_5_basic_mean	-0.09	0.03	312.46	157
EMA_Price_2_year_basic_mean	0.02	390.30	475.96	2,2
EMA_Price_3_year_basic_mean	11.39	393.25	476.45	2,1
EMA_Price_5_year_basic_mean	15.30	402.06	469.09	1,8
Percent_Change_EMA_2_basic_mean	-0.09	0.06	340.89	235
Percent_Change_EMA_5_basic_mean	-0.02	0.06	296.78	133
SMA_Price_2_year_dist_perc_change	-0.74	0.05	0.17	10,
SMA_Price_3_year_dist_perc_change	-0.74	0.05	0.17	10,

Feature	Min	Median	Mean	Ma
SMA_Price_5_year_dist_perc_change	-0.74	0.04	0.06	15.
Percent_Change_SMA_2_dist_perc_change	-Inf	-0.24	NaN	Inf
Percent_Change_SMA_5_dist_perc_change	-Inf	-0.14	NaN	Inf
EMA_Price_2_year_dist_perc_change	-0.74	0.06	0.18	10,
EMA_Price_3_year_dist_perc_change	-0.73	0.06	0.08	15.
EMA_Price_5_year_dist_perc_change	-0.63	0.06	0.07	12.
Percent_Change_EMA_2_dist_perc_change	-Inf	-0.13	NaN	Inf
Percent_Change_EMA_5_dist_perc_change	-556.60	-0.10	Inf	Inf
SMA_Price_2_year_basic_mean_perc_change	-0.55	0.05	0.12	9,3
SMA_Price_3_year_basic_mean_perc_change	-0.55	0.05	0.11	9,3
SMA_Price_5_year_basic_mean_perc_change	-0.50	0.04	0.06	5.9
Percent_Change_SMA_2_basic_mean_perc_change	-Inf	-0.19	NaN	Inf
Percent_Change_SMA_5_basic_mean_perc_change	-Inf	-0.12	NaN	Inf
EMA_Price_2_year_basic_mean_perc_change	-0.53	0.06	0.12	9,3
EMA_Price_3_year_basic_mean_perc_change	-0.47	0.06	0.08	23.
EMA_Price_5_year_basic_mean_perc_change	-0.37	0.06	0.07	4.8
Percent_Change_EMA_2_basic_mean_perc_change	-Inf	-0.13	NaN	Inf
Percent_Change_EMA_5_basic_mean_perc_change	-136.59	-0.11	Inf	Inf

Outcome Variables

The final step in creating the modeling data is to define the outcome variables. For our purposes, we create two dependent variables: - Sold. Whether a tax lot sold in a given year. Used in the Probability of Sale classification model. - Sale Price. The price-per-square foot assocaited with a transaction, if a sale took place. Used in the Sale Price Regression model.

The following table describes the distributions of both outcome variables:

	Sold	Sale Price per SF
Min.	0.00	0.0
1st Qu.	0.00	163.5
Median	0.00	375.2
Mean	0.04	644.8
3rd Qu.	0.00	783.3
Max.	1.00	83,598.7

Algorithm

Previous works (see: Antipov and Pokryshevskaya (2012); also Schernthanner H. (2016)) have found the Random Forest algorithm (Breiman 2001) suitable to prediction tasks invlovling real estate. While algorithms exist that may marginally outperform Random Forest in terms of predictive accuracy (such as nueral networks and functional gradient descent algorithms), Random Forest is highly scalable and paralellizable, and therefore a natural choice for comparing different feature engineering strategies (such as in this paper).

Random Forest can be used for both classification and regression tasks. The Random Forest algorithm works by generating a large number of independent classification or regression decision trees and then employing majority voting (for classification) or averaging (for regression) to generate predictions. Over a dataset of N rows by M predictors, a bootstrap sample of the data is chosen ($n < N$) as well as a subset of the predictors ($m < M$). Individual decision/regression trees are built on the n by m sample. Because the trees can be built independently (and not sequentially, as is the case with most functional gradient descent algorithms), the tree

building process can be executed in parallel across an arbitrary number of computer cores. With a sufficiently large number of cores, the model training time can be significantly reduced. This provides a highly accurate, robust prediction model that avoids many of the drawbacks of traditional parametric techniques, such as OLS.

The primary advantages to using Random Forest with real estate data are:

1. Can handle an arbitrarily large number of variables while avoiding the curse of dimensionality associated with regression techniques. Increasing the number of predictors in a multiple regression can quickly lead to overfitting.
2. Can accomodate categorical variables with many levels. Real estate data often contains information describing the location of the property, or the property itself, as one of a large set of possible choices, such as neighborhood, county, census tract, district, property type, and zoning information. Because factors need to be recoded as individual dummy variables in the model building process, factors with many levels will quickly encounter the curse of dimensionality in multiple regression techniques.
3. Appropriately handles missing data. Predictions can be made with the parts of the tree which are succesfully built, and therefore, there is no need to filter out incomplete observations or impute missing values. Since much real destate data is self reported, incomplete fields are common in the data.
4. Robust against outliers. Because of bootstrap sampling, outliers appear in individual trees less often, and therefore, are reduced in terms of importance. Real estate data, especially with regards to pricing, tends to contain outliers. For example, the dependent variable in one of our models, Sale Price (see: Table 5), shows a clear divergence in median and mean, as well as a maximum significantly higher than the third quartile.
5. Can recognize non-linear relationships in data, which is useful when modeling spatial relationships.
6. Is not affected by colinearity in the data. This is highly valuable as real estate data can be highly correlated.
7. The algorithm can be parallelized and is relatively fast compared to nueral networks and functional gradient descent algorithms.

To run the model, we have chosen the h2o.randomForest function from the h2o R open source library. The h2o implementation of the Random Forest algorithm is particularly well-suited for high parallelization. For more information, see: https://www.h2o.ai/.

Data Validation

The goal of the predictive models are to be able to successfully predict both the probability and amount of real estate sales into the near future. As such, our models will use out-of-time validation to assess performance. As shown in Figure **??** The models will be trained using data from 2003-2015. 2016 modeling data will be used during the model training process as cross-validation data. Finally, we will score our model using 2017 as a held-out sample. Using out-of-time validation should ensure that the models generalize well into the immediate future.

Variable Selection

For ease of processing and to improve the ability of the model to generalize into the future, a variable selection step is added to the modeling process. A Random Forest model is first trained on a 1% subsample of the modeling data. Variable importance of the resulting model is calculated using the technique proposed by Friedman (2001), i.e., for a collection of decision trees $[T_m]_1^m$:

$$\hat{I}_j^2 = \frac{1}{M} \sum_{m=1}^{M} \hat{I}_j^2(T_m)$$

Where influence I for variable j is calculatd as the sum of corresponding improvements in squared-error for node tree T. After calculating variable importance for the model data subset, the variables are rank-ordered by descending importance. Variables which account for 80% of the total variable importance are chosen to advance to the model training round on the full modeling data sets.

Evaluation Metrics

We have chosen evaluation metrics that will allow us to easily compare the performance of the models against other models with the same outcome variable. The classification models (Probability of Sale) will be compared using Area Under the ROC Curve (AUC). The regression models (Sale Price) will be compared using Root Mean Squared Error (RMSE). Both evaluation metrics are common for their respective outcome variable types, and as such will be useful for comparing within model-groups.

Area Under ROC Curve (AUC)

A classification model typically outputs a probability that a given row in the data belongs to a group. In the case of binary classification, the value falls between 0 and 1. There are many techniques for determining the cut off threshold for classification; a typical method is to assign anything above a 0.5 into the "1" or posistive class. An ROC curve (receiver operating characteristic curve) plots the True Positive Rate vs. the False Positive rate at different classification thresholds; it is a meaurement of the performance of a classification model across all possible thresholds, and therefore sidesteps the need to arbitrarily assign a cutoff.

Area Under the ROC Curve, or AUC measures the entire two-dimensional area underneath the ROC curve. It is the integration of the curve from (0,0) to (1,1), defined as $AUC = \int_{(0,0)}^{(1,1)} f(x)dx$.

AUC provides a relatively standard measure of performance across all possible classification thresholds, and can be interpreted as the probability that the model ranks a random positive example more highly than a random negative example. A value of 0.5 represents a perfectly random model, while a value of 1.0 represents a model that can perfectly discriminate between the two classes. AUC is useful for comparing classification models against one another because they are both scale and threshold-invariant.

One of the drawbacks to AUC is that is does not describe the tradeoffs between false positives and false negatives. In certain cirmustances, a false positive might be considerably less desirable than a fasle negative, oe vice-versa. For our purposes, we rank false positives and false negatives as equally undesirable outcomes.

Root Mean Squared Error

The Root Mean Squared Error (RMSE) is a common measurement of the differences between values predicted by a regression model and the observed values. It is formally defined as $RMSE = \sqrt{\frac{\sum_1^T (\hat{y}_t - y_t)^2}{T}}$, where $\hat{y}$ represents the prediction and y represents the observed value at observation t.

Lower RMSE scores are typically more desireable. An RMSE value of 0 would indicate a perfect fit to the data. RMSE can be difficult to interpret on its own, however, it is useful for comparing models with similar outcome variables. In our case, the outcome variables (Sales Price per Square Foot) are consistent across modeling data sets, and therefore can be reasonably compared using RMSE.

Antipov, Evgeny A., and Elena B. Pokryshevskaya. 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.*

Breiman, L. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

Friedman, Jerome H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *The Annals*

of Statistics 29 (5): 1189–1232.

Schernthanner H., Gonschorek J., Asche H. 2016. "Spatial Modeling and Geovisualization of Rental Prices for Real Estate Portals." *Computational Science and Its Applications* 9788.