

**The Spatially-Conscious Machine Learning Model:
Leveraging Spatial Analysis to Boost the Accuracy of Real Estate Sales
Predictions**

By

Tim Kiely

Thesis Project

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN DATA SCIENCE

Northwestern University

Nathaniel D. Bastian, Ph.D., First Reader

Candice Bradley, Second Reader

Abstract

Successfully predicting gentrification could have many social and commercial applications, however, real estate sales are difficult to predict because they belong to a chaotic system comprised of intrinsic and extrinsic characteristics, perceived value, and market speculation. We combine modern techniques of data science and machine learning with traditional spatial analysis to create robust real estate prediction models for both classification and regression tasks. We compare several cutting edge machine learning algorithms across spatial, semi-spatial and non-spatial feature engineering techniques, and we empirically show that spatially-conscious machine learning models outperform non-spatial models when married with advanced prediction techniques such as feed-forward artificial neural networks and gradient boosting machine models.

Keywords: Real estate, Artificial neural networks, Machine learning, Recommender systems, Supervised learning, Predictive modeling

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | The Spatially-Conscious Machine Learning Model | 1 |
| 1.2 | Motivation: Combating Income Inequality by Predicting Gentrification . . . | 1 |
| 2 | Literature Review | 3 |
| 2.1 | What is Economic Displacement? | 3 |
| 2.2 | A Review of Mass Appraisal Techniques | 4 |
| 2.3 | Predicting Gentrification Using Machine Learning | 6 |
| 3 | Data and Methodology | 9 |
| 3.1 | Methodology Overview | 9 |
| 3.2 | Data | 11 |
| 3.2.1 | Data Sources | 11 |
| 3.2.2 | Global Filtering of the Data | 13 |
| 3.2.3 | Exploratory Data Analysis | 13 |
| 3.3 | Feature Engineering | 18 |
| 3.3.1 | Base Modeling Data | 18 |
| 3.3.2 | Zip Code Modeling Data | 18 |
| 3.3.3 | Spatial Lag Modeling Data | 21 |
| 3.4 | Dependent Variables | 26 |
| 3.5 | Algorithms Comparison | 26 |
| 3.5.1 | Random Forest | 27 |
| 3.5.2 | Generalized Linear Model | 29 |
| 3.5.3 | Gradient Boosting Machine | 29 |
| 3.5.4 | Feed-Forward Artificial Neural Network | 30 |
| 3.6 | Model Validation | 31 |
| 3.7 | Evaluation Metrics | 32 |
| 3.7.1 | Area Under ROC Curve | 32 |
| 3.7.2 | Root Mean Squared Error | 33 |
| 4 | Results | 33 |
| 4.1 | Summary of Results | 33 |
| 4.2 | Stage 1) Random Forest Models Using All Data | 35 |
| 4.2.1 | Sale Price Regression Models | 35 |
| 4.2.2 | Probability of Sale Classification Models | 38 |
| 4.3 | Stage 2) Model Comparisons Using Specific Geographies and Building Types | 40 |
| 4.3.1 | Regression Model Comparisons | 40 |
| 4.3.2 | Classification Model Comparisons | 43 |
| 4.4 | Variable Importance Analysis of Top Performing Models | 46 |
| 5 | Future Research and Conclusions | 48 |
| 5.1 | Future Research | 48 |
| 5.2 | Conclusion | 49 |

1 Introduction

1.1 The Spatially-Conscious Machine Learning Model

Things near each other tend to be like each other. This concept is a well-known problem in traditional spatial analysis and is typically referred to as *spatial autocorrelation*. In statistics, this is said to “reduce the amount of information” pertaining to spatially proximate observations as they can, in part, be used to predict each other (DiMaggio, 2012). But can spatial features be used in a machine-learning context to make better predictions? This work demonstrates that the addition of “spatial lag” features to machine learning models significantly increases accuracy when predicting real estate sales and sale prices.

1.2 Motivation: Combating Income Inequality by Predicting Gentrification

Researchers at the Urban Institute (Greene, Pendall, Scott, & Lei, 2016) recently identified *economic exclusion* as a powerful contributor to income inequality in the United States. Economic exclusion can be defined as follows: vulnerable populations—disproportionately communities of color, immigrants, refugees, and women—who are physically displaced by local economic prosperity enter a continuous cycle of diminished access to good jobs, good schools, health care facilities, public spaces, and other economic and social resources. Diminished access leads to more poverty, which leads to more displacement. This self-reinforcing cycle of poverty and exclusion gradually exacerbates income inequality over the course years and generations. What can be done to intervene?

Stopping Economic Exclusion requires preventing displacement. *Displacement* can be thought of as the negative consequence of gentrification, where affordability pressures force vulnerable populations to relocate due to localized economic prosperity (Clay, 1979). Reliably predicting gentrification would be a valuable tool for preventing displacement at an early stage; however, such a task has proven difficult historically.

When an area experiences economic growth, increased housing demands and subsequent affordability pressures can lead to voluntary or involuntary relocation of low-income families and small businesses. Government agencies and nonprofits tend to intervene once displacement is already underway, and after-the-fact interventions can be costly and ineffective. Several preemptive actions exist which can be deployed to stem divestment and ensure that existing residents benefit from local prosperity. Potential interventions include job training, apprenticeships, subsidies, zoning laws, charitable aid, matched savings programs, financial literacy coaching, homeowner assistance, housing vouchers, and more (Greene et al., 2016). Not unlike medical treatment, early detection is the key to success.

Consequently the Urban Institute published a series of essays in 2016 outlining the few ways city governments employ “Big data and crowdsourced data” to identify vulnerable individuals and connect them with the proper services and resources, noting that “much more could be done” (Greene et al., 2016). This work seeks to further the research into using free, open data, open-source software and cutting-edge techniques to predict trends in gentrification, with the aim of identifying high-risk households and areas in need of intervention.

To date, many government agencies have demonstrated the benefits of applied predictive modeling, ranging from prescription drug abuse prevention to homelessness intervention to recidivism reduction (Ritter, 2013). However, few if any examples exist of large-scale, systematic applications of data analysis to aid vulnerable populations experiencing displacement. This paper belongs to an emerging trend known as the “science of cities” which aims to use large data sets and advanced simulation and modeling techniques to understand and improve urban patterns and how cities function (Batty, 2013).

This work explores techniques that can dramatically boost the accuracy of existing gentrification prediction models. We use real estate transactions in New York City, both their occurrence (probability of sale) and their dollar amount (sale price per square foot) as a proxy for gentrification. The technique marries the use of machine-learning predictive modeling with “spatial-lag” features typically seen in geographically-weighted regressions

(GWR). We employ a two-step modeling process in which we 1) determine the optimal building types and geographies suited to our feature engineering assumptions and 2) perform a comparative analysis across several state-of-the-art algorithms (generalized linear model, random forest, gradient boosting machine, and artificial neural network). We conclude that spatially-conscious machine learning models consistently outperform traditional real estate valuation and predictive modeling techniques.

2 Literature Review

This literature review discusses the academic study of economic displacement, primarily as it relates to gentrification. We also examine *mass appraisal techniques*, which are automated analytical techniques used for valuing large numbers of real estate properties. Finally, we examine recent applications of machine learning as it relates to predicting gentrification.

2.1 What is Economic Displacement?

Economic displacement has been intertwined with the study of gentrification since shortly after the latter became academically relevant in the 1960s. The term gentrification was first used in 1964 to describe the *gentry* in low-income neighborhoods in London (Glass, 1964). Originally, academics described gentrification in predominantly positive terms as a “tool of revitalization” for declining neighborhoods (Zuk et al., 2015). However, by 1979 the negative consequences of gentrification became more well understood, especially with regards to economic exclusion (Clay, 1979). Today, the term has a more neutral connotation, describing the placement and distribution of populations (Zuk et al., 2015). Specific to cities, recent literature defines gentrification as “the transformation of a working-class or vacant area of the central city into middle-class residential or commercial use” (Lees, Slater, & Wyly, 2013).

Studies of gentrification and displacement generally take two approaches in the literature:

supply-side and demand-side, or “the flows of capital versus flows of people to neighborhoods,” respectively (Zuk et al., 2015). Supply-side arguments for gentrification tend to focus on investments and policies (Zuk et al., 2015) and are much more often the subject of academic literature on economic displacement. This kind of research may be more common because it has the advantage of being more directly linked to influencing public policy. According to Dreier, Mollenkopf, & Swanstrom (2004), public policies that can increase economic displacement have been, among others, automobile-oriented transportation infrastructure spending and mortgage interest tax deductions for homeowners. Others who have argued for supply-side gentrification include Smith (1979), who stated that the return of capital from the suburbs to the city, or the “political economy of capital flows into urban areas” are what primarily drive both the positive and negative consequences of urban gentrification.

More recently, researchers have explored economic displacement as a contributor to income inequality (Reardon & Bischoff, 2011; Watson, 2009). Wealthy households tend to influence local political processes to reinforce exclusionary practices. The exercising of political influence by prosperous residents results in a feedback loop producing downward economic pressure on households who lack such resources and influence. Gentrification prediction tools could be used to help break such feedback loops through early identification and intervention.

Many studies conclude that gentrification in most forms leads to exclusionary economic displacement; however, Zuk et al. (2015) characterizes the results of many recent studies as “mixed, due in part to methodological shortcomings.” This work attempts to further the understanding of gentrification prediction by demonstrating a technique to better predict real estate sales in New York City.

2.2 A Review of Mass Appraisal Techniques

Much research on predicting real estate prices has been in service of creating mass appraisal models. Local governments most commonly use mass appraisal models to assign

taxable values to properties. Mass appraisal models share many characteristics with predictive machine learning models in that they are data-driven, standardized methods that employ statistical testing (Eckert, 1990). A variation on mass appraisal models are the *automated valuation models* (AVM). Both mass appraisal models and AVMs seek to estimate the market value of a single property or several properties through data analysis and statistical modeling (d’Amato & Kauko, 2017).

Scientific mass appraisal models date back to 1936 with the reappraisal of St. Paul, Minnesota (Joseph, n.d.). Since that time, and accelerating with the advent of computers, much statistical research has been done relating property values and rent prices to various characteristics of those properties, including characteristics of their surrounding area. Multiple regression analysis (MRA) has been the most common set of statistical tools used in mass appraisal, including maximum likelihood, weighted least squares, and the most popular, ordinary least squares, or *OLS* (d’Amato & Kauko, 2017). The primary drawbacks of MRA techniques are excessive multicollinearity among attributes and spatial autocorrelation among residuals (d’Amato & Kauko, 2017). Another group of models that seek to correct for spatial dependence are known as spatial auto-regressive models (SAR), chief among them the spatial lag model, which aggregates weighted summaries of nearby properties to create independent regression variables (d’Amato & Kauko, 2017).

So-called *hedonic regression models* seek to decompose the price of a good based on the intrinsic and extrinsic components. Koschinsky, Lozano-Gracia, & Piras (2012) is a recent and thorough discussion of parametric hedonic regression techniques. Koschinsky derives some of the variables included in his models from nearby properties, similar to the techniques used in this work, and these spatial variables were found to be predictive. The basic real estate hedonic model explains the price of a given property as:

$$P_i = P(q_i, S_i, N_i, L_i)$$

where P_i represents the price of house i , which is a “composite good” made up of a vector

of structural characteristics S , a vector of social and neighborhood characteristics N , and a vector of locational characteristics L (Koschinsky et al., 2012 pg. 322). Specifically, the model calculates spatial lags on properties of interest using neighboring properties within 1,000 feet of a sale. The derived variables include characteristics like average age, the number of poor condition homes, percent of homes with electric heating, construction grades, and more, within 1,000 feet of the property in question. Koschinsky found that in all cases homes near each other were typically similar to each other and priced accordingly. Koschinsky concluded that locational characteristics should be valued at least as much “if not more” than intrinsic structural characteristics.

As recently as 2015, much research has dealt with mitigating the drawbacks of MRA, including the use of multi-level hierarchical models. Fotheringham, Crespo, & Yao (2015) explored the combination of geographically weighted regression (GWR) with time-series forecasting to predict home prices over time. GWR is a variation on OLS that allows for adaptive bandwidths of local data to be included, i.e., for each estimate, the number of data points included varies and can be optimized using cross-validation.

2.3 Predicting Gentrification Using Machine Learning

Both mass appraisal techniques and AVMs seek to predict real estate prices using data and statistical methods; however, traditional techniques typically fall short. These techniques fail partly because property valuation is inherently a “chaotic” process that cannot be modeled effectively using linear methods (Zuk et al., 2015). The value of any given property is a complex combination of fungible intrinsic characteristics, perceived value, and speculation. The value of any building or plot of land belongs to a rich network where decisions about and perceptions of neighboring properties influence the final market value. Guan, Shi, Zurada, & Levitan (2014) compared traditional MRA techniques to alternative data mining techniques resulting in mixed results. However, as Helbich, Jochem, Mücke, & Höfle (2013) states, hedonic pricing models might be improved upon in two primary ways: through novel

estimation techniques, and by ancillary structural, locational, and neighborhood variables. Recent research generally falls into these two buckets: better algorithms and better data.

In the “better data” category, researchers have been striving to introduce new independent variables to increase the accuracy of predictive models. Alexander Dietzel, Braun, & Schäfers (2014) successfully used internet search query data provided by Google Trends to serve as a sentiment indicator and improve commercial real estate forecasting models. Pivo & Fisher (2011) examined the effects of walkability on property values and investment returns. Pivo found that on a 100-point scale, a 10-point increase in walkability increased property investment values by up to 9%.

Research into better prediction algorithms does not necessarily happen at the exclusion of better data. For example, Fu et al. (2014) created a prediction algorithm, called *ClusRanking*, for real estate in Beijing, China. ClusRanking first estimates neighborhood characteristics using taxi cab traffic vector data, including relative access to business areas. Then, the algorithm performs a rank-ordered prediction of investment returns segmented into five categories. Similar to Koschinsky et al. (2012), though less formally stated, Fu et al. (2014) thought of a property’s value as a composite of individual, peer and zone characteristics. In their predictive modeling, Fu included characteristics of the neighborhood (individual), the values of nearby properties (peer), and the prosperity of the affiliated latent business area (zone) based on taxi cab data (Fu et al., 2014).

Several other recent studies compare various advanced statistical techniques and algorithms either to other advanced techniques or to traditional ones. Most studies conclude that the advanced, non-parametric techniques outperform traditional parametric techniques, while several conclude that the Random Forest algorithm is particularly well-suited to predicting real estate values.

Kontrimas & Verikas (2011) compares the accuracy of linear regression against the SVM technique and found the latter to outperform. Schernthanner, Asche, Gonschorek, & Scheele (2016) compared traditional linear regression techniques to several techniques

such as kriging (stochastic interpolation) and Random Forest. They concluded that the more advanced techniques, particularly Random Forest, are sound and more accurate when compared to traditional statistical methods. Antipov & Pokryshevskaya (2012) came to a similar conclusion about the superiority of Random Forest for real estate valuation after comparing 10 algorithms: multiple regression, CHAID, exhaustive CHAID, CART, 2 types of k-nearest neighbors, multilayer perceptron artificial neural network, radial basis function neural network, boosted trees and finally Random Forest.

Guan et al. (2014) compared three different approaches to defining spatial neighbors: a simple radius technique, a k-nearest neighbors technique using only distance and a k-nearest neighbors technique using all attributes. Interestingly, the location-only KNN models performed best, although by a slight margin. Park & Bae (2015) developed several housing-price prediction models based on machine learning algorithms including C4.5, RIPPER, naive Bayesian, and AdaBoost, finding that the RIPPER algorithm consistently outperformed the other models. Rafiei & Adeli (2015) employed a restricted Boltzmann machine (neural network with back propagation) to predict the sale price of residential condos in Tehran, Iran, using a non-mating genetic algorithm for dimensionality reduction with a focus on computational efficiency. The paper concludes that two primary strategies help in this regard: weighting property sales by temporal proximity (i.e., sales which happened closer in time are more alike), and also using a learner to accelerate the recognition of important features. The paper compares this technique to several other common neural network approaches and finds that while not necessarily the only way to get the best answer, it is an efficient and fast way to get to the best answer.

Finally, we note that many studies, whether exploring advanced techniques, new data, or both, rely on aggregation of data by some arbitrary boundary. For example, Turner (2001) predicted gentrification in the Washington, D.C. metro area by ranking census tracts in terms of development. Chapple (2009) created a gentrification early warning system by identifying low-income census tracts in central city locations. Pollack, Bluestone, & Billingham (2010)

analyzed 42 census block groups near rail stations in 12 metro areas across the United States, studying changes between 1990 and 2000 for neighborhood socioeconomic and housing characteristics. All of these studies, and many more, relied on the aggregation of data at the census-tract or census-block level. In contrast, this paper compares boundary-aggregation techniques (specifically, aggregating by Zip Codes) to spatial lag techniques and finds the spatial lag techniques to generally outperform.

3 Data and Methodology

3.1 Methodology Overview

Our goal was to compare “spatially-conscious” machine learning predictive models to traditional feature engineering techniques. To accomplish this comparison, we created three separate modeling datasets:

- **Base modeling data:** includes building characteristics such as size, taxable value, usage, and others
- **Zip code modeling data:** includes the base data as well as aggregations of data at the zip-code level
- **Spatial lag modeling data:** includes the base data as well as aggregations of data within 500-meters of each building

The second and third modeling datasets are incremental variations of the first, using competing feature engineering techniques to extract additional predictive power from the data. We combined three open-source data repositories provided by New York City via nyc.gov and data.cityofnewyork.us. Our base modeling dataset included all building records and associated sales information from 2003-2017. For each of the three modeling datasets, we also compared two predictive modeling tasks, using a different dependent variable for each:

- 1) **Classification task: probability of sale** The probability that a given property will sell in a given year (0,1)

Table 3.1: Six Predictive Models

| # | Model | Model Task | Data | Outcome Var | Outcome Type | Eval 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 Lag | 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 |

- 2) **Regression task: sale-price-per-square-foot** Given that a property sells, how much is the sale price per square foot? (\$/SF)

Table 3.1 shows the six distinct modeling task/data combinations.

We conducted our analysis in a two-stage process. In Stage 1, we used the Random Forest algorithm to evaluate the suitability of the data for our feature engineering assumptions. In Stage 2, we created subsets of the modeling data based on the analysis conducted in Stage 1. We then compared the performance of different algorithms across all modeling datasets and prediction tasks. The following is an outline of our complete analysis process:

Stage 1: Random Forest algorithm using all data

- 1) Create a “base” modeling dataset by sourcing and combining building characteristic and sales data from open-source New York City repositories
- 2) Create a “zip” modeling dataset by aggregating the base data at a Zip-Code level and appending these features to the base data
- 3) Create a “spatial lag” modeling dataset by aggregating the base data within 500 meters of each building and appending these features to the base data
- 4) Train a Random Forest model on all three datasets, for both classification (probability of sale) and regression (sale price) tasks
- 5) Evaluate the performance of the various Random Forest models on hold-out test data
- 6) Analyze the prediction results by building type and geography, identifying those buildings for which our feature-engineering assumptions (e.g., 500-meter radii spatial

lags) are most appropriate

Stage 2: Many algorithms using refined data

- 7) Create subsets of the modeling data based on analysis conducted in Stage 1
- 8) Train machine learning models on the refined modeling datasets using several algorithms, for both classification and regression tasks
- 9) Evaluate the performance of the various models on hold-out test data
- 10) Analyze the prediction results of the various algorithm/data/task combinations

3.2 Data

3.2.1 Data Sources

The New York City government makes available an annual dataset which describes all tax lots in the five boroughs. The Primary Land Use and Tax Lot Output dataset, known as PLUTO¹, contains a single record for every tax lot in the city along with a number of building-related 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 had made this dataset available for all years between 2002-2017, excluding 2008. For convenience, we also exclude the 2002 dataset from our analysis because corresponding 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 were interested in both the occurrence and the amount of real estate sales transactions. Sales transactions are made available separately by the New York City government, known as the NYC Rolling Sales Data². At the time of this writing, sales transactions were available for the years 2003-2017. The sales transactions data contains

¹<https://www1.nyc.gov/site/planning/data-maps/open-data/bytes-archive.page?sorts%5Byear%5D=0>

²<http://www1.nyc.gov/site/finance/taxes/property-annualized-sales-update.page>

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, we first had to transform the sales data 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 BBL's. Mapping the sales data directly to the PLUTO data results in a mapping error rate of 23.1% (mainly due to condos). Therefore, the sales transactions data must first be mapped to another data source, the NYC Property Address Directory, or PAD³, which contains an exhaustive list of all BBL's in NYC. After combining the sales data with PAD, the data can then be mapped to PLUTO with an error rate of 0.291% (See: Figure 3.1).

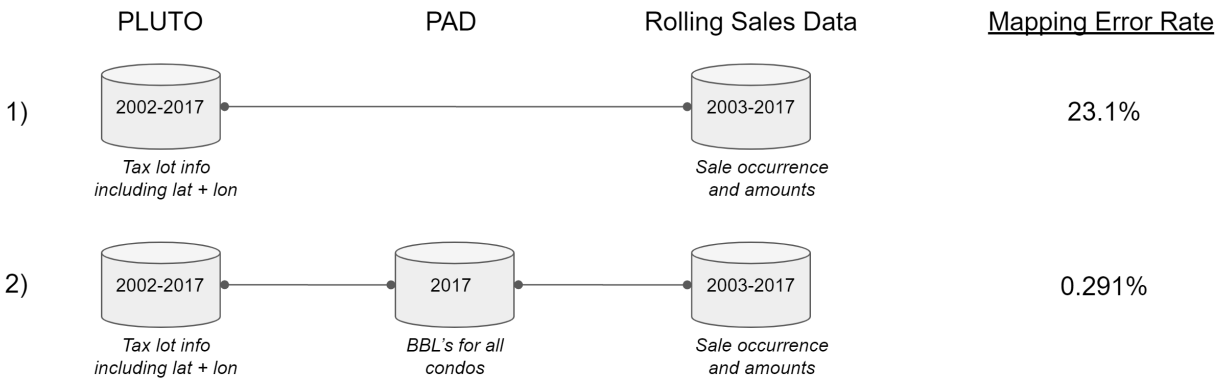


Figure 3.1: Overview of Data Sources

After combining the Sales Transactions data with PAD and PLUTO, the resulting data were filtered so that only BBL's with less than or equal to 1 transactions in a year occur. The

³<https://data.cityofnewyork.us/City-Government/Property-Address-Directory/bc8t-ecyu/data>

final dataset 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. A description of all variables can be seen in Table 3.2.

3.2.2 Global Filtering of the Data

We only included building categories of significant interest in our initial modeling data. Generally speaking, by significant interest we are referring to building types that are regularly bought and sold on the free market. These include residences, office buildings and industrial buildings, and exclude things like government-owned buildings and hospitals. We also excluded hotels as they tend to be comparatively rare in the data and exhibit unique sales characteristics. The included building types are displayed in Table 3.3.

The data were further filtered to include only records with equal to or less than 2 buildings per tax lot, effectively excluding large outliers in the data such as the World Trade Center and Stuyvesant Town. The global filtering of the dataset reduces the base modeling data from 12,012,780 records down to 8,247,499, retaining 68.6%% of the original data.

3.2.3 Exploratory Data Analysis

The data contain building and sale records across the five boroughs of New York City for the years 2003-2017. One challenge with creating a predictive model of real estate sales data is the heterogeneity within the data in terms of frequency of sales and sale price. These two metrics (sale occurrence and amount) vary meaningfully across year, borough and building classes (among other attributes). Table 3.4 displays statistics which describe the base dataset (pre-filtered) by year. Note how the frequency of transactions (# of Sales) and the sale amount (Median Sale \$/SF) tend to covary, particularly through the downturn of 2009-2012. This covariance may be due to the fact that the relative size of transactions tends to decrease as capital becomes more constrained.

We observe similar variances across asset types. Table 3.5 shows all buildings classes in

Table 3.2: Description of Base Data

| variable | type | nobs | mean | sd | mode | min | max | median | n_missing |
|--------------------------------|-----------|------------|---------|------------|---------|-------|---------------|---------|------------|
| Annual_Sales | Numeric | 12,012,780 | 2 | 8 | NA | 1 | 2,591 | 1 | 11,208,593 |
| AssessLand | Numeric | 12,012,780 | 93,493 | 2,870,654 | 103,050 | 0 | 2,146,387,500 | 10,348 | 65 |
| AssessTot | Numeric | 12,012,780 | 302,375 | 4,816,339 | 581,400 | 0 | 2,146,387,500 | 25,159 | 1,703,150 |
| BldgArea | Numeric | 12,012,780 | 6,228 | 70,161 | 18,965 | 0 | 49,547,830 | 2,050 | 45 |
| BldgDepth | Numeric | 12,012,780 | 46 | 34 | 50 | 0 | 9,388 | 42 | 44 |
| BldgFront | Numeric | 12,012,780 | 25 | 33 | 100 | 0 | 9,702 | 20 | 44 |
| Block | Numeric | 12,012,780 | 5,297 | 3,695 | 1 | 0 | 71,724 | 4,799 | 44 |
| BoroCode | Numeric | 12,012,780 | 3 | 1 | 5 | 1 | 5 | 4 | 47 |
| BsmtCode | Numeric | 12,012,780 | 2 | 2 | 0 | 0 | 3,213 | 2 | 859,406 |
| BuiltFAR | Numeric | 12,012,780 | 1 | 10 | 3 | 0 | 8,695 | 1 | 850,554 |
| ComArea | Numeric | 12,012,780 | 2,160 | 58,192 | 18,965 | 0 | 27,600,000 | 0 | 44 |
| CommFAR | Numeric | 12,012,780 | 0 | 1 | 3 | 0 | 15 | 0 | 7,716,603 |
| CondoNo | Numeric | 12,012,780 | 8 | 126 | 0 | 0 | 30,000 | 0 | 1,703,113 |
| Easements | Numeric | 12,012,780 | 0 | 2 | 0 | 0 | 7,500 | 0 | 48 |
| ExemptLand | Numeric | 12,012,780 | 37,073 | 2,718,194 | 0 | 0 | 2,146,387,500 | 1,290 | 65 |
| ExemptTot | Numeric | 12,012,780 | 107,941 | 3,522,172 | 0 | 0 | 2,146,387,500 | 1,360 | 1,703,149 |
| FacilFAR | Numeric | 12,012,780 | 2 | 2 | 5 | 0 | 15 | 2 | 7,716,603 |
| FactryArea | Numeric | 12,012,780 | 126 | 3,890 | 0 | 0 | 1,324,592 | 0 | 850,555 |
| GarageArea | Numeric | 12,012,780 | 130 | 5,154 | 0 | 0 | 2,677,430 | 0 | 850,554 |
| GROSS SQUARE FEET | Numeric | 12,012,780 | 4,423 | 45,691 | NA | 0 | 14,962,152 | 1,920 | 11,217,669 |
| lat | Numeric | 12,012,780 | 41 | 0 | 41 | 40 | 41 | 41 | 427,076 |
| lon | Numeric | 12,012,780 | -74 | 0 | -74 | -78 | -74 | -74 | 427,076 |
| Lot | Numeric | 12,012,780 | 115 | 655 | 10 | 0 | 9,999 | 38 | 44 |
| LotArea | Numeric | 12,012,780 | 7,852 | 362,618 | 5,716 | 0 | 214,755,710 | 2,514 | 44 |
| LotDepth | Numeric | 12,012,780 | 104 | 69 | 84 | 0 | 9,999 | 100 | 45 |
| LotFront | Numeric | 12,012,780 | 40 | 74 | 113 | 0 | 9,999 | 25 | 44 |
| LotType | Numeric | 12,012,780 | 5 | 1 | 5 | 0 | 9 | 5 | 865,340 |
| NumBldgs | Numeric | 12,012,780 | 1 | 4 | 1 | 0 | 2,740 | 1 | 46 |
| NumFloors | Numeric | 12,012,780 | 2 | 2 | 4 | 0 | 300 | 2 | 44 |
| OfficeArea | Numeric | 12,012,780 | 742 | 21,566 | 0 | 0 | 5,009,319 | 0 | 850,556 |
| OtherArea | Numeric | 12,012,780 | 673 | 49,848 | 0 | 0 | 27,600,000 | 0 | 850,555 |
| ProxCode | Numeric | 12,012,780 | 1 | 2 | 1 | 0 | 5,469 | 1 | 197,927 |
| ResArea | Numeric | 12,012,780 | 3,921 | 31,882 | 0 | 0 | 35,485,021 | 1,776 | 44 |
| ResidFAR | Numeric | 12,012,780 | 1 | 1 | 2 | 0 | 12 | 1 | 7,716,603 |
| RetailArea | Numeric | 12,012,780 | 309 | 14,394 | 6,965 | 0 | 21,999,988 | 0 | 850,554 |
| SALE PRICE | Numeric | 12,012,780 | 884,036 | 13,757,706 | NA | 0 | 4,111,111,766 | 319,000 | 11,208,593 |
| sale_psf | Numeric | 12,012,780 | 220 | 5,153 | NA | 0 | 1,497,500 | 114 | 11,250,396 |
| SALE_YEAR | Numeric | 12,012,780 | 2,009 | 5 | NA | 2,003 | 2,017 | 2,009 | 11,208,593 |
| Sold | Numeric | 12,012,780 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| StrgeArea | Numeric | 12,012,780 | 169 | 5,810 | 12,000 | 0 | 1,835,150 | 0 | 850,554 |
| TOTAL_SALES | Numeric | 12,012,780 | 884,036 | 13,757,706 | NA | 0 | 4,111,111,766 | 319,000 | 11,208,593 |
| UnitsRes | Numeric | 12,012,780 | 4 | 36 | 0 | 0 | 20,811 | 1 | 45 |
| UnitsTotal | Numeric | 12,012,780 | 4 | 42 | 1 | 0 | 44,276 | 2 | 47 |
| Year | Numeric | 12,012,780 | 2,010 | 4 | 2,017 | 2,003 | 2,017 | 2,011 | 0 |
| YearAlter1 | Numeric | 12,012,780 | 159 | 540 | 2,000 | 0 | 2,017 | 0 | 45 |
| YearAlter2 | Numeric | 12,012,780 | 20 | 202 | 0 | 0 | 2,017 | 0 | 48 |
| YearBuilt | Numeric | 12,012,780 | 1,830 | 449 | 1,884 | 0 | 2,040 | 1,930 | 47 |
| ZipCode | Numeric | 12,012,780 | 11,007 | 537 | 10,301 | 0 | 11,697 | 11,221 | 59,956 |
| Address | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 17,902 |
| AssessTotal | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 10,309,712 |
| bbl | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 0 |
| BldgClass | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 16,372 |
| Borough | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 0 |
| BUILDING CLASS AT PRESENT | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 11,219,514 |
| BUILDING CLASS AT TIME OF SALE | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 11,208,593 |
| BUILDING CLASS CATEGORY | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 11,208,765 |
| Building_Type | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 16,372 |
| CornerLot | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 11,163,751 |
| ExemptTotal | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 10,309,712 |
| FAR | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 11,162,270 |
| IrrLotCode | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 16,310 |
| MaxAllwFAR | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 4,296,221 |
| OwnerName | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 137,048 |
| OwnerType | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 10,445,328 |
| TAX CLASS AT PRESENT | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 11,219,514 |
| TAX CLASS AT TIME OF SALE | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 11,208,593 |
| ZoneDist1 | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 18,970 |
| ZoneDist2 | Character | 12,012,780 | NA | NA | NA | NA | NA | NA | 11,715,653 |

Table 3.3: Included Building Category Codes

| 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 |

Table 3.4: Sales By Year

| Year | N | # Sales | Median Sale | Median Sale \$/SF |
|------|--------|---------|-------------|-------------------|
| 2003 | 850515 | 78919 | \$218,000 | \$79.37 |
| 2004 | 852563 | 81794 | \$292,000 | \$124.05 |
| 2005 | 854862 | 77815 | \$360,500 | \$157.76 |
| 2006 | 857473 | 70928 | \$400,000 | \$168.07 |
| 2007 | 860480 | 61880 | \$385,000 | \$139.05 |
| 2009 | 860519 | 43304 | \$245,000 | \$41.25 |
| 2010 | 860541 | 41826 | \$273,000 | \$75.35 |
| 2011 | 860320 | 40852 | \$263,333 | \$56.99 |
| 2012 | 859329 | 47036 | \$270,708 | \$52.72 |
| 2013 | 859372 | 50408 | \$315,000 | \$89.44 |
| 2014 | 858914 | 51386 | \$350,000 | \$115.71 |
| 2015 | 859464 | 53208 | \$375,000 | \$135.62 |
| 2016 | 859205 | 53772 | \$385,530 | \$147.06 |
| 2017 | 859223 | 51059 | \$430,000 | \$171.71 |

Table 3.5: Sales By Asset Class

| Bldg Code | Build Type | N | # Sales | Median Sale | Median Sale \$/SF |
|-----------|----------------------|---------|---------|-------------|-------------------|
| A | One Family Dwellings | 4435615 | 252283 | \$320,000 | \$215.85 |
| B | Two Family Dwellings | 3431762 | 219492 | \$340,000 | \$155.79 |
| C | Walk Up Apartments | 1873447 | 135203 | \$330,000 | \$67.20 |
| D | Elevator Apartments | 188689 | 45635 | \$398,000 | \$4.69 |
| E | Warehouses | 84605 | 5126 | \$200,000 | \$31.48 |
| F | Factory | 67174 | 4440 | \$350,000 | \$56.44 |
| G | Garages | 221620 | 13965 | \$0 | \$78.57 |
| H | Hotels | 10807 | 619 | \$5,189,884 | \$184.82 |
| I | Hospitals | 17650 | 687 | \$600,000 | \$62.66 |
| J | Theatres | 2662 | 152 | \$113,425 | \$4.01 |
| K | Retail | 265101 | 14841 | \$200,000 | \$60.63 |
| L | Loft | 18239 | 1259 | \$1,937,500 | \$101.36 |
| M | Religious | 78063 | 1320 | \$375,000 | \$91.78 |
| N | Asylum | 8498 | 190 | \$275,600 | \$35.90 |
| O | Office | 93973 | 5294 | \$550,000 | \$143.29 |
| P | Public Assembly | 15292 | 437 | \$350,000 | \$85.47 |
| Q | Recreation | 55193 | 232 | \$0 | \$0 |
| R | Condo | 78188 | 40157 | \$444,750 | \$12.65 |
| S | Mixed Use Residence | 467555 | 29396 | \$250,000 | \$78.29 |
| T | Transportation | 4012 | 49 | \$0 | \$0 |
| U | Utility | 32802 | 129 | \$0 | \$175 |
| V | Vacant | 449667 | 29091 | \$0 | \$134.70 |
| W | Educational | 38993 | 704 | \$0 | \$0 |
| Y | Gov't | 7216 | 44 | \$21,451.50 | \$0.30 |
| Z | Misc | 49583 | 2740 | \$0 | \$0 |

the 2003-2017 period. Unsurprisingly, residences tend to have the highest volume of sales while offices tend to have the highest sale prices.

Sale price per square foot, in particular, varies considerably across geography and asset class. Table 3.6 shows the breakdown of sales prices by borough and asset class. Manhattan tends to command the highest sale-price-per-square-foot across asset types. “Commercial” asset types such as Office and Elevator Apartments tend to fetch much lower price-per-square foot than do residential classes such as one and two-family dwellings. Table 3.7 shows the number of transactions across the same dimensions.

Table 3.6: Sale Price Per Square Foot by Asset Class and Borough

| Build Type | BK | BX | MN | QN | SI |
|----------------------|----------|----------|----------|----------|----------|
| Elevator Apartments | \$2.65 | \$1.74 | \$10.80 | \$1.87 | \$1.23 |
| Factory | \$33.33 | \$53.19 | \$135.62 | \$92.42 | \$55.01 |
| Garages | \$78.94 | \$80.57 | \$94.43 | \$71.11 | \$67.46 |
| Loft | \$46.32 | \$78.26 | \$141.56 | \$150.37 | \$61.82 |
| Office | \$118.52 | \$123.04 | \$225.96 | \$148.45 | \$105 |
| One Family Dwellings | \$221.26 | \$176.98 | \$757.58 | \$232.69 | \$203.88 |
| Two Family Dwellings | \$140.95 | \$131.06 | \$296.10 | \$181.84 | \$160.76 |
| Walk Up Apartments | \$69.97 | \$84.05 | \$50.61 | \$36.94 | \$75.38 |

Table 3.7: Number of Sales by Asset Class and Borough

| Build Type | BK | BX | MN | QN | SI |
|----------------------|--------|--------|--------|---------|--------|
| Elevator Apartments | 8,377 | 4,252 | 23,641 | 9,196 | 169 |
| Factory | 2,265 | 453 | 109 | 1,520 | 93 |
| Garages | 5,386 | 2,659 | 1,097 | 4,000 | 823 |
| Loft | 119 | 21 | 1,108 | 8 | 3 |
| Office | 1,112 | 340 | 2,081 | 1,162 | 599 |
| One Family Dwellings | 45,009 | 17,508 | 1,654 | 126,333 | 61,779 |
| Two Family Dwellings | 83,547 | 25,920 | 1,566 | 83,940 | 24,519 |
| Walk Up Apartments | 63,552 | 18,075 | 19,824 | 31,932 | 1,820 |

3.3 Feature Engineering

3.3.1 Base Modeling Data

The base modeling dataset was constructed by combining several open-source data repositories, outlined in the Data Sources section. In addition to the data provided by New York City, several additional features were engineered and appended to the base data. A summary table of the additional features is presented in Table 3.8. A binary variable was created to indicate whether a tax lot had a building on it (i.e., whether it was an empty plot of land). In addition, building types were quantified by what percent of their square footage belonged to the major property types: Commercial, Residential, Office, Retail, Garage, Storage, Factory and Other.

Importantly, two variables were created from the sale prices: A price-per-square-foot-figure (“Sale_Price”) and a total sale price (“Sale_Price_Total”). Sale-price-per-square-foot eventually became the outcome variable in one of the predictive models. We then created a feature to carry forward the previous sale price of a tax lot, if there was one, through successive years. The previous sale price was then used to create simple moving averages (SMA), exponential moving averages (EMA), and percent change measurements between the moving averages. In total, 69 variables were input to the feature engineering process and 92 variables were output. The final base modeling dataset was 92 variables by 8,247,499 rows.

3.3.2 Zip Code Modeling Data

The first of the two comparative modeling datasets was the zip code modeling data. Using the base data as a starting point, several features were generated to describe characteristics of the zip code where each tax lot resides. A summary table of the zip code level features is presented in 3.9.

In general, the base model data features were aggregated to a zip code level and appended, including the SMA, EMA and percent change calculations. Additionally, a second set of

Table 3.8: Base Modeling Data Features

| 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 |
| 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 |

Table 3.9: Zip Code Modeling Data Features

| 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 | | |
| 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 |
| 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 |

features were added, denoted as “bt_only”, which aggregated the data in a similar fashion but only included tax lots of the same building type in the calculations. In total, the zip code feature engineering process input 92 variables and output 122 variables.

3.3.3 Spatial Lag Modeling Data

Spatial lags are variables created from physically proximate observations. For example, calculating the average age of all buildings within 100 meters of a tax lot constitutes 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. Intuitively, knowing the average sale price of all buildings within 500 meters of a building can be more informative than knowing the sale prices of all buildings in the same zip code. However, creating spatial lags is computationally expensive. In addition, it can be challenging to set a proper radius for the spatial lag calculation; in a city, 500 meters may be appropriate (for certain building types), whereas several kilometers or more may be appropriate for less densely populated areas. In this paper, we present a solution for the computational challenges and suggest a potential approach to solving the radius-choice problem.

3.3.3.1 Creating the Point-Nighbor Relational Graph

To build our spatial lags, for each point in the data, we must identify which of all other points in the data fall within a specified radius. This neighbor identification process requires iteratively running point-in-polygon operations. This process is conceptually illustrated in figure 3.2.

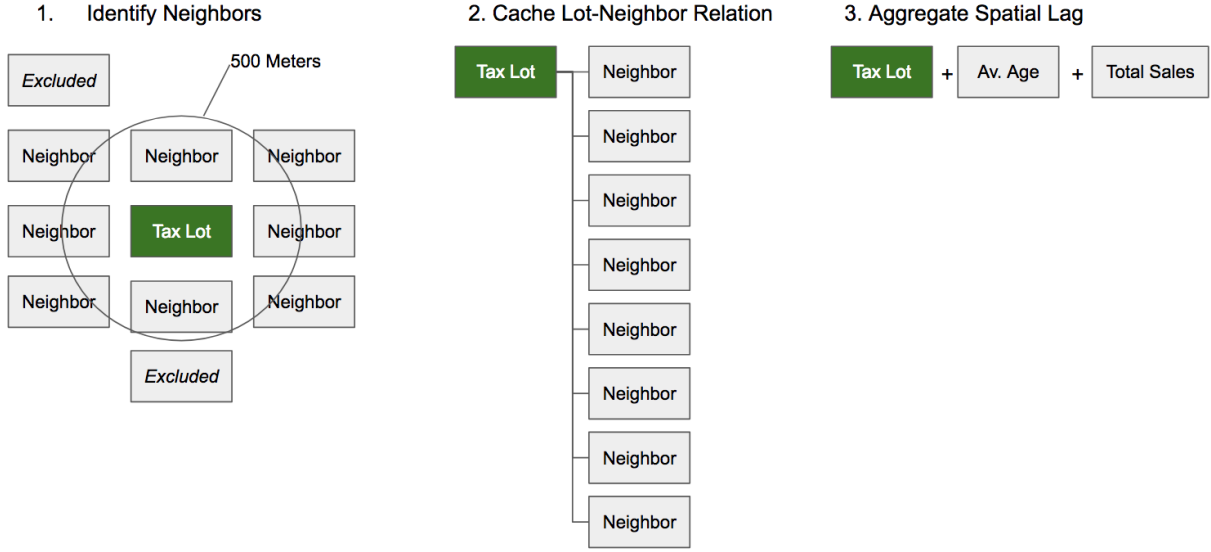


Figure 3.2: Spatial Lag Feature Creation Process

Given that, for every point q_i in our dataset, we need to determine whether every other point q_i falls within a given radius, this means that we can approximate the time-complexity of our operation as:

$$O(N(N - 1))$$

Since the number of operations approaches N^2 , calculating spatial lags for all 8,247,499 observations in our modeling data would be infeasible from a time and computation perspective. Assuming that tax lots rarely if ever move over time, we first reduced the task to the number of unique tax lots in New York City from 2003-2017, which is 514,124 points. Next, we implemented an indexing technique that greatly speeds up the process of creating a point-neighbor relational graph. The indexing technique both reduces the relative search space for each computation and also allows for parallelization of the point-in-polygon operations by dividing the data into a gridded space. The gridded spatial indexing process is outlined in Algorithm 1.

Each gridded partition of the data is married with a corresponding search space S , which

Algorithm 1 Gridded Spatial Indexing

```

1: for each grid partition  $G$  do
2:   Extract all points  $G_i$  contained within partition  $G$ 
3:   Calculate convex hull  $H(G)$  such that the buffer extends to distance  $d$ 
4:   Define Search space  $S$  as all points within Convex hull  $H(G)$ 
5:   Extract all points  $S_i$  contained within  $S$ 
6:   for each data point  $G_i$  do
7:     Identify all points in  $S_i$  that fall within  $abs(G_i + d)$ 
8:   end for
9: end for

```

is the convex hull of the partition space buffered by the maximum distance d . In our case, we buffered the search space by 500 meters. Choosing an appropriate radius for buffering presents an additional challenge in creating spatially-conscious machine learning predictive models. In this paper, we chose an arbitrary radius, and use a two-stage modeling process to test the appropriateness of that assumption. Future work may want to explore implementing an “adaptive bandwidth” technique using cross-validation to determine the optimal radius.

By partitioning the data into spatial grids, we were able to reduce the search space for each operation by an arbitrary number of partitions G . This improves the base run-time complexity to:

$$O(N(\frac{N-1}{G}))$$

By making G arbitrarily large (bounded by computational resources only), we reduced the runtime substantially. Furthermore, binning the operations into grids allowed us to parallelize the computation, further reducing the overall runtime. Figure 3.3 shows a comparison of computation times between the basic point-in-polygon technique and a sequential version of the gridded indexing technique. Note that the grid method starts as slower than the basic point-in-polygon technique due to pre-processing overhead, but quickly wins out in terms of speed as the complexity of the task increases. This graph also does not reflect the parallelization of the grid method, which further reduced the time required to calculate the point-neighbor relational graph.

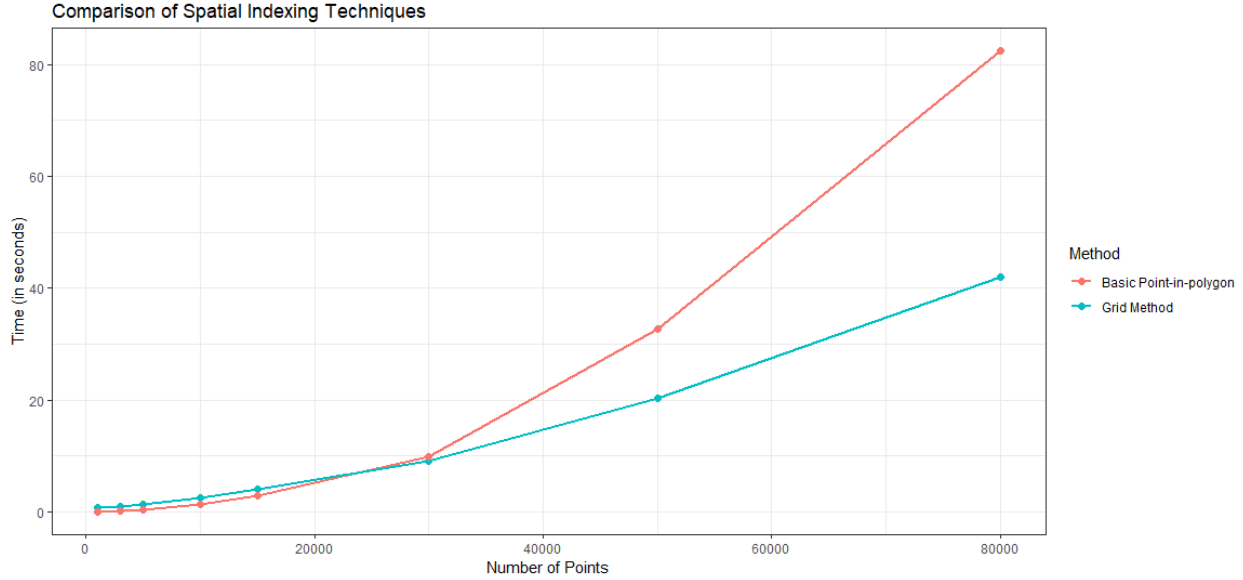


Figure 3.3: Spatial Index Time Comparison

3.3.3.2 Calculating Spatial Lags

Once we constructed the point-neighbor relational graph, we then used the graph to aggregate the data into spatial lag variables. One advantage of using spatial lags is the abundant number of potential features which can be engineered. Spatial lags can be weighted based on a distance function, e.g., physically closer observations can be given more weight. For our modeling purposes, we created two sets of features: inverse-distance weighted features (denoted with a "`__dist`" in Table 3.10) and simple average features (denoted with "`__basic`" in Table 3.10).

Temporal and spatial derivatives of the spatial lag features, presented in Table 3.10, were also added to the model, including: variables weighted by Euclidean distance ("dist"), basic averages of the spatial lag radius ("basic mean"), SMA for 2 years, 3 years and 5 years, EMA for 2 years, 3 years and 5 years, and year-over-year percent changes for all variables ("perc change"). In total, the spatial lag feature engineering process input 92 variables and output 194 variables.

Table 3.10: All Spatial Lag Features

| Feature | Min | Median | Mean | Max |
|--|----------|------------|------------|---------------|
| Radius_Total_Sold_In_Year | 1.00 | 20.00 | 24.00 | 201.00 |
| Radius_Average_Years_Since_Last_Sale | 1.00 | 4.43 | 4.27 | 14.00 |
| Radius_Res_Units_Sold_In_Year | 0.00 | 226.00 | 289.10 | 2,920.00 |
| Radius_All_Units_Sold_In_Year | 0.00 | 255.00 | 325.94 | 2,923.00 |
| Radius_SF_Sold_In_Year | 0.00 | 259,403.00 | 430,891.57 | 8,603,639.00 |
| Radius_Total_Sold_In_Year_sum_over_2_years | 2.00 | 41.00 | 48.15 | 256.00 |
| Radius_Average_Years_Since_Last_Sale_sum_over_2_years | 2.00 | 9.25 | 8.70 | 26.00 |
| Radius_Res_Units_Sold_In_Year_sum_over_2_years | 0.00 | 493.00 | 584.67 | 3,397.00 |
| Radius_All_Units_Sold_In_Year_sum_over_2_years | 1.00 | 555.00 | 660.67 | 4,265.00 |
| Radius_SF_Sold_In_Year_sum_over_2_years | 2,917.00 | 580,947.00 | 872,816.44 | 14,036,469.00 |
| Radius_Total_Sold_In_Year_percent_change | -0.99 | 0.00 | 0.27 | 77.00 |
| Radius_Average_Years_Since_Last_Sale_percent_change | -0.91 | 0.13 | 0.26 | 8.00 |
| Radius_Res_Units_Sold_In_Year_percent_change | -1.00 | -0.04 | | |
| Radius_All_Units_Sold_In_Year_percent_change | -1.00 | -0.04 | | |
| Radius_SF_Sold_In_Year_percent_change | -1.00 | -0.02 | | |
| Radius_Total_Sold_In_Year_sum_over_2_years_percent_change | -0.96 | -0.03 | 0.03 | 15.00 |
| Radius_Average_Years_Since_Last_Sale_sum_over_2_years_percent_change | -0.72 | 0.12 | 0.17 | 2.50 |
| Radius_Res_Units_Sold_In_Year_sum_over_2_years_percent_change | -1.00 | -0.04 | | |
| Radius_All_Units_Sold_In_Year_sum_over_2_years_percent_change | -0.99 | -0.04 | 0.12 | 84.00 |
| Radius_SF_Sold_In_Year_sum_over_2_years_percent_change | -0.98 | -0.04 | 0.18 | 361.55 |
| Percent_Com_dist | 0.00 | 0.04 | 0.07 | 0.56 |
| Percent_Res_dist | 0.00 | 0.46 | 0.43 | 0.66 |
| Percent_Office_dist | 0.00 | 0.01 | 0.03 | 0.48 |
| Percent_Retail_dist | 0.00 | 0.02 | 0.02 | 0.09 |
| Percent_Garage_dist | 0.00 | 0.00 | 0.00 | 0.27 |
| Percent_Storage_dist | 0.00 | 0.00 | 0.01 | 0.26 |
| Percent_Factory_dist | 0.00 | 0.00 | 0.00 | 0.04 |
| Percent_Other_dist | 0.00 | 0.00 | 0.00 | 0.09 |
| Percent_Com_basic_mean | 0.00 | 0.04 | 0.07 | 0.54 |
| Percent_Res_basic_mean | 0.00 | 0.46 | 0.43 | 0.66 |
| Percent_Office_basic_mean | 0.00 | 0.01 | 0.03 | 0.44 |
| Percent_Retail_basic_mean | 0.00 | 0.02 | 0.02 | 0.08 |
| Percent_Garage_basic_mean | 0.00 | 0.00 | 0.00 | 0.29 |
| Percent_Storage_basic_mean | 0.00 | 0.00 | 0.01 | 0.23 |
| Percent_Factory_basic_mean | 0.00 | 0.00 | 0.00 | 0.03 |
| Percent_Other_basic_mean | 0.00 | 0.00 | 0.00 | 0.04 |
| Percent_Com_dist_perc_change | -0.90 | 0.00 | 0.00 | 6.18 |
| Percent_Res_dist_perc_change | -0.50 | 0.00 | 0.03 | 36.73 |
| Percent_Office_dist_perc_change | -1.00 | 0.00 | | |
| Percent_Retail_dist_perc_change | -0.82 | 0.00 | | |
| Percent_Garage_dist_perc_change | -1.00 | 0.00 | | |
| Percent_Storage_dist_perc_change | -1.00 | -0.01 | | |
| Percent_Factory_dist_perc_change | -1.00 | 0.00 | | |
| Percent_Other_dist_perc_change | -1.00 | 0.00 | | |
| SMA_Price_2_year_dist | 0.00 | 400.01 | 496.30 | 3,816.57 |
| SMA_Price_3_year_dist | 0.00 | 396.94 | 492.00 | 3,816.57 |
| SMA_Price_5_year_dist | 8.83 | 425.55 | 515.29 | 3,877.53 |
| Percent_Change_SMA_2_dist | -0.13 | 0.03 | 552.33 | 804,350.67 |
| Percent_Change_SMA_5_dist | -0.09 | 0.02 | 317.46 | 322,504.58 |
| EMA_Price_2_year_dist | 0.00 | 378.63 | 475.54 | 3,431.17 |
| EMA_Price_3_year_dist | 8.83 | 382.25 | 476.05 | 3,296.46 |
| EMA_Price_5_year_dist | 7.88 | 386.34 | 468.91 | 2,813.34 |
| Percent_Change_EMA_2_dist | -0.09 | 0.06 | 346.51 | 480,829.57 |
| Percent_Change_EMA_5_dist | -0.02 | 0.06 | 303.55 | 273,458.42 |
| SMA_Price_2_year_basic_mean | 0.02 | 412.46 | 496.75 | 2,509.79 |
| SMA_Price_3_year_basic_mean | 0.02 | 409.00 | 492.43 | 2,509.79 |
| SMA_Price_5_year_basic_mean | 17.16 | 443.34 | 515.67 | 2,621.01 |
| Percent_Change_SMA_2_basic_mean | -0.13 | 0.04 | 543.51 | 393,749.99 |
| Percent_Change_SMA_5_basic_mean | -0.09 | 0.03 | 312.46 | 157,500.00 |
| EMA_Price_2_year_basic_mean | 0.02 | 390.30 | 475.96 | 2,259.21 |
| EMA_Price_3_year_basic_mean | 11.39 | 393.25 | 476.45 | 2,136.36 |
| EMA_Price_5_year_basic_mean | 15.30 | 402.06 | 469.09 | 1,848.27 |
| Percent_Change_EMA_2_basic_mean | -0.09 | 0.06 | 340.89 | 235,378.24 |
| Percent_Change_EMA_5_basic_mean | -0.02 | 0.06 | 296.78 | 133,547.59 |

Table 3.11: Distributions for 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 |

3.4 Dependent Variables

The final step in creating the modeling data was to define the dependent variables reflective of the prediction tasks; a binary variable for classification and a continuous variable for regression:

- 1) **Binary: Sold** whether a tax lot sold in a given year. Used in the Probability of Sale classification model.
- 2) **Continuous: Sale-Price-per-SF** The price-per-square-foot associated with a transaction, if a sale took place. Used in the Sale Price Regression model.

Table 3.11 describes the distributions of both outcome variables.

3.5 Algorithms Comparison

We implemented and compared several algorithms across our two-stage process. In Stage 1, the Random Forest algorithm was used to identify the optimal subset of building types and geographies for our spatial lag aggregation assumptions. In Stage 2, we analyzed the hold-out test performance of several algorithms including Random Forest, generalized linear model (GLM), gradient boosting machine (GBM), and feed-forward artificial neural network (ANN). Each algorithm was run over the three competing feature engineering datasets and for both the classification and regression tasks.

3.5.1 Random Forest

Random Forest was proposed by Breiman Breiman (2001) as an ensemble of prediction decision trees iteratively trained across randomly generated subsets of data. Algorithm 2 outlines the procedure (Hastie, Tibshirani, & Friedman, 2001).

Algorithm 2 Random Forest for Regression or Classification

1. For $b = 1$ to B
 - (a) Draw a bootstrap sample Z of the size N from the training data.
 - (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
 - i. Select m variables at random from the p variables
 - ii. Pick the best variable/split-point among the m .
 - iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x :

Regression: $\hat{f}_{rf}^B(x) = \frac{1}{B} \sum_{b=1}^B T_b(x)$

Classification: Let $\hat{C}_b(x)$ be the class prediction of the b th random-forest tree. Then $\hat{C}_{rf}^B(x) = \text{majority vote } \{\hat{C}_b(x)\}_1^B$

Previous works have found the Random Forest algorithm suitable to prediction tasks involving real estate (Antipov & Pokryshevskaya, 2012; Scherthanner et al., 2016). While algorithms exist that may outperform Random Forest in terms of predictive accuracy (such as neural networks and functional gradient descent algorithms), Random Forest is highly scalable and parallelizable, and is therefore an attractive choice for quickly assessing the predictive power of different feature engineering techniques. For these reasons and more outlined below, we selected Random Forest as the primary algorithm for Stage 1 of our modeling process.

Random Forest, like all predictive algorithms used in this work, suits 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 develop independently (and not sequentially, as is the case with most functional gradient descent algorithms), the tree building process can be executed in parallel. With a sufficiently large number of computer cores, the model training time can be significantly reduced.

We chose Random Forest as the algorithm for Stage 1 because:

- 1) The algorithm can be parallelized and is relatively fast compared to neural networks and functional gradient descent algorithms
- 2) Can accommodate 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 successfully built, and therefore, there is no need to filter out incomplete observations or impute missing values. Since much real estate 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, their influence is curtailed. Real estate data, especially with regards to pricing, tends to contain outliers. For example, the dependent variable in one of our models, sale price, 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 co-linearity in the data. This is highly valuable as real estate data can be highly correlated.

To run the model, we chose the `h2o.randomForest` implementation 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/>.

3.5.2 Generalized Linear Model

A generalized linear model (GLM) is an extension of the general linear model that estimates an independent variable y as the linear combination of one or more predictor variables. A GLM is made up of a linear predictor taking the form $\eta_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip}$, a link function that describes how the mean, $E(Y_i) = \mu_i$ depends on the linear predictor, and a variance function that describes how the variance, $\text{var}(Y_i)$ depends on the mean (Hoffmann, 2004). The observed value of the dependent variable y for observation i ($i = 1, 2, \dots, n$) is modeled as a linear function of $(p - 1)$ independent variables x_1, x_2, \dots, x_{p-1} as

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_{p-1} x_{i(p-1)} + e_i$$

Several family types of GLM's exist. For a binary independent variable, a binomial logistic regression is appropriate. For a continuous independent variable, the Gaussian or another distribution is appropriate. For our purposes, the Gaussian family is used for our regression task and binomial for the classification.

3.5.3 Gradient Boosting Machine

Gradient Boosting Machine (GBM) is one of the most popular machine learning algorithms available today. The algorithm uses iteratively refined approximations, obtained through cross-validation, to incrementally increase predictive accuracy. Similar to Random Forest, GBM is well-suited to using regression trees as the base learner. Gradient boosting constructs additive regression models by sequentially fitting a simple parameterized function (a *base learner*, in our case, a regression tree) to pseudo-residuals by least squares at each iteration (Friedman, 2002). The pseudo-residuals are the gradient of the loss function being

minimized, with respect to the model values at each training data point evaluated at the current step. The tree-variant of the generic gradient boosting algorithm is outlined in algorithm 3 (Hastie et al., 2001).

Algorithm 3 Gradient Tree Boosting Algorithm

1. Initialize: $f_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$.

2. For $m = 1$ to M :

(a) For $1, 2, \dots, N$ compute "pseudo-residuals":

$$r_{im} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}(x)}$$

(b) Fit a regression tree to the targets r_{im} giving terminal regions $R_{jm}, j = 1, 2, \dots, J_m$

(c) For $j = 1, 2, \dots, J_m$ compute:

$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$

(d) Update $f_m(x) = F_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$

3. Output $\hat{f}(x) = f_m(x)$

3.5.4 Feed-Forward Artificial Neural Network

The artificial neural network (ANN) implementation used in this work is a multi-layer feed-forward artificial neural network trained with stochastic gradient descent using back-propagation. Common synonyms for an ANN model are multi-layer perceptron and also deep neural network. The feed-forward ANN is one of the most common neural network algorithms, but other types exist, such as the convolutional neural network (CNN) which performs well on image classification tasks, and the recurrent neural network (RNN) which is well-suited for sequential data such as text and audio (Schmidhuber, 2015). The feed-forward ANN is typically best suited for tabular data.

The neural network model has unknown parameters, often called weights, and we seek values for them that make the model fit the training data well (Hastie et al., 2001). We denote the complete set of weights by θ , which consists of $\{\alpha_{0m}, \alpha_m; m = 1, 2, \dots, M\}M(p+1)$ weights and $\{\beta_{0k}, \beta_k; k = 1, 2, \dots, K\}K(p+1)$ weights.

For both classification and regression, we use sum-of-squared errors as our measure of fit (error function)

$$R(\theta) = \sum_{k=1}^K \sum_{i=1}^N (y_{ik} - f_k(x_i))^2$$

The generic approach to minimizing $R(\theta)$ is by gradient descent, called back-propagation in this setting (Hastie et al., 2001). Because of the compositional form of the model, the gradient can be derived using the chain rule for differentiation. This can be computed by a forward and backward sweep over the network, keeping track only of quantities local to each unit. Here is back-propagation in detail for squared error loss:

$$R(\theta) = \sum_{i=1}^N R_i$$

$$\sum_{i=1}^N \sum_{k=1}^K (y_{ik} - f_k(x_i))^2$$

For our implementations, we used the rectifier activation function with 1024 hidden layers, 100 epochs and L1 regularization set to 0.00001. The implementation we chose was the h2o.deeplearning open source R library. For more information, see: <https://www.h2o.ai/>.

3.6 Model Validation

The goal of our predictive modeling efforts were to be able to successfully predict both the probability and amount of real estate sales into the near future. As such, trained and evaluated our models using use out-of-time validation to assess performance. As shown in Figure 3.4 The models were trained using data from 2003-2015. We used 2016 data during the model training process for model validation purposes. Finally, we scored our models using 2017 data as a hold-out sample. Using out-of-time validation ensured that our models generalized well into the immediate future.

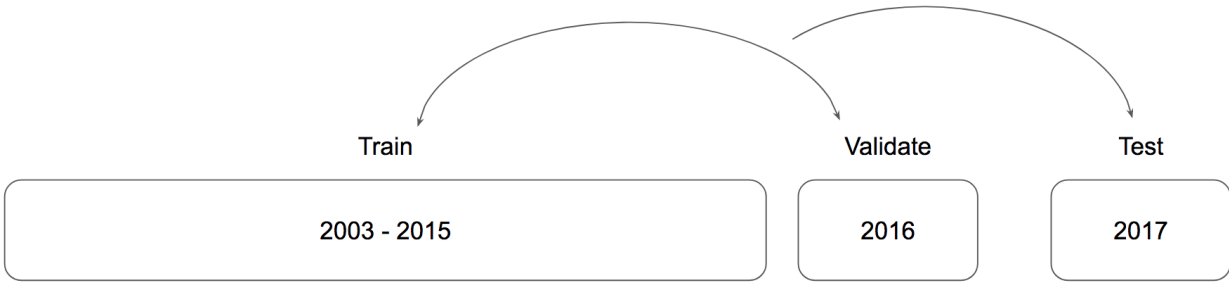


Figure 3.4: Out-of-time validation

3.7 Evaluation Metrics

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

3.7.1 Area Under ROC Curve

A classification model typically outputs a probability that a given case 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 “positive” 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 measurement of the performance of a classification model across all possible thresholds, and therefore sidesteps the need to arbitrarily assign a cutoff.

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 enumerates

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 it does not describe the trade-offs between false positives and false negatives. In certain circumstances, a false positive might be considerably less desirable than a false negative, or vice-versa. For our purposes, we rank false positives and false negatives as equally undesirable outcomes.

3.7.2 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 desirable. 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 datasets, and therefore can be reasonably compared using RMSE.

4 Results

4.1 Summary of Results

We have conducted comparative analyses across a two-stage modeling process. In Stage 1, using the Random Forest algorithm, we tested 3 competing feature engineering techniques (base, zip-code aggregation, and spatial-lag aggregation) for both a classification task (predicting the occurrence of a building sale) and a regression task (predicting the sale

price of a building). We analyzed the results of the first stage to identify which geographies and building types our model assumptions worked best. In Stage 2, using a subset of the modeling data (selected via an analysis of the output from Stage 1), we compared four algorithms – GLM, Random Forest, GBM and ANN – across our 3 competing feature engineering techniques for both classification and regression tasks. We analyzed the performance of the different model/data combos as well as conducted an analysis of the variable importances for the top performing models.

In Stage 1 (Random Forest, using all data), we found that models which utilized spatial features outperformed those models using zip-code features the majority of the time for both classification and regression. Of three models, the Sale Price Regression model using Spatial features finished 1st or 2nd 24.1% of the time (using RMSE as a ranking criterion), while the Zip Code Regression model finished in the top two spots only 11.2% of the time. Both models performed worse than the Base Regression model overall, which ranked in 1st or 2nd place 31.5% of the time. The story for the classification models was largely the same: the Spatial features tended to outperform the Zip Code data while the Base data won out overall. All models had similar performances on training data, but the Spatial and Zip Code datasets tended to underperform when generalizing to the hold-out test data, suggesting problems with overfitting.

We then analyzed the performance of both the regression and classification Random Forest models by geography and building type. We found that the models performed considerably better on Walk Up Apartments and Elevator Buildings (building types C and D) and in Manhattan, Brooklyn and the Bronx. Using these as filtering criteria, we created a subset of the modeling data for the subsequent modeling stage.

During Stage 2 (many algorithms using a subset of modeling data), we compared four algorithms across the same three competing feature engineering techniques using a filtered subset of the original modeling data. Unequivocally, the spatial features performed best across all models and tasks. For the classification task, the GBM algorithms performed best

Table 4.1: Sale Price Model Rankings, RMSE by Borough and Building Type

| Model Rank | 1 | 2 | 3 | Average Rank |
|-------------|-------|-------|-------|--------------|
| Base | 22.2% | 9.3% | 1.9% | 1.39 |
| Spatial Lag | 5.6% | 18.5% | 9.3% | 2.11 |
| Zip | 5.6% | 5.6% | 22.2% | 2.50 |

in terms of AUC, followed by ANN and Random Forest. For regression, the ANN algorithms performed best (as measured by RMSE as well as Mean Absolute Error and R-squared) with the spatial features ANN model performing best.

We conclude that spatial lag features can significantly increase the accuracy of machine learning-based real estate sale prediction models. We find that model overfitting presents a challenge when using spatial features, but that this can be overcome by implementing different algorithms, specifically ANN and GBM. Finally, we find that our implementation of spatial-lag features works best for certain kinds of buildings in specific geographic areas, and we hypothesize that this is due to the assumptions made when building the spatial features.

4.2 Stage 1) Random Forest Models Using All Data

4.2.1 Sale Price Regression Models

We analyzed the RMSE of the Random Forest models predicting Sale Price across feature engineering methods, Borough and Building Type. Table 4.1 displays the average ranking by model type as well as the distribution of models that ranked first, second and third for each respective Borough/Building Type combination. When we rank the models by performance for each Borough, Building Type combination, we find that the Spatial Lag models outperform the Zip Code models in 72% of cases with an average model-rank of 2.11 and 2.5, respectively.

The Base modeling dataset tends to outperform both enriched datasets, suggesting an issue with model overfitting in some areas. We see further evidence of overfitting in Table

Table 4.2: Sale Price Model RMSE For Validation and Test Hold-out Data

| type | base | zip | spatial lag |
|------------|--------|--------|-------------|
| Validation | 280.63 | 297.97 | 286.23 |
| Test | 287.83 | 300.60 | 297.92 |

4.2 where, despite similar performances on Validation data, the Zip and Spatial models have higher validation-to-test-set spreads. Despite this, the Spatial Lag features outperform all other models in specific locations, notably in Manhattan as shown in Figure 4.1.

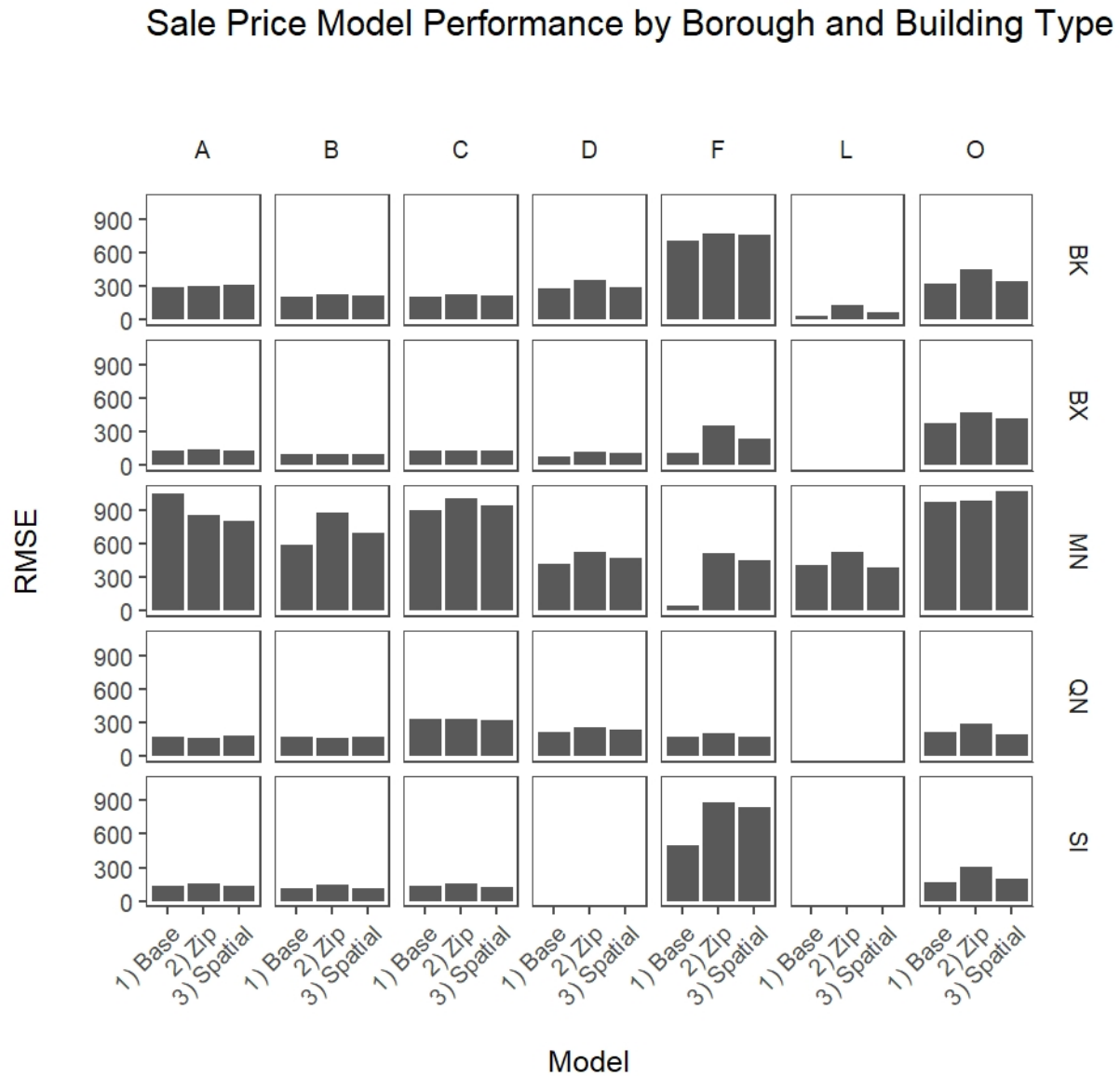


Figure 4.1: RMSE By Borough and Building Type

Figure 4.1 displays test RMSE by model, faceted by Borough on the y-axis and Building Type on the x-axis (See Table 3.3 and Table 3.5 for a description of building type codes). We make the following observations from Figure 4.1:

- The Spatial modeling data outperforms both Base and Zip Code in 6 cases, notably for Type A buildings (One Family Dwellings) and Type L buildings (Lofts) in Manhattan as well as Type O Buildings (Office) in Queens

Table 4.3: Probability of Sale Model AUC

| Model AUC | Base | Zip | Spatial Lag |
|------------|-------|-------|-------------|
| Validation | 0.832 | 0.829 | 0.829 |
| Test | 0.830 | 0.825 | 0.828 |

- The “residential” building Types A (One-family Dwellings), B (Two Family Dwellings), C (Walk Up Apartments) and D (Elevator Apartments) have generally lower RMSE scores compared to the non-residential types
- Spatial features perform best in Brooklyn, Bronx and Manhattan and for residential building types

4.2.2 Probability of Sale Classification Models

Similar to the results of the Sale Price regression models we find the Spatial models perform better on the hold-out test data compared to the Zip Code data when using Area Under the ROC Curve (AUC) as an evaluation metric, as shown in Table 4.3. The Base Modeling data continues to outperform the Spatial and Zip Code data overall.

Figure 4.2 shows a breakdown of model AUC faceted along the x-axis by Building Type and along the y-axis by Borough. The coloring indicates by how much a model’s AUC diverges from the cell average, which is useful for spotting over performers. We make the following observations about Figure 4.2:

- The Spatial models outperform all other models for Elevator Buildings (Type D) and Walk Up Apartments (Type C), particularly in Brooklyn, the Bronx, and Manhattan
- Classification tends to perform poorly in Manhattan vs. other Boroughs
- The Spatial models perform well in Manhattan for the residential building types (A, B, C and D)

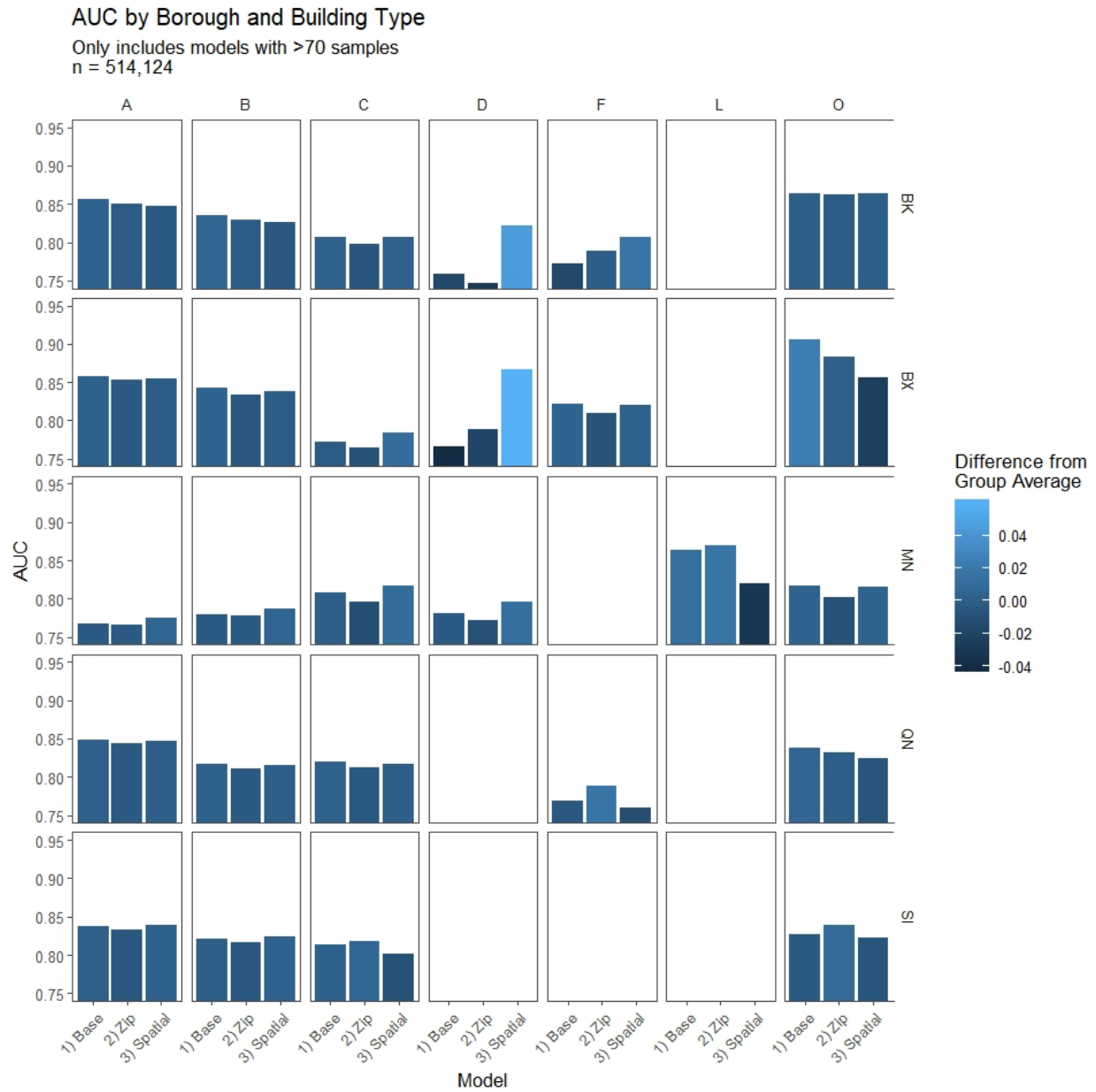


Figure 4.2: AUC By Borough and Building Type

If we rank the classification models' performance for each Borough and Building Type, we see that the Spatial models consistently outperform the Zip Code models, as shown in Table 4.4. From this (as well as from similar patterns seen in the regression models) we can infer that the Spatial data is a superior data engineering technique; however, the algorithm used needs to account for potential model overfitting. In the next section, we discuss refining

Table 4.4: Distribution and Average Model Rank for Probability of Sale by AUC across Borough and Building Types

| Model Rank | 1 | 2 | 3 | Average Rank |
|-------------|-------|-------|-------|--------------|
| Base | 16.2% | 12.0% | 5.1% | 2.22 |
| Spatial Lag | 11.1% | 13.7% | 8.5% | 2.09 |
| Zip | 6.0% | 7.7% | 19.7% | 1.69 |

the data used as well as employing different algorithms to maximize the predictive capability of the Spatial features.

4.3 Stage 2) Model Comparisons Using Specific Geographies and Building Types

Using the results from the first modeling exercise, we conclude that Walk Up Apartments and Elevator Buildings in Manhattan, Brooklyn and the Bronx are suitable candidates for prediction using our current assumptions. These buildings share the characteristics of being residential as well as being reasonably uniform in their geographic density. We analyze the performance of four algorithms (GLM, RF, GBM, and ANN), using three feature engineering techniques, for both classification and regression, making the total number $4 \times 3 \times 2 = 24$ models.

4.3.1 Regression Model Comparisons

The predictive accuracies of the various regression models were evaluated using RMSE, described in detail in the methodology section, as well as Mean Absolute Error (MAE), Mean Squared Error (MSE) and R-Squared. These four indicators were calculated using the hold-out test data, which ensured that the models performed well when predicting sale prices into the near future. The comparison metrics are presented in Table 4.5 and Figure 4.3. We make the following observations about Table 4.5 and Figure 4.3:

- 1) The ANN models perform best in nearly every metric across nearly all feature sets,

Table 4.5: Prediction Accuracy of Regression Models on Test Data

| Data | Model | RMSE | MAE | MSE | R2 |
|------------|-------|--------|--------|----------|------|
| 1) Base | GLM | 446.35 | 221.16 | 199227.6 | 0.12 |
| 2) Zip | GLM | 426.93 | 206.49 | 182270.1 | 0.19 |
| 3) Spatial | GLM | 382.32 | 195.00 | 146170.5 | 0.35 |
| 1) Base | RF | 387.99 | 174.24 | 150536.3 | 0.33 |
| 2) Zip | RF | 475.20 | 190.33 | 225811.7 | 0.00 |
| 3) Spatial | RF | 430.92 | 180.17 | 185695.5 | 0.18 |
| 1) Base | GBM | 384.11 | 179.27 | 147543.5 | 0.35 |
| 2) Zip | GBM | 454.53 | 186.00 | 206593.1 | 0.09 |
| 3) Spatial | GBM | 406.70 | 170.97 | 165408.0 | 0.27 |
| 1) Base | ANN | 363.02 | 178.58 | 131782.5 | 0.42 |
| 2) Zip | ANN | 360.88 | 171.22 | 130232.2 | 0.42 |
| 3) Spatial | ANN | 337.94 | 158.91 | 114202.0 | 0.49 |

with GBM a close second in some circumstances

- 2) ANN and GLM improve linearly in all metrics as you move from Base to Zip to Spatial, with Spatial performing the best. GBM and Random Forest, on the other hand, perform best on the Base and Spatial feature sets and poorly on the Zip features
- 3) We see a similar pattern in the Random Forest results compared to the previous modeling exercise using the full dataset: Base Features outperform both Spatial and Zip, with Spatial coming in second consistently. This pattern further validates our reasoning that Spatial features are highly predictive but suffer from overfitting and other algorithm-related reasons
- 4) The highest model R-Squared is the ANN using Spatial features at 0.494, indicating that this model can account for nearly 50% of the variance in the test data. Compared to the R-Squared of the more traditional Base GLM at 0.12, this represents a more than 3-fold improvement in predictive accuracy

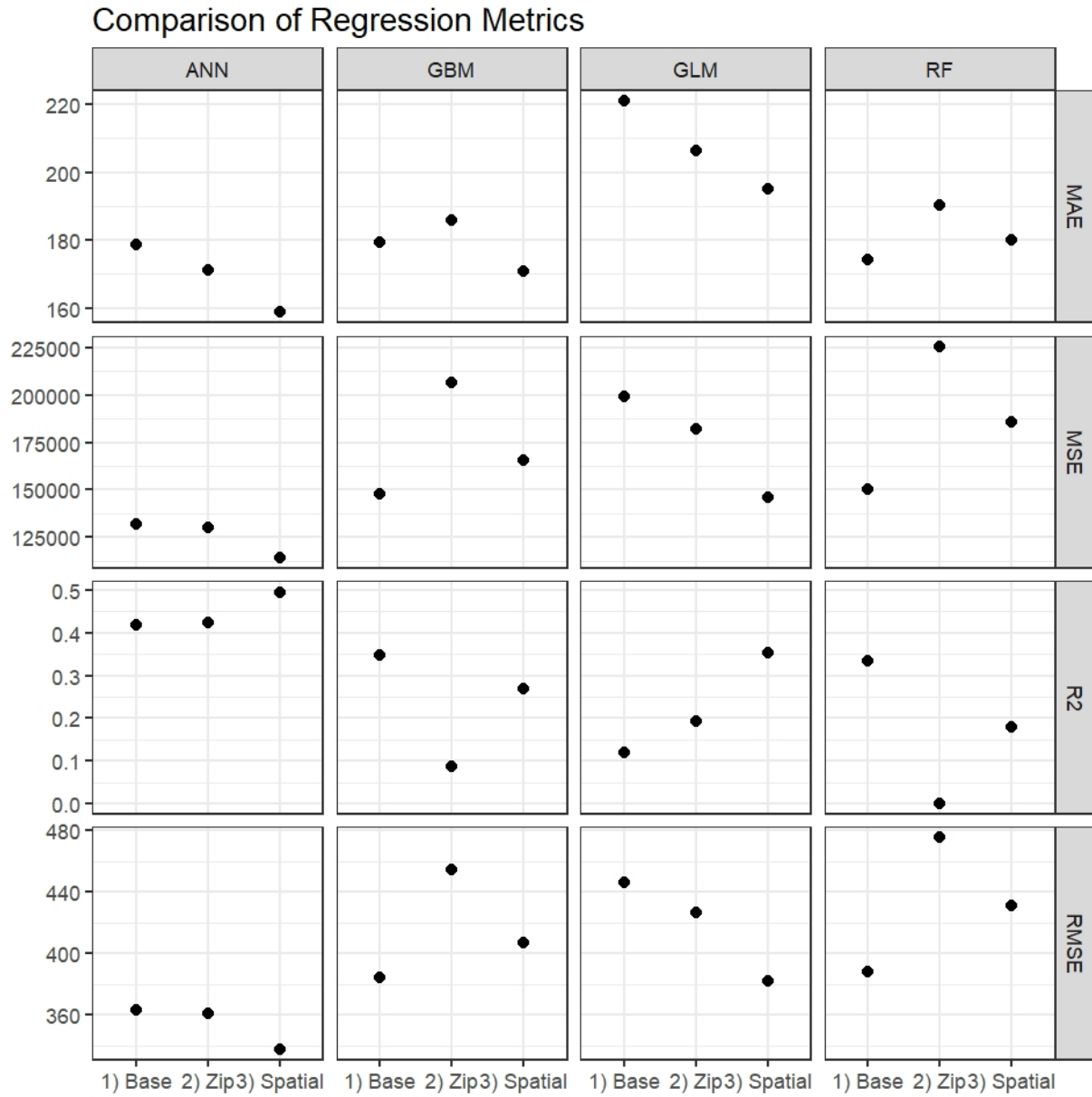


Figure 4.3: Comparative Regression Metrics

Figure 4.4 shows clusters of performance across R-Squared and Mean Absolute Error, with the ANN models outperforming their peers. This figure also makes clear that the marriage of Spatial features with the ANN algorithm results in a dramatic reduction in error rate compared to the other techniques.



Figure 4.4: Regression Model Performances On Test Data

4.3.2 Classification Model Comparisons

The classification models were assessed using AUC as well as MSE, RMSE, and R-squared. As with the regression models, these four metrics were calculated using the hold-out test data, ensuring that the models generalize well into the near future. The comparison metrics are presented in Table 4.6. Figure 4.5 shows the ROC curves and corresponding AUC for

Table 4.6: Prediction Accuracy of Classification Models on Test Data

| Data | Model | AUC | MSE | RMSE | R2 |
|------------|-------|------|------|------|-------|
| 1) Base | GLM | 0.57 | 0.03 | 0.17 | 0.00 |
| 2) Zip | GLM | 0.58 | 0.03 | 0.17 | 0.00 |
| 3) Spatial | GLM | 0.50 | 0.03 | 0.17 | -0.01 |
| 1) Base | RF | 0.58 | 0.03 | 0.17 | -0.03 |
| 2) Zip | RF | 0.56 | 0.03 | 0.17 | -0.06 |
| 3) Spatial | RF | 0.78 | 0.03 | 0.17 | 0.00 |
| 1) Base | GBM | 0.61 | 0.03 | 0.17 | -0.03 |
| 2) Zip | GBM | 0.61 | 0.03 | 0.17 | -0.03 |
| 3) Spatial | GBM | 0.82 | 0.03 | 0.16 | 0.04 |
| 1) Base | ANN | 0.55 | 0.03 | 0.17 | -0.03 |
| 2) Zip | ANN | 0.57 | 0.03 | 0.17 | -0.04 |
| 3) Spatial | ANN | 0.76 | 0.03 | 0.17 | -0.01 |

each algorithm/feature set combination.

We make the following observations of Table 4.6 and Figure 4.5:

- 1) Unlike the regression models, the GBM algorithm with Spatial features proved to be the best performing classifier. All Spatial models performed relatively well except the GLM Spatial model
- 2) Only 3 models have positive R-Squared values: ANN Spatial, RF Spatial, and GLM Base, indicating that these models are adept at predicting positive cases (occurrences of sales) in the test data
- 3) GLM Spatial returned an AUC of less than 0.5, indicating a model that is conceptually worse than random. This is likely the result of overfitting

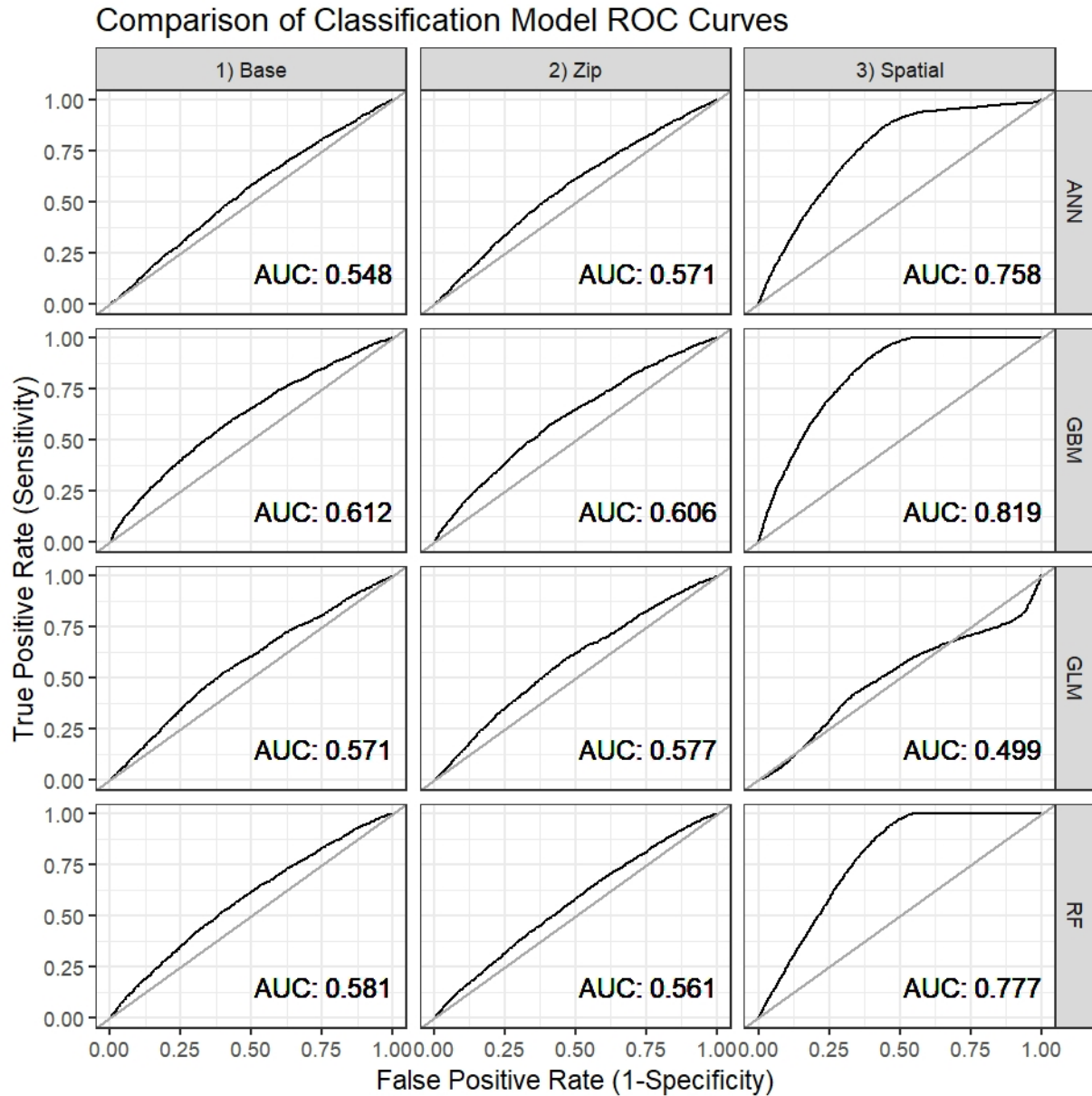


Figure 4.5: Comparison of Classification Model ROC Curves

Figure 4.6 plots the individual models by AUC and R-Squared. The Spatial models tend to outperform the other models by a significant margin. Interestingly, when compared to the regression model scatterplot in Figure 4.4, the classification models tend to cluster in their performance by feature set. In 4.4, we see the regression models tending to cluster by algorithm rather than features.

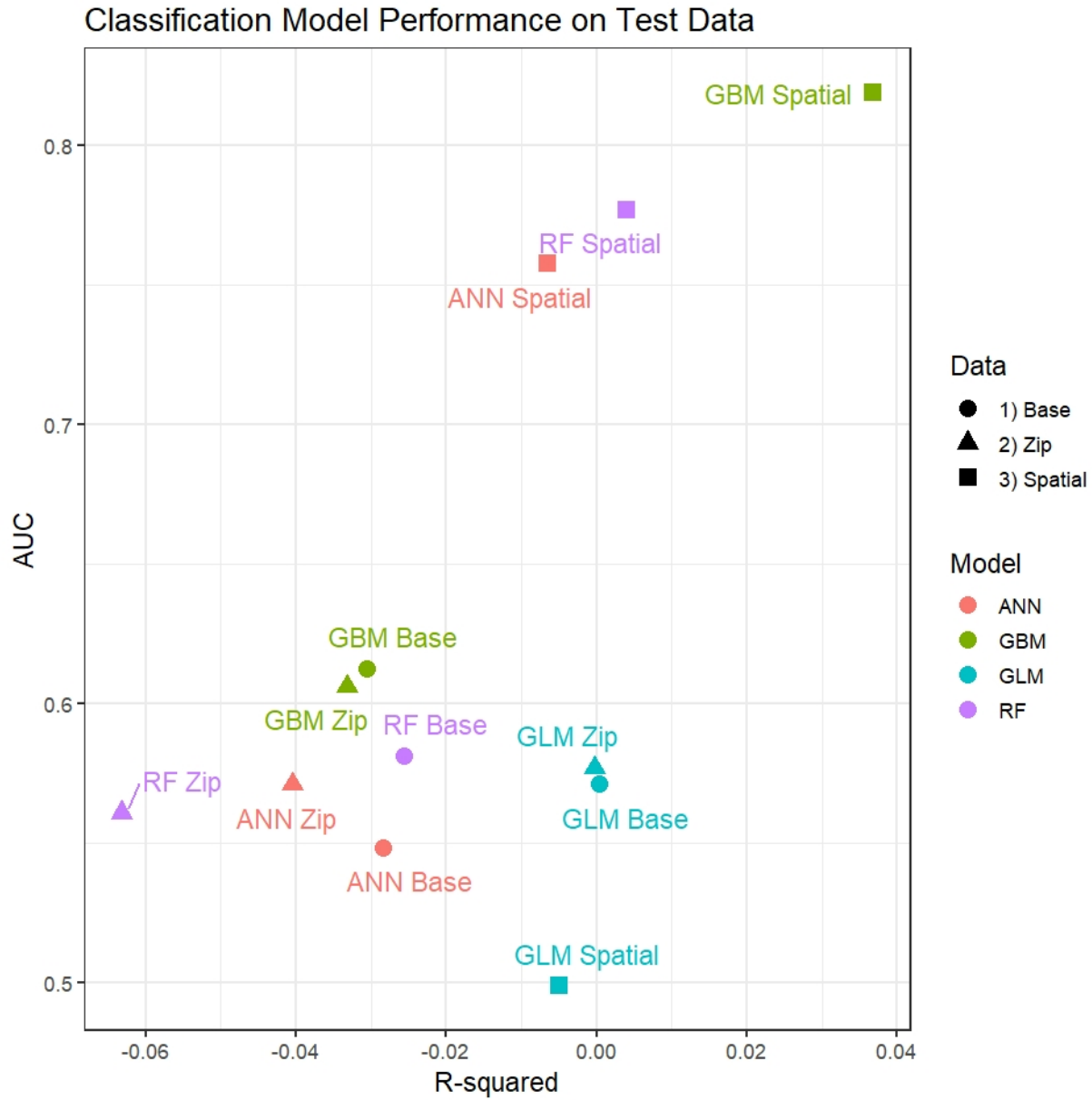


Figure 4.6: Scatterplot of Classification Models

4.4 Variable Importance Analysis of Top Performing Models

We calculated feature importance for each variable as the proportional to the average decrease in the squared error after including that variable in the model. The most important variable gets a score of 1; scores for other variables are derived by standardizing their measured reduction in error relative to the largest one. The top 10 variables for both the most successful

Table 4.7: Feature Importance of Top Performing Regression Model

| Variable | Description | Scaled Importance (Max = 1) | Cumulative % |
|-----------------------|------------------------------------|-----------------------------|--------------|
| BuiltFAR | Floor area ratio built | 1.000 | 1.80% |
| FacilFAR | Maximum Allowable Floor Area Ratio | 0.922 | 3.40% |
| Last_Sale_Price_Total | The previous sale price | 0.901 | 5.10% |
| Last_Sale_Date | Date of last sale | 0.893 | 6.70% |
| Last_Sale_Price | The previous sale price | 0.870 | 8.20% |
| Years_Since_Last_Sale | Number of years since last sale | 0.823 | 9.70% |
| ResidFAR | Floor Area Ratio not yet built | 0.814 | 11.20% |
| lon | Longitude | 0.773 | 12.60% |
| Year | Year of record | 0.759 | 13.90% |
| BldgDepth | Square feet from font to back | 0.758 | 15.30% |

Table 4.8: Feature Importance of Top Performing Classification Model

| Variable | Description | Scaled Importance (Max = 1) | Cumulative % |
|-------------------------------|--|-----------------------------|--------------|
| Percent_Neighbors_Sold | Percent of Nearby Properties Sold in the Previous Year | 1.000 | 21.90% |
| Percent_Office | Percent of the build which is Office | 0.698 | 37.20% |
| Percent_Garage | Percent of the build which is Garage | 0.634 | 51.10% |
| Percent_Storage | Percent of the build which is Storage | 0.518 | 62.40% |
| Building_Age | The Age of the building | 0.225 | 67.40% |
| Last_Sale_Price | Price of building last time is was sold | 0.165 | 71.00% |
| Percent_Retail | Percent of the build which is Retail | 0.147 | 74.20% |
| Years_Since_Last_Sale | Year since building last sold | 0.121 | 76.90% |
| ExemptTot | Total tax exempted value of the building | 0.069 | 78.40% |
| Radius_Res_Units_Sold_In_Year | Residential units within 500 meters sold in past year | 0.056 | 79.60% |

regression and most successful classification models are presented in tables 4.7 and 4.8.

We observe that the regression model has a much higher dispersion of feature importances compared to the classification model. The top variable in the regression model, BuiltFAR, which is a measure of how much of a building’s floor to area ratio has been used (a proxy for overall building size) contributes only 1.8% of the reduction in the error rate in the overall model. Conversely, in the classification model, we see the top variable, “Percent_Neighbors_Sold” (a measure of how many buildings within 500 meters were sold in the past year) contributes 21.9% of the total reduction in squared error.

Variable importance analysis of the regression model indicates that the model favors variables which reflect building size (BuiltFAR, FacilFAR, BldgDepth) as well as approximations for previous sale prices (Last_Sale_Price and Last_Sale_Date). The classification model tends to favor spatial lag features, such as how many buildings were sold in the past year within 500 meters (Percent_Neighbors_Sold and Radius_Res_Units_Sold_In_Year)

as well as characteristics of the building function (Percent_Office, Percent_Storage, etc.).

5 Future Research and Conclusions

5.1 Future Research

This research has shown that the addition of spatial lag features can meaningfully increase the predictive accuracy of machine learning models compared to traditional real estate valuation techniques. There are several areas that could be further explored regarding spatially-conscious machine learning models, some of which we mention below.

First, it became apparent in the research that generalization was a problem for some of the models, likely due to overfitting of the training data. We corrected for this issue by employing more robust algorithms; however, further work could be done to create variable selection processes or hyperparameter tuning to prevent overfit and generally accelerate model training and scoring.

Additionally, the spatial lag features seemed to perform best for certain boroughs and residential building types. We hypothesize that using a 500-meter radius to build spatial lag features, a distance which we arbitrarily chose, works best for this type of asset in these areas. Fotheringham et al. (2015) used an adaptive bandwidth technique to adjust the spatial lag radius based on cross-validation with much success. The techniques presented in this paper could be expanded to use cross-validation similarly to assign the optimal spatial lag radius for each building type and location. Furthermore, additional work could be done using cross-validation or a similar technique to refine the spatial lags further to include appropriate building types and distance weights tailored to each property.

Finally, this research aimed to predict real estate transactions 1 year into the future. While this is a promising start, 1-year of lead time may not be sufficient to respond to growing gentrification challenges. In addition, modeling at the annual level could be improved to quarterly or monthly, given that the sales data contains date information down to the day.

To make this system practical for combating displacement, it may be helpful to predict at a more granular level and further into the future.

5.2 Conclusion

Societies and communities can benefit materially from gentrification, however, the downside should not be overlooked. Displacement causes Economic Exclusion, which over time contributes to rising income inequality. Combating displacement allows communities to benefit from gentrification without suffering the negative consequences. One way to practically combat displacement is to predict gentrification, which this paper has attempted to do.

Spatial lags, typically seen in geographically weighted regression, were employed successfully to enhance the predictive power of machine learning models. The spatial lag models performed best for particular building types and geographies; however, we feel confident that the technique could be expanded to work equally as well for all buildings with some additional research. Regarding algorithms, Artificial Neural Networks performed the best for predicting sale price, while Gradient Boosting Machines performed best for predicting sale occurrence.

While this research is not intended to serve as a full early-warning system for gentrification and displacement, it is a step in that direction. More research is needed to help address the challenges faced by city planners and governments trying to help incumbent residents reap the benefits of local investments. Income inequality is a complicated and grave issue, but new tools and techniques to inform and prevent will help ensure equality of opportunity for all.

References

- Alexander Dietzel, M., Braun, N., & Schäfers, W. (2014). Sentiment-based commercial real estate forecasting with google search volume data. *Journal of Property Investment & Finance*, 32(6), 540–569.
- Almanie, T., Mirza, R., & Lor, E. (2015). Crime prediction based on crime types and using spatial and temporal criminal hotspots. *International Journal of Data Mining & Knowledge Management Process*, 5. <https://doi.org/10.5121/ijdkp.2015.5401>
- Antipov, 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*.
- Batty, M. (2013). The new science of cities. *MIT Press*.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
- Carmela Quintos PHD, M. (2015). Estimating latent effects in commercial property models. *Journal of Property Tax Assessment & Administration*, 12(2), 37.
- Chapple, K. (2009). Mapping susceptibility to gentrification: The early warning toolkit. *Berkeley, CA: Center for Community Innovation*.
- Chapple, K., & Zuk, M. (2016). Forewarned: The use of neighborhood early warning systems for gentrification and displacement. *Cityscape*, 18(3), 109–130.
- Clay, P. L. (1979). *Neighborhood renewal: Middle-class resettlement and incumbent upgrading in american neighborhoods*. Free Press.
- d’Amato, M., & Kauko, T. (2017). *Advances in automated valuation modeling*. Springer.
- DiMaggio, C. (2012). Spatial epidemiology notes: Applications and vignettes in r. Columbia University press.
- Dreier, P., Mollenkopf, J. H., & Swanstrom, T. (2004). *Place matters: Metropolitcs for the twenty-first century*. University Press of Kansas.
- Eckert, J. K. (1990). *Property appraisal and assessment administration*. International Association of Assessing Officers.

Fotheringham, A. S., Crespo, R., & Yao, J. (2015). Exploring, modelling and predicting spatiotemporal variations in house prices. *The Annals of Regional Science*, 54(2), 417–436.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4), 367–378.

Fu, Y., Xiong, H., Ge, Y., Yao, Z., Zheng, Y., & Zhou, Z.-H. (2014). Exploiting geographic dependencies for real estate appraisal: A mutual perspective of ranking and clustering. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1047–1056). ACM.

Geltner, D., & Van de Minne, A. (2017). Do different price points exhibit different investment risk and return commercial real estate.

Glass, R. (1964). *Aspects of change*. London: MacGibbon & Kee, 1964.

Greene, S., Pendall, R., Scott, M., & Lei, S. (2016). Open cities: From economic exclusion to urban inclusion. *Urban Institute Brief*.

Guan, J., Shi, D., Zurada, J., & Levitan, A. (2014). Analyzing massive data sets: An adaptive fuzzy neural approach for prediction, with a real estate illustration. *Journal of Organizational Computing and Electronic Commerce*, 24(1), 94–112.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. New York, NY, USA: Springer New York Inc.

Helbich, M., Jochem, A., Mücke, W., & Höfle, B. (2013). Boosting the predictive accuracy of urban hedonic house price models through airborne laser scanning. *Computers, Environment and Urban Systems*, 39, 81–92.

Hoffmann, J. P. (2004). *Generalized linear models: An applied approach*. Pearson College Division.

Johnson, K., Benefield, J., & Wiley, J. (2007). The probability of sale for residential real estate. *Journal of Housing Research*, 16(2), 131–142.

Joseph, D. S. (n.d.). The assessment of real property in the united states. *Special Report of the State Tax Commission, New York*, (10).

Kontrimas, V., & Verikas, A. (2011). The mass appraisal of the real estate by computational intelligence. *Applied Soft Computing*, 11(1), 443–448.

Koschinsky, J., Lozano-Gracia, N., & Piras, G. (2012). The welfare benefit of a home's location: An empirical comparison of spatial and non-spatial model estimates. *Journal of Geographical Systems*, 14(3), 319–356.

Lees, L., Slater, T., & Wyly, E. (2013). *Gentrification*. Routledge.

Miller, J., Franklin, J., & Aspinall, R. (2007). Incorporating spatial dependence in predictive vegetation models. *Ecological Modelling*, 202(3), 225–242.

Park, 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.

Pivo, G., & Fisher, J. D. (2011). The walkability premium in commercial real estate investments. *Real Estate Economics*, 39(2), 185–219.

Pollack, S., Bluestone, B., & Billingham, C. (2010). Maintaining diversity in america's transit-rich neighborhoods: Tools for equitable neighborhood change.

Rafiei, M. H., & Adeli, H. (2015). A novel machine learning model for estimation of sale prices of real estate units. *Journal of Construction Engineering and Management*, 142(2), 04015066.

Reardon, S. F., & Bischoff, K. (2011). Income inequality and income segregation. *American Journal of Sociology*, 116(4), 1092–1153.

Ritter, N. (2013). Predicting recidivism risk: New tool in philadelphia shows great promise. *National Institute of Justice Journal*, 271.

Schernthanner, H., Asche, H., Gonschorek, J., & Scheele, L. (2016). Spatial modeling and geovisualization of rental prices for real estate portals. *Computational Science and Its Applications*, 9788.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. <https://doi.org/https://doi.org/10.1016/j.neunet.2014.09.003>

Smith, N. (1979). Toward a theory of gentrification a back to the city movement by capital, not people. *Journal of the American Planning Association*, 45(4), 538–548.

Turner, M. A. (2001). Leading indicators of gentrification in dc neighborhoods: DC policy forum.

Watson, T. (2009). Inequality and the measurement of residential segregation by income in american neighborhoods. *Review of Income and Wealth*, 55(3), 820–844.

Zuk, M., Bierbaum, A. H., Chapple, K., Gorska, K., Loukaitou-Sideris, A., Ong, P., & Thomas, T. (2015). Gentrification, displacement and the role of public investment: A literature review. In *Federal reserve bank of san francisco* (Vol. 79).